OPAALS PROJECT

Contract n° IST-034824

WP9: Communication and Dissemination

Del 9.4 – 1st OPAALS Conference Proceedings

Project funded by the European Community under the "Information Society Technology" Programme

Contract Number: IST-034824
Project Acronym: OPAALS

Deliverable N°: 9.4
Due date: May 2008
Delivery Date: 15 July 2008

Short Description:

Proceedings of the 1st OPAALS Conference.

November 26-27 2007, Roma Eventi Conference Centre, Rome (Italy).

The proceedings contain the contributions in alphabetical order,with an Organizing and Program Committee description and an Introduction of the project.

Workshop website:

http://oks.opaals.org/website/conference.php?head=head-9e10e9541b241c665acd55bece8f80ae322f459b

Author: CREATE-NET (Francesco Botto, Csaba Szabo)
Partners contributed: Paolo Dini (LSE), Frauke Zeller (Kassel University)
Made available to: all project partners and EC

Versioning		
Version	**Date**	**Name, organization**
1	28/5/2008	Francesco Botto (CN)
2	15/7/2008	Francesco Botto (CN)

Quality check
Internal Reviewers: Paul Krause (UNIS), Mehita Iqani (LSE)

Dependences:

Work Packages	All the WPs could benefit from this deliverable
Partners	All partner could benefit from the reading of this deliverable
Domains	The biological science, computer science and social science domains are involved in this deliverable
Targets	Research community

Table of Contents

Conference Organizing and Program Committees... p. 6

Introduction.. p. 7
F. Botto and P. Dini

Social Network Analysis of Dynamic Forum. "Conversê" Case Study..................... p. 10
F. A.B. Colugnati and S. F.A. Barretto

More Notes on Abstract Algebra and Logic: Towards their Application to Cell Biology and Security.. p. 17
P. Dini and D. Schreckling

An Efficient Algorithm for Conceptual Subspace Clustering................................ p. 42
H. Fu

An Examination of the Use of Inner Source in Multinational Corporations. A Preliminary Framework to Understand Inner Source Software Development........ p. 48
G. Gaughan, B. Fitzgerald, L. Morgan and M. Shaikh

An Evolutionary Framework for Language in a Digital Ecosystem........................ p. 61
S. Hendryx

A Business Model of a Digital Business Ecosystem Negotiation Platform............. p. 64
V. Hoyer and K. Stanoevska-Slabeva

Community Visualisations in Open Knowledge Space.. p. 71
J. Huhtamäki

An Evolutionary Multidimensional Trust Model for Digital Ecosystems.................. p. 78
M. Ion, L. Telesca, J. McGibney and D. Botvich

Peripheral Simulation Framework for Digital Ecosystems.................................. p. 88
T. Kurz, J. Nougera, R. Eder, T. Heistracher and J. Gabaldon

Ecosystem-Oriented Architectures.. p. 95
S. Lain, A. Munro, G. Briscoe and P. Dini

Linguistic Dimension of Terms: Towards Community's Common Vocabulary and Concepts.. p. 119
O. Lapteva and H. Peukert

Enabling Trust with a Distributed Accountability Model for Digital Ecosystems..... p. 127
P. Malone and B. Jennings

Declarative Service Composition with SBVR.. p. 136
A. Marinos and P. Krause

Transaction Vectors A Non-Interleaving Semantics for Long-Running Transactions.. p. 143
S. Moschoyiannis, A. Razavi and P. Krause

Interpretation Logics.. p. 152
O. Nykänen

Mediating the Dialectic Relations between Indigenous Knowledge and Identity: Lessons from DEAL Project.. p. 160
D. Pattanaik, R. Sarkar and J. Chatterjee

Interface, Perception and Visualization... p. 169
R. Piazzalunga

The Impact of DEAL on Community Network: A Case Review............................ p. 172
R. Rajagopalan and R. Sarkar

The Agent-based Digital Business Ecosystem: Towards the Semantic Web.......... p. 177
A. Razavi, S. Moschoyiannis and P. Krause

Self-Organising Map Based Tag Clouds. Recreating Spatially Meaningful Representations of Tagging Data.. p. 188
J. Salonen

Digital Ecosystems – Community Networks or Networked Communities?............. p. 193
R. Sarkar and R. Rajagopalan

Community Currencies. A Critical Approach to Community Currency Systems....... p. 202
P. Tsatsou

A Framework for Digital Ecosystems: The 'First' of Theoretical Interpolations....... p. 211
F. Zeller and P. Dini

Conference Organizing and Program Committees

General Chair

Paolo Dini	London School of Economics (UK)	p.dini@lse.ac.uk

Steering Committee Chairs

Paolo Dini	London School of Economics (UK)	p.dini@lse.ac.uk
Andrea Nicolai	T6 (IT)	a.nicolai@t-6.it
Csaba Szabo	CREATE-NET (IT)	csaba.szabo@create-net.org

Program Committee Chairs

Brian Fitzgerald	University of Limerick (IE)	brian.fitzgerald@ul.ie
Paul Krause	University of Surrey (UK)	P.Krause@surrey.ac.uk
Frauke Zeller	University of Kassel (DE)	kinema@uni-kassel.de

Technical Program Committee

Hristo Koshutanski	University of Malaga (ES)	hristo@lcc.uma.es
Daniele Miorandi	CREATE-NET (IT)	daniele.miorandi@create-net.org
Paul Malone	Waterford Institute of Technology (IE)	pmalone@tssg.org
Gary Gaughan	University of Limerick (IE)	gary.gaughan@ul.ie
Maha Shaikh	London School of Economics (UK)	M.I.Shaikh@lse.ac.uk
Alastair Munro	Dundee University (UK)	a.j.munro@dundee.ac.uk

Publications and Publicity Chair and Co-Chair

Neil Rathbone	London School of Economics (UK)	rathbone@daventryhouse.com
Mehita Iqani	London School of Economics (UK)	m.iqani@lse.ac.uk

Local Arrangement Chair

Carissa Prosperi	CREATE-NET (IT)	carissa.prosperi@create-net.org

Web Chair

Javier Noguera	Techideas (ES)	javier.noguera@techideas.es

Introduction

F. Botto and P. Dini

This OPAALS project deliverable is a collection of papers submitted to and presented at the 1st OPAALS Conference, held in Rome on November 26-27, 2007. It is an empirical illustration of the immense depth and complexity of the project at hand, as well as an output of efforts of interdisciplinary exchange. Given the number and the complexity of the issues involved, it could be difficult for a newcomer to approach the OPAALS research. In fact also the people involved in the project experienced a similar sense of disorientation and then keep feeling that the project challenges disciplinary borders and acquired conceptual understandings. Therefore this introduction will first briefly present the Digital Ecosystem concept, in order to introduce the OPAALS Network of Excellence and the scope of the conference to the reader.

Digital Ecosystems

The OPAALS project is a European Commission 6th Framework Programme IST project (No. IST-2005-034824), a part of the Digital Ecosystems (DE) cluster. The ecosystem idea is that individual members should develop in clusters to survive and evolve inside a specific environment. By using this metaphor new ideas have been developed recently: Business Ecosystems, Digital Ecosystems and Digital Business Ecosystems.

Business Ecosystems (BEs) were suggested by Moore (1993) as a way to consider and sustain the recent need of business agencies to develop in clusters. In this definition big enterprises like Apple, IBM and Ford are examples of BEs leaders with powerful and not democratic role. Differently, DEs have been recently adopted by the European Commission as a way to help SMEs to 'go digital' without a fixed business leadership and under a public policy intervention (EC, 2001).

Adding 'digital' in front of Moore's BEs, the EC started supporting a new interpolation between Business and Digital Ecosystems (Nachira, 2002). BEs are intended as economic communities that balance cooperation and competition. The early DBE understanding of the EC is facing two challenges. First, the DE technical definition has been recently criticized by social science researchers inside the OPAALS project, who argue that it is crucial to consider simultaneously the technological and the social dimensions of ecosystems. Second, the original 'business' attitude is increasingly being regarded as a specific instance: Digital Ecosystem is increasingly referring to the most general concept ('root class') that encompasses all other types of digital ecosystem as instances: DBEs, DEs of knowledge, DEs of cultural expression, etc. The most narrow technical interpretation of Digital Ecosystem as the technical infrastructure based on P2P distributed

Open Source software technology that enables BEs in a digital world is still useful in some specialised contexts where the functional mapping from biological models is emphasised.

A Digital Ecosystem is therefore composed of heterogeneous and autonomous users, companies and resources which interact in a complex, distributed and dynamic digital environment. The complexity of interactions between different institutions is increased by the fact that institutions sometimes compete against each other and other times collaborate with each other and form stable and unstable federations. DEs are interconnected by a network to form a complex and dynamic environment (Nachira, Nicolai, Dini, Le Louarn and Rivera Leon, 2007).

The OPAALS Project

Inside the EC Digital Business Ecosystems projects cluster, the OPAALS *(Open Philosophies for Associative Autopoietic Digital Ecosystems)* Network of Excellence's main objectives are to develop an integrated scientific foundation for Digital Ecosystems research and to build a community of researchers in this new field. Partners from seven different European countries plus Rwanda, India and Brazil participate in the project. The very different research communities involved in the project are organized in three domains: natural science, computer science and social science.

More than merely creating the DE science, the expected project impact is to strengthen the ability of European SMEs to participate in advanced software application development and in an innovative open knowledge sharing paradigm. The project is based on Open Source Software/Knowledge ideas, autopoietic systems theory (autonomous generation), theories of association from social science (communities of practice, socio-technical theories, reflexive research methodologies, and open structures of governance), and the need to sustain advanced services based on distributed Peer-to-Peer (P2P) architectures.

This will require the accomplishment of the eight sub-objectives of the project:
1. Theoretical foundation for associative autopoietic DEs.
2. Durable integration of different disciplinary viewpoints.
3. Initiate recursive, reflexive and self-reinforcing knowledge creation and community building.
4. Creation of an Open Knowledge Space.
5. Definition of sustainable Open Source/Knowledge and business models.
6. Identification of dynamic P2P architectures for distributed services.
7. Context-dependent automatic code structure and workflow generation.

The four-year long project started in June 2006. In the first year it mainly provided some disciplinary conceptual understandings and empirical insights to start work on jointly. The interdisciplinary work needed time to start concretely. As social sciences researchers involved in the project noticed, the different 'epistemic cultures' are far from naturally understanding each other. After many disciplinary and cross-disciplinary project meetings and workshops, the 1st OPAALS conference was intended to share the work done so far

with the broad research community.

The 1st OPAALS Conference

The First OPAALS Conference should be more properly called the First OPAALS Workshop. Due to the comprehensible challenges in providing clear enough and sufficiently stabilized scientific results after the first year of the project, the project coordinator suggested to change the character of the event. It was decided to give the interdisciplinary group of partners the chance to start discussing and discovering each other in a smaller and focused workshop setting. In fact, it had been the third occasion for all to meet together since the Kick-off Meeting and the General Meeting in March 2007. The second OPAALS Conference will be larger and will involve more participants from outside the project. The final conference will be the largest and the one with greatest international scope.

Even if the workshop contributions did not reach the goal to fully describe the DEs issues, they started the process. It is the nature of workshops and conferences to create spaces for intellectual exploration, learning and discussion. In the case of OPAALS, however, this exchange extends to the inter-disciplinary and the inter-epistemological. It is (somewhat) unprecedented that social scientists attend and listen to technical presentations on software modelling or algebra, or that computer scientists or mathematicians attend and listen to presentations on linguistic or social philosophy. The mere fact that the papers have been presented in one space, and were attended to by an interdisciplinary audience, show movement towards interdisciplinary communication, and are evidence of the beginnings of an *actual* exchange.

The papers included in these proceedings reflect the status of the research in most of the project's activities. Hopefully they also begin to show a convergence of how these different strands of research fit together. It is therefore an opportunity to start avoiding the disciplinary-based organization of contributions. These proceedings therefore contain the papers in alphabetical order of the first author and consist in an aggregation of the PDF files provided by authors.

References:

European Commission (2001). Communication from the Commission: "Helping SMEs to "go digital". COM(2001)136 final. 2001, 13 March.

Moore, M. (1993) Predators and Prey: A New Ecology of Competition. Harvard Business Review May-June 1993, p.75-86.

Nachira, F. (2002). "Toward a network of digital business ecosystems fostering the local development", www.europa.eu.int/information_society/topics/ebusiness/godigital/sme_research/index_en.htm

Nachira F., Nicolai A., Dini P., Le Luoran M. and Rivera Leon L. (Eds. 2007), Digital Business Ecosystems, European Commission, Bruxelles, www.digital-ecosystems.org/book/de-book2007.html

Social network analysis of dynamic forum

"Conversê" case study

[OPAALS Conference 2007]

Fernando A.B. Colugnati, Saulo F.A. Barretto
Research Institute for Technology and Innovation - IPTI
São Paulo, Brasil
fcolugnati@ipti.org.br, saulo@ipti.org.br

Abstract— **The understanding of network structures and dynamics plays an important role in the study of Digital Ecosystems, once the interpretation of these characteristics may explain some phenomena in communities building and sustainability, like impact of interventions, factors that could affect communities rising, among others. This paper describes a preliminary analysis over one year of a Brazilian community about Free License Open Source Software (FLOSS) by means of dynamic forum threads. Social Network Analysis is used to explore the dynamics of this network.**

Keywords- Social Network Analysis; community dynamics; Conversê

I. Introduction

Concept of networks relies on a permanent conflict among different frameworks in social science, frequently based on dualities like subject/society, actor/structure, subjectivism/objectvism, where each one put analytical emphasis in one of those parts of the double. Social Network can be understood as a representation of autonomous actors, linking ideas and/or resources around shared values and interests (Marteleto, 2001), where the double actor and structure are disjoint units that can help in the study of patterns and dynamics of social relationships.

Although association and collaboration in network are as older as the humankind, just in the last decades Social Network (SN) has been considered an organizational tool to explore complex structures, establishing a new paradigm in the analysis of network phenomena. This tool, has extrapolated social science domain and is currently adopted in many different research areas, like biology (ref) and epidemiology (ref).

The understanding of network structures and dynamics plays an important role in the study of Digital Ecosystems, once the interpretation of these characteristics may explain some phenomena in communities building and sustainability, like impact of interventions, factors that could affect communities rising, among others.

Social Network Analysis (SNA) is a set of mathematical and statistical tools that, using the distribution of node degrees and measures of distance, can estimate meaningful parameters that responds questions about information flow like centrality, articulation and prestige, always regarding actors and network.

This paper describes a preliminary analysis over one year of a Brazilian community about Free License Open Source Software (FLOSS), within the *Cultura Digital* (Digital Culture) project from Brazilian Ministry of Culture. The idea is to apply basic concepts of SNA exploration, trying to identify communities formation, leadership and the network activity evolution.

Sociograms and network parameters are presented for the whole period, and 4 months are studied in more details, due to singular characteristics that can abstract the dynamic in a simpler way.

Section II describes the network studied, the platform used and how the network can help to understand digital ecosystems. Section III brings a brief introduction to SNA, including the definition of the parameters used along the analysis. Section IV presents the results about the whole one year period, and the particular cases. Section V is about some concluding remarks and some discussion about results.

II. The "Conversê" Network

Conversê (a popular denomination for jamming or chat in Brazilian Portuguese) is a platform developed in the *Cultura Digital* (Digital Culture) program from Brazilian Ministry of Culture (*MinC*). This platform aims to be an open discussion workspace for *Cultura Digital* participants (mainly) and any other person interested in the subjects posted there. The idea behind *Conversê* is to enhance the creation of social networks, where Brazilian cultural stakeholders can share experiences, establish collaboration and activism.

Roughly speaking, it is a dynamic forum following the so called Web 2.0 concept, where participants can exchange experiences, asking for help and discuss about politics, technology and other issues. All posts can be tagged using folksonomy, such that main and more frequently used and more accessed tags gain highlights in the page.

The platform, www.converse.org.br, was developed using Drupal technology, and it is on line since 2005. Conceived to

be used within the *Cultura Digital*, it is nowadays accessed by common users that found an interesting virtual space to share opinion and knowledge.

Data used in this paper are related to the period from January to December 2006.

III. Social Network Analysis (SNA)

SNA introduced herein is existentially exploratory, based mainly in concepts described in Wasserman [1]. The research in this area has been growing, and more sophisticated approaches were developed using stochastic process [2], Dynamic Models [3] and Bayesian statistical modeling [4] allowing researchers to handle more complex and larger structures.

Basically, SNA can be divided in graphical exploration (visualization), quantification of parameters that describe some nodes and network features, and structure exploration.

A. *Sociograms*

Sociograms are the basic visualization tool for network representation, using concepts from theory of graphs. Each node represents an actor in the network, and the links among actors can be edges (ties, for some authors) or arcs (arrows), for undirected or directed networks respectively. An edge occurs when A replied to B and B replied to A, i.e., there is reciprocity between them. An arc occurs when there is no reciprocity.

Graphical attributes are commonly used to represent some properties. For example, thickness of the tie and/or color indicate the intensity and/or type of connection between two actors. Also, different sizes, shapes and colors of nodes can represent specific features of the actors, generally representing the parameters estimated.

A major problem for sociograms is the optimal visualization. How to better represent the structure? Algorithms were developed, considering the network as a physical system of particles, and they search for the configuration with minimum cross-edges in an equilibrium state for the system. The most used ones are the Kamada-Kawai [5] and the Fruchterman-Reingold [6]. The former is not recommended for large scale structures, and the last can be used to that, but the convergence for the equilibrium state sometimes is quite slow. Other methods has been developed. A promising one is the Latent Position, based on geometric and Bayesian statistical methods [7].

B. *Quantification*

Quantitative parameters used in SNA comes, in general, from the graphs theory and are based mainly in the concept of Degree and Distance. All these parameters can be estimated for the actors and for the whole network, giving different interpretations.

An actor **Degree** is defined as the number of links that it has, that is, we say that node **A** has degree *k* if there are **A** is connected to other *k* actors. In the case of directed networks, Outdegree and Indegree are defined, or the number of arcs that depart from A or reach A, respectively. The actor degree represents the intensity in which it is connected to the network structure. The higher the degree, the most active is the actor. In the same way a network has a degree, which is the sum of all actors degree, and represents the cohesion of the network.

The **Distance** between two actors *u* and *v*, *D(u, v)*, is the number of steps in the minimum path from *u* to reach *v*. If the network is undirected so *D(u, v)=D(v, u)*, but it is not always true in the case of directed networks.

Having this two concepts in mid, we can quantify actors in terms of three three basic parameters:

- Degree centrality: it is the degree itself. What matters is where those connections lead to and how they connect to the not directely connected actors. This parameter can be interpreted as a level of activity or communication an actor has.
- Betweenness centrality: is the proportion of all distances between pairs of other that include this node. An actor with high betweenness has great influence over what flows in the network, indicating important links and single point of failure. Articulators have high betweenness.
- Closeness centrality: is the measure of how close an actor is to everyone else. It is calculated as the number of all other actors, divided by the sum of all distances between the actor and all the others. The value indicates how fast an actor can reach any other. A “close” actor has

These measures have their counterparts for the whole SN as well, representing an average of them and varying from 0 to 1:

- Degree centralization: indicates how centralized a network is. A star network has maximum centralization, with value 1.
- Betweenness centralization: is an indicator of transitivity in the network. The closer to 1 the value, more intermediation in the SN. However, values for this parameter are always low, and the better use is the comparison in time or to other SN with similar morphology.
- Closeness centrality: is an indicator of how spread the network is. The value 0 indicates the most spread one. A problem for this measure for the whole SN is the fact that it depends on weak ties

C. *Structure exploration*

Concepts about the network structure helps to identify substructures, communities and/or clusters that can represent different behaviors, dynamics or other characteristic. Cliques, Structural Holes and Bi-components are the most used ones.

Structural hole concerns ego-centered network, that is, network studied from a specific actor and its relationships,

mostly based on weak ties. Although very important in a deeper SNA, this is not the scope of this paper, and will not be explored here.

For the not ego-centered network, the Bi-component concept is useful in identifying sub-networks that can be interpreted as communities. To better understand this concept, it is necessary to introduce the concepts of bridge and articulator. A *bridge* is a tie that if removed, increases the number of sub-structures in the network, such that, is the unique link between two sub-structures. An *articulator* is an actor that if removed, increases the number of components too, such that, it is the bottleneck among two or more sub-structures. It is also called a cut-vertex. Despite the fact of almost equal definitions, it is important to note that the former is related to a tie, and the last is related to a node.

Finally, a Bi-component is a sub-structure of minimum size 3 that does not contain an articulator. It is a closed network that cannot be splitted, but can contain weak ties. It should not be confounded with Cliques. A Bi-component is a clique, but the reversal is not always true. In this case study this concept will be used to identify communities and its articulators.

IV. *Metric Used*

SNA depends on the definition of what should be represented in terms of the social ties. In Forum platforms a natural choice for tie definition is reply to the author of any thread posted. Once this definition is stated, metric is the following step to be defined for network representation, bringing quantitative meaning for the ties. Assuming more importance for ties that had occurred firstly between actors in terms of order of response, we adopted the following weights:

4, if it was the first answer to the thread;

3, if it was the second;

2, for the third;

1 for any further reply.

Other aspect is the order of ties. The more realistic modeling for this kind of network is the use of arcs, directed ties. However, exchanges in dynamic forums are in great part two way directed, but with different weights for each direction, turning analysis more complex. In this first exploration we will be restricted to edges, undirected ties, for simplicity sake. Since that actors can chat about different threads and can reply more than once in the same thread, the weight for a tie between two actors is the sum of all single ties during that period, thread and direction independent.

The networks presented herein represent all activities enrolled in 2006, from January by December. They will be analyzed as independent snapshots, month-by-month, what means that network is formed by answers to threads posted that month, but those answers could be made anytime after the post (e.g.: a thread posted in May can comply answers posted in June, July, whenever until December).

V. Case Study

Table 1 shows network measures. It is clear that the total number of registered users is not correlated with the number of active actors in the network, its degree. But in this case, the degree is correlated with the Betweenness index, that express the capability of information flow through the network, besides central actors. We can interpret that during months when the network had more activity, there were a more decentralized structure. Indeed, the Closeness Centralization for the network could not be computed, because of the absence of weak ties. These facts can be interpreted as very dense and cohesive SN in the middle year months, such that any actors can interact with everyone by complete paths. However, betweenness increased, because information intermediation was enhanced by the dense structure. Degree Centralization confirms this dynamic of centralization as well, starting from 0.13 in January, reaching an average of 0.15 between June and August, and decreasing to 0.10 in December.

It happened mostly after some regional workshops, when the Cultural Hot spots received specialists in multimedia techniques and free softwares, triggering many threads about what was learned, doubts, critics and suggestions, among other issues.

During the months of February until July, many peripheral activities occurred (see June, Figure 2), giving some clue about independent structures that could rise. This kind of behavior was absorbed by the main SN, and returned just in the last month of the year.

TableI- Measures for networks by month of activity

Month	*Users Registered*	*Network Degree*	*Degree Centralization*	*Betweenness Centralization*
January[a]	2003	47	0.135	0.020
February	2167	53	0.213	0.032
March	2409	30	0.073	0.008
April	2975	44	0.173	0.030
May	3380	71	0.121	0.010
June[a]	3836	108	0.172	0.032
July	4318	73	0.150	0.030
August	4813	108	0.174	0.020
September[a]	5430	103	0.116	0.023
October	5451	88	0.103	0.016
November	6783	65	0.063	0.080
December[a]	7516	50	0.105	0.011

a. Months discussed.

To better explore the dynamics, four months were elected to be analyzed and presented, due their good representation of typical behaviors that have been found along the whole year: January, June, September and December.

A. January

The first month of the year presented 2003 users but only 47 active actors. The first sociogram presents the general network, where it is possible to identify a centralized structure. It was possible to identify Authorities, represented by the nodes size in the Figure 1. They are actors that centralize and spread information, and the greater the size of the node, the highest the influence of them in the information flow. When centralization happens, it is common to be formed sub-structures, that can represent communities or groups formed around a specific subject.

In the up-right corner in the figure it is possible to identify a Bi-component (circle), formed by 5 actors, linked to the main structure just by one cut-vertex (articulator).

B. June

From May to June the network degree increased 52%, with 108 active actors, while this rate for the number of users registered was just 13%, totalizing 3836 users. This fact can be explained, as said before, by the regional workshops that occurred in many cities, like Porto Alegre (southern region), Belém (Northern region), Ouro Preto (South-east) and Anápolis (West-center). Great part of the increasing in the SN can be observed in the peripheral activity.

From Figure 2 and from the measures (Table 1) it can be said that it is the most centralized structure, where two actors are the main centers (bigger black nodes) with high activity concentrated around them (darker and thicker ties). That actors were contracted to do articulation, and this sociogram confirms they did their job. Two sub-networks claim attention (circles), being weakly connected to the main structure.

C. September

This is the one of the most active month, with 4813 users registered and 108 active nodes in September, a huge increase in the network degree. As can be seen in the sociogram, Figure 2, it is hard now to identify any sub-structure, the configuration is dense, highly between-centralized (not ego or close-centralized) and the central nodes are really close one each other in the center of the structure. Peripheral activity is almost inexistent. Even the algorithm for Bi-component was unable to identify articulators, which means that there are no unique bridges among any triad set (or higher configuration). It is just possible to identify Islands, sets of high activity compared to the whole network (Figure 3). In September, 4 islands are evidenced, and they are represented as the circles in the sociogram.

D. December

After a year of activity, the network presents the most interesting configuration, with few (50) but very active actors. It is possible to say that this time communities have been created, with important role for the 5 articulators identified bridging them. Figure 4 presents the activity in the network, and Figure 5 the communities formed, once articulators are deleted.

It is curious to note a central articulator. This actor, a musician, was a very active one since the beginning. He was not part of the coordination or articulation team contracted by MinC, but he emerged as a natural central point, when the SN was dense, and an important articulator, bridging at least four different communities.

VI. Remarks

This paper aims to explore a SN formed by threads and responses in a dynamic forum created for a specific purpose, an open platform for the FLOSS community involved in the *Cultura Digital* project.

Despite the fact that Conversê was on-line since middle 2005, its use was restricted, in primary beta versions, so that January 2006 can be considered the starting point, when community involved, and other actors from the whole society, began to use it to discuss subjects related to the project, share knowledge about free software and the multimedia toolkits, among other subjects.

Looking for the sequence of figures presented herein, from 1 to 5, it is interesting to note a behavior that reflects an ecosystem evolution. The SN started with few actors, almost as a unique structure, formed by weak ties. Even though, presence of an articulator bridging a sub-structure can be observed, reflecting the prior existence of a community. June is a key month, due to the regional workshops carried out. The network "explodes" in number of actors, and in the level of activity mainly in the center of the main component, making this SN highly centralized. But many sub-structures and peripheral activity is observed, due to the diversity of issues that composed the forum threads. In September, those peripheral activities, including the communities, were absorbed by the main structure, forming a dense and cohesive network, with high transitivity, making hard to find any kind of sub-structure. It can be said that all communities are sharing experiences among each other, and it is reflected in the centered islands of activity along the diametric region of the SN. After this almost chaotic behavior, December shows a small SN, like January, however quite structured, with many communities with intense flow of information, and the figure of one central articulator.

The SN started, "exploded", increased the entropy and after that "imploded", however totally structured. Almost like a specie that arise, multiply fostered by the environment conditions, starts to associate with other ones, and afterwards experience a selection, such that selected individuals maintained their synergy and collaboration, living in communities, that still exchange some needs and offers. This behavior can be explored for Digital Ecosystem stakeholders.

The forum presented here did not experienced any kind of control or targeted intervention, the network evolution was totally free. It is possible to argue that the regional workshops served as intervention, but it was not intended to do so. In terms of Digital Ecosystems, where types of control and interventions can be applied, and even needed,it is important to identify where and when do that. Identification of articulation and central points is essential to better understand in which nodes of the network one should intervene.

This kind of real data and monitoring of the digital environment by means of SNA, are good tools for Digital Ecosystems management, fostering and control.

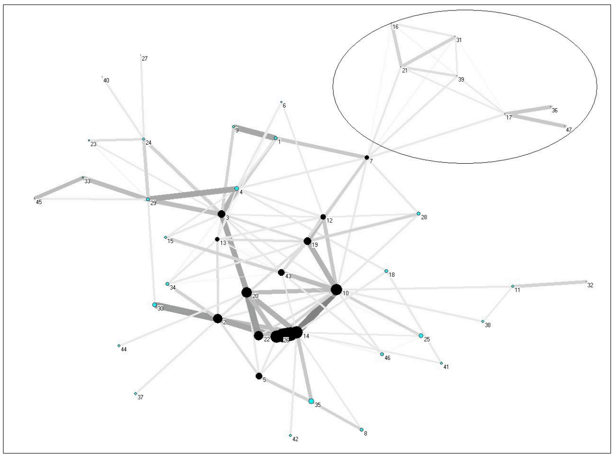


Figure1- January sociogram. Gray scale and thickness for ties represent intensity. Node size represents Betweeness Centrality.

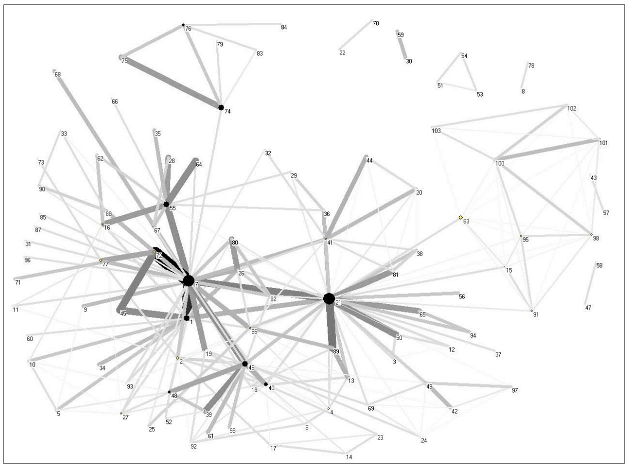


Figure2- June sociogram. Gray scale and thickness for ties represent intensity. Node size represents Betweeness Centrality.

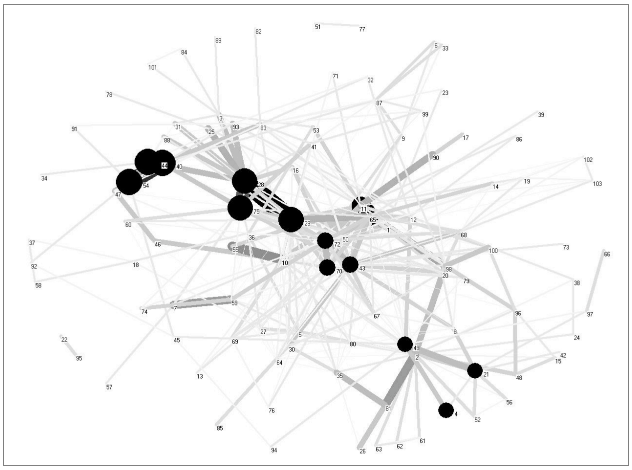


Figure3- September Sociogram. Gray scale and thickness for ties represent intensity. Node size represents Islands of Activity.

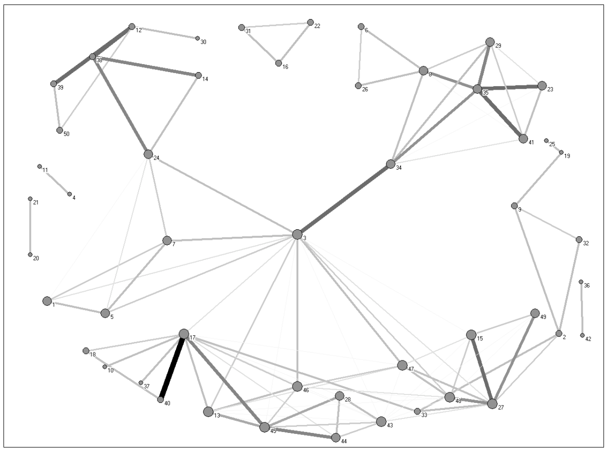


Figure4- December sociogram. Gray scale and thickness for ties represent intensity.

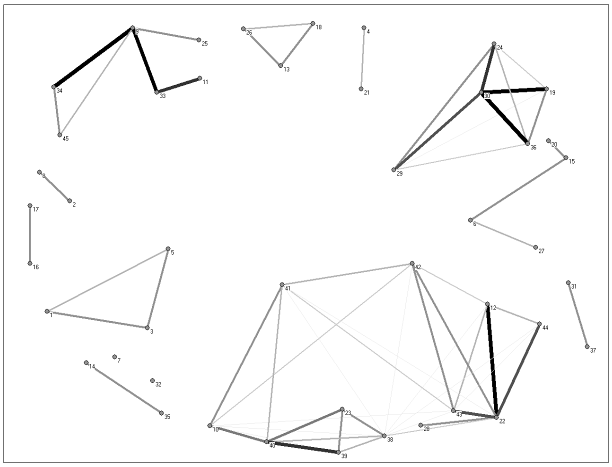


Figure5- December sociogram without articulators. Gray scale and thickness for ties represent intensity.

ACKNOWLEDGMENT

We would like to thank Luis Fagundes for data extraction, and all Cultura Digital participants.

REFERENCES

1] S. Wasserman, K. Faust, 1994. Social Network Analysis: Methods and Applications. Cambridge University Press, Cambridge, 1994.

2] J.W. McDonald, J.W. Snith and J.J. Forster, "Markov chain Monte Carlo exact inference for social networks", Social Networks, 29, pp. 127-136, 2007.

3] P. Sakar and A.W. Moore, "Dynamic social network analysis using latent space models", SIGKDD Explorations, 7(2), pp. 31-40, 2005

4] J.H. Koskinen and T.A.B. Snijders, "Bayesian inference for dynamic social network data." Journal of Statistical Planning and Inference, 137, pp. 3930-3938, 2007.

5] T. Kamada and S. Kawai, "An algorithm for drawing general undirected graphs". Information Processing Letters, 31, pp. 7-15 1989.

6] T.M.J. Fruchterman and E.M. Reingold, "Graph Drawing by Force-

Directed Placement." Software: Practice and Experience, 21(11), 1991.

7] P.D. Hoff, A.E. Raftery, M. S. Handcock, "Latent Space Approaches to Social Network Analysis", Journal of the American Statistical Association, Vol. 97, 2002

More Notes on Abstract Algebra and Logic: Towards their Application to Cell Biology and Security

Paolo Dini
Department of Media and Communications
London School of Economics and Political Science
London, United Kingdom
e-mail: p.dini@lse.ac.uk

Daniel Schreckling
Security in Distributed Systems
University of Hamburg
Hamburg, Germany
e-mail: daniel.schreckling@informatik.uni-hamburg.de

***Abstract*— This paper continues to chart and critically analyse the formal connections between algebra, logic, and cell biology on the one hand, and algebra, logic, and software security on the other. Much of the discussion is necessarily conceptual. Where the discussion is more formal the current distance between these disciplines appears evident. The paper begins with an introduction to abstract algebra that is meant to be accessible to interested non-mathematicians. It then discusses the algebra of network coding since it provides a practical example of some of the more abstract algebraic concepts. This is followed by an introduction to logic and algebraic logic for interested non-logicians. Specific parallels between algebra and logic are highlighted. The discussion then turns to the more advanced temporal logics and their application to software security. The paper concludes with a brief discussion of recent work in the application of algebra and logic to the DNA code and its potential relevance to interaction computing.**

I. Introduction

In this article we wish to begin to discuss some ideas related to the connections between cell biology and software security.[1] The bridge that we are in the process of building between these two very different fields relies on abstract algebra and logic and can be simplistically depicted as follows:

$$\text{biology} \rightarrow \text{algebra} \rightarrow \text{logic} \rightarrow \text{security}$$

In this context security represents one of the possible applications of a formalism that we expect to be of wider relevance. When referring to biologically inspired computing, reliance on some kind of evolutionary framework tends to be assumed by default. Whereas biological evolution does represent an essential model for biologically inspired computing, in this paper we are focussing on the 'other' function of DNA. By this we mean all the processes relating to the life of the individual organism, thus a better name could be 'development', or 'morphogenesis', or 'gene expression'. As Stuart Kauffman says,

> Darwin's answer to the sources of the order we see all around us is overwhelmingly an appeal to a single singular force: natural selection. It is this single-force view which I believe to be inadequate, for it fails to notice, fails to stress, fails to incorporate the possibility that simple and complex systems exhibit order spontaneously. [3]

Spontaneous construction of order in biology happens at different time scales. The slowest process is through evolution

[1]This article builds on the BIONETS deliverable [1] and on the paper [2]

Process	Mechanism	Result	Time Scale
Evolution	genetic operators natural selection	phylogenetic trees	millions of years
Morphogenesis	gene expression cell differentiation	adult organism	years
Metabolism	gene expression	structure and behaviour	days-hours-minutes
Cognition	neuronal network	thought	msec

TABLE I
CONSTRUCTION OF ORDER IN BIOLOGY

by natural selection, so in evolution order is constructed across many generations, in response to a selection pressure. During the life of the individual morphogenesis relates to the growth period from the embryo to the adult. The processes that run the metabolism of the individual take place on the order of days, hours, or minutes. Finally, our mental processes, which are far from understood, take place on the order of microseconds to seconds. Table I summarises these facts for a species whose individuals live for a few decades.

Thus, if evolution is the model that is able to explain phylogeny (a succession of organic forms sequentially generated by reproductive relationships), in this paper we are addressing the construction of a model that may eventually be relevant to ontogeny (the history of structural changes in a particular living being) [4]. The former relates to the first row of the table, while the latter to the second and third rows. Before we can begin to understand and model morphogenesis, however, we need to understand gene expression, on which morphogenesis depends. The biology part of this paper is therefore focused entirely on the cell. Our objective is in fact to develop a model inspired by cell metabolic processes that can represent equally well biological and computing processes. The motivation for studying gene expression over evolution is that it is a much faster and more powerful process of self-organisation.

As discussed in Section IV and in [4] [5] [6] [7], although we can make the mapping from cell metabolic cycles to digital algorithms seem plausible, we still face the problem of the absence of physical interaction forces between digital entities, and of the concept of temperature. In other words, we cannot rely on the minimisation of free energy as the driver of software systems. More importantly, the interaction forces bring with them a built-in regularity that is a direct consequence of the regularity and universality of physical laws and that, it seems plausible to conclude, gives rise to the

observed regularities in structure and behaviour of biological systems. In other words, not only do bio-chemical systems approach equilibrium spontaneously due to these interaction forces, but the manner in which they do so is constrained by the characteristics of physical law and by the spatio-geometrical properties of the constituent elements of matter to behave in certain ways and not others. For example, the six-fold symmetry of the snowflake and its predominantly flat shape are a consequence of the minimum energy configuration of the initial ice crystal 'seed'. This is provided by 7 water molecules, and happens to be a co-planar configuration, as can be easily demonstrated by six identical coins surrounding another identical coin at their centre. Having accounted for all the actors that participate in the spontaneous construction of order of biological systems, by elimination the culprit must be the interplay between physical law and the spatio-geometrical characteristics of matter, in the presence of an energy flow through an open non-equilibrium system.

Our research is based on the assumption that the regularities that result from this interplay can be formalised through the mathematical theory of groups, which is a branch of abstract algebra. Parallel work by [8] supports this perspective and is in fact ahead of our results. If we then identify the flow of energy with a flow of information we do not really need to worry about the lack of interaction forces. The behaviour of the users of the software and communication system will provide a constant flow of information which, in our view, can be constrained by algebraic transformation and interaction rules to produce ordered structures and behaviour. In reference to Fig. 1, the arrow between algebra and cell biology signifies the formalisation through algebra of DNA structure and interaction behaviour. We should emphasise that whereas algebra is seen as *necessary* it is not necessarily *sufficient*: we also need to account for very complex dynamical and interactive behaviour between different threads of activity, which is why this article addresses also temporal logics.

Our work is therefore based on the following argument [5] [4] [6] [7], which can be taken as a starting position for the research discussed in this article:

- The self-organisation exhibited by biological systems is driven by interaction forces and entropy maximisation (minimisation of free energy).
- The order, symmetry, and regularities exhibited by biological systems, furthermore, are a consequence of the regularities and symmetries in the underlying physical laws.
- In the absence of interaction forces or of the concept of temperature in digital systems, Kauffman's view of self-organisation (which is complementary to Evolution) may only be realisable in digital systems through the imposition of artificial constraints that embody a structure analogous to that caused by physical interaction laws in biological systems.
- Over the past 3000 years, abstract algebra developed largely as an effort to formalise in the most general way possible the regularities that we perceive in the world around us. It therefore seems like a good starting point for the task at hand.
- The importance and effectiveness of abstract algebra, symmetries, fields, and groups is demonstrated well by network coding techniques, which will therefore be discussed in this paper as a useful example of the theory.

More recently, the realisation that logic is also strongly related to algebra has strengthened our conviction that the direction we are working in is very interesting for the agenda of developing an effective theory of biologically-inspired computing.

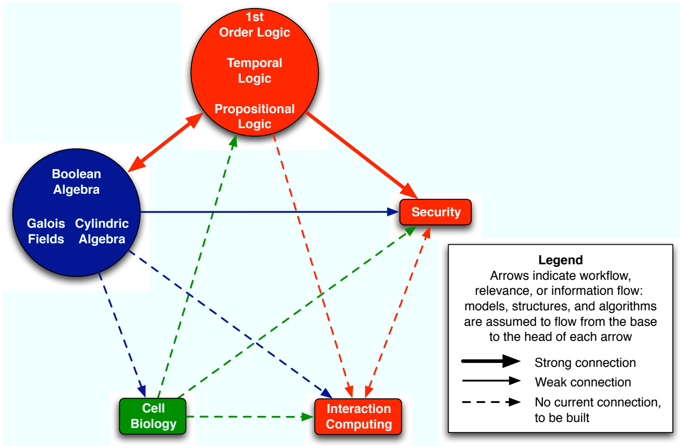


Fig. 1. **Disciplinary connections of relevance to bio-inspired security**

The arrow from cell biology to Interaction Computing is more difficult to explain, partly because the concept of interaction computing is still being formed [6] [7]. In order for the scenario described above to function, the digital system needs to become *reactive* to the inputs and the behaviour of the users. In other words, there cannot be computation without interaction. Iterating this concept recursively to components that are farther removed from the user interface, they too cannot change their states without being 'pushed' by the components that precede them. The picture that emerges can therefore be characterised conceptually as a set of coupled and interacting finite state machines whose state spaces are subdivided into permissible regions bounded by surfaces defined by algebraic structure laws. Each state machine is performing a piece of the algorithm. This is thus what we mean by interaction computing for the execution of a distributed algorithm[2] that is partly resident in the 'DNA' of the digital system and partly in the environment.

Since Boole published his short book on the mathematics of logic in 1847 [10], algebra and logic have been closely connected. The algebraisation of logic has been one of the success stories of the 20th Century [11] [12] [13]. Therefore, we do not expect to say anything fundamentally new in this regard, in this paper. However, if we take the interface between algebra and logic to be the 'centre of gravity' of this article, we are interested in shedding some light in two different directions. On the one hand, in Section III we will review the mapping of propositional logic to Boolean algebra, briefly explain the algebraisation of First-Order Logic, and finally show the link between First-Order and Temporal Logic, which is widely used in the security domain to support applications that are able to describe, detect, and prevent specific security features or breaches, respectively. On the other hand, after reviewing the fundamentals of abstract algebra, touching on coding and Galois theory, in Section IV we will discuss

[2]The term 'distributed algorithm' has been used extensively in the theory of timed automata, e.g. [9], which is one of the research topics we are also planning to pursue, in connection with temporal logics.

briefly how physical and evolutionary forces have induced an algebraic structure on the DNA code. We see this as the first step in the charting of a possible path from order construction in biology to autonomic security algorithms and data structures that weaves its way through algebra and logic.

II. Abstract Algebra

In this section we will present the very basics of abstract algebra, which is relevant to coding theory and Galois theory. The objective is two-fold, i.e. to show how the structure of algebra resembles to some extent the structure of logic, to be presented in the next chapter, and to provide a vocabulary and formalism that will at a later date be applied to biology. In this short article we will not attempt to review and define an isomorphic map between the parts of algebra and logic that are specifically relevant to biology and security, mainly because this is quite a broad question that will take much more work to address. We will also not attempt to 'solve' any cell biology problems. The algebra discussed in this section is quite elementary because we think it is useful to have self-contained discussion of the concepts alongside the review of logic and algebraic logic presented in Section III of this article. Our purpose is to take the first tentative steps in the direction of an integrative formalism.

A. Starting Concepts

The language of abstract algebra is set theory. Since it is easy to read and understand for most people, it is used here. Abstract algebra deals with the operations between the elements of a set, with the mappings between different sets, and with mappings defined from a set to itself. Different kinds of operations and mappings give rise to different relationships between these constitutive parts of algebra, a fact which is partly responsible for associating a *structure* with algebra. The elements of a set can be anything at all, including other sets. They can be integers, polynomials, functions, 'variables', and so forth. What Boole realised is that they could also be the constitutive elements of logic, which provides one of the fundamental motivations for the line of investigation pursued in this article. Finally, we are making an explicit assignment of some yet unspecified elements of the cell as elements of a set. The temptation is to treat the cell itself as a set. Although this is not necessarily wrong, we are starting with much simpler sets such as, for example, treating the four bases of the DNA as the elements of a set that we can call the alphabet of DNA. Another example could be the 20 amino-acids from which all proteins are built.

Algebra has successfully identified and formalised the fact that certain sets, although apparently different on the surface, exhibit identical behaviour in how their elements relate to or combine with each other. This motivates the definition of abstract 'objects' that model the structure and/or behaviour of sets that recur again and again in countless applications. We will now define these basic building blocks, relying almost exclusively on examples from various sets of numbers (the integers $\mathbb{Z}$, the real numbers $\mathbb{R}$, etc) and of polynomials. The following statements can be found in any algebra book, but we are borrowing freely mainly from [14] [15] [16].

A **binary relation** is a statement which, for any two elements of a set, is either true or false for that pair. Another way to put it is that a binary relation R on a set A is a subset of the Cartesian product $A \times A$. For example, if $A = \{1, 2, 3\}$, then the relation 'less than' on A is the set $\{(1, 2), (1, 3), (2, 3)\}$.

Let R be a binary relation on A. An **equivalence relation** is a binary relation that satisfies these three conditions:

- (Reflexive) $(a, a) \in R, \forall a \in A$
- (Symmetric) If $(a, b) \in R,$ then $(b, a) \in R$
- (Transitive) If $(a, b) \in R$ and $(b, c) \in R,$ then $(a, c) \in R$

The **equivalence class** of the element $a \in A$ is the set

$$\{b \in A \ : \ (a, b) \in R\}$$

In the example above, the relation 'less than' fails all three properties, so it is not an equivalence relation. An important fact that is not necessarily obvious is that the set of equivalence classes of a relation R on a set A is a **partition** of A. In other words, the equivalence classes do not intersect and, together, they cover all of A.

B. Groups, Rings and Fields

Before we can move forward we need to define three of the abstract objects, or types of set, that were indicated above: groups, rings and fields.

A **group** is a set G of elements with a binary operation between them $\circ$ such that:

- (G0- Closure) $\forall\ g,\ h \in G,\ g \circ h \in G$
- (G1- Associative) $g \circ (h \circ k) = (g \circ h) \circ k,\ \forall\ (g,\ h,\ k) \in G$
- (G2- Identity) $\exists\ e \in G \colon g \circ e = e \circ g = g,\ \forall g \in G$
- (G3- Inverse) $\forall\ g \in G,\ \exists\ h \in G \colon\ g \circ h = h \circ g = e$
- (G4- Commutative) $g \circ h = h \circ g,\ \forall (g,\ h) \in G$

The last condition is satisfied only by so-called abelian or commutative groups.

A **ring** R is a set with two operations, that are usually called 'addition' and 'multiplication', such that:

- (A0- Closure) $\forall\ a,\ b \in R,\ a + b \in R$
- (A1- Associative) $a+(b+c) = (a+b)+c,\ \forall\ (a,\ b,\ c) \in R$
- (A2- Zero) $\exists\ 0 \in R \colon a + 0 = 0 + a = a,\ \forall a \in R$
- (A3- Inverse) $\forall\ a \in R,\ \exists\ b \in R \colon\ a + b = b + a = 0$
- (A4- Commutative) $a + b = b + a,\ \forall (a,\ b) \in R$
- (M0- Closure) $\forall\ a,\ b \in R,\ ab \in R$
- (M1- Associative) $a(bc) = (ab)c,\ \forall\ (a,\ b,\ c) \in R$
- (D- Distributive) $(a + b)c = ac + bc,\ c(a + b) = ca + cb \forall\ (a,\ b,\ c) \in R$

Rings can have additional structure if they satisfy one or more of these additional axioms:

- (M2- Identity) $\exists\ 1 \in R (1 \neq 0) \colon a1 = 1a = a,\ \forall a \in R$
- (M3- Inverse) $\forall\ a \in R (a \neq 0),\ \exists\ b \in R \colon\ ab = ba = 1$
- (M4- Commutative) $ab = ba,\ \forall (a,\ b) \in R$

A ring that satisfies M2 is called a **ring with identity**. If it satisfies M2 and M3 it is called a **division ring**. A ring that satisfies the first 8 axioms plus M4 is a **commutative ring**. A ring that satisfies *all* the axioms, i.e. a commutative division ring with identity, is a **field**. Subrings and subfields are subsets of rings or fields that are themselves rings or fields.

In preparation for the next section on logic, it is interesting to represent the field $\{0, 1\}$ using the language of truth tables, as shown in Table II.

		Operations	
		+	·
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

TABLE II

TRUTH TABLE FOR THE $\mathbb{Z}_2$ FIELD

C. Cosets and Homomorphisms

Now we are going to start building up some more complicated concepts using the basics defined so far. It is probably fair to say that the previous section is a useful reference that we can go back to when the derivations below get confusing, whereas Section II.A will be used right away.

The presentation of this material is necessarily abstract and a little dry. We will try to compensate for this by spending extra effort in explaining the concepts thoroughly, whenever possible, since most of the concepts are extremely interesting. Given also their great depth, however, in most cases we will not be able to do justice to the beauty and broad relevance of the theory. We will use numbers and polynomials as motivating examples, keeping an eye out for connections with logic, which is presented in the next chapter, and with biology, which is presented in the last chapter.

A particular kind of equivalence class that we are going to need is the coset. To define a coset we need a ring R and one of its subrings, S. What we say is that we partition R into the cosets of S. The 'co' in 'coset' stands for 'complementary'; therefore, S itself is *one of* the cosets. For this to work out we can't just take a random subset of R and call it S. The fact that the cosets partition (are disjoint and cover) R tells us that each coset could be an equivalence class. In fact, this is precisely how they are defined: a **coset** of S (or S itself) is the equivalence class of a relation E on the ring R that satisfies the rule

$$(a, b) \in E \text{ if } b - a \in S, \quad a, b \in R \tag{2.1}$$

Written in this way, this definition seems to imply that S must be known in advance. In fact, we just need to know a characteristic of S that allows us to construct it by 'filtering' the elements of R.

> For example, take as a coset the equivalence class of the binary relation 'equality mod 5, with remainder 0' on the ring $R = \mathbb{Z}$. An algorithm to construct this coset involves picking two integers a and b and asking: does $a \bmod 5 = b \bmod 5 = 0$? If yes, a and b belong to S, which is clearly the infinite set
>
> $$S = \{..., -15, -10, -5, 0, 5, 10, 15, ...\}. \tag{2.2}$$

We notice that the difference between any two members of this set is indeed another member of this set, but we did not use this fact to construct S in this example. This shows how we already know something quite fundamental about S before constructing it, which is an example of what we mean by algebraic structure.

If we look at all the other equivalence classes that we could build from the binary relation 'mod 5' we will construct an important object, $\mathbb{Z}_5$. Allenby [15] defines it in general as follows:

> $\mathbb{Z}_n$ is the set of equivalence classes determined by the equivalence relation '= (mod n)'.

So it is a set of sets, each of which looks similar to Eq (2.2). Because this is hard to write down, we define a new concept, the **coset representative**, as an element of a coset that is convenient to use because it makes it easy to recognise what coset it refers to. For example, the most convenient coset representative of Eq (2.2) is 0. The set of sets $\mathbb{Z}_5$, therefore, can be conveniently written down as

$$\mathbb{Z}_5 = \{0, 1, 2, 3, 4\}, \tag{2.3}$$

Each of these numbers 'points' to an infinite subset of $\mathbb{Z}$. These subsets of $\mathbb{Z}$ are the 5 cosets referred to as $\mathbb{Z}_5$, and they partition $\mathbb{Z}$. A useful shorthand to refer to each coset is to notice that S can be identified with the 0 coset representative, and can be related to the others as follows:

$$\begin{aligned} S &= \{..., -15, -10, -5, 0, 5, 10, 15, ...\} \to 0 \\ S+1 &= \{..., -14, -9, -4, 1, 6, 11, 16, ...\} \to 1 \\ S+2 &= \{..., -13, -8, -3, 2, 7, 12, 17, ...\} \to 2 \\ S+3 &= \{..., -12, -7, -2, 3, 8, 13, 18, ...\} \to 3 \\ S+4 &= \{..., -11, -6, -1, 4, 9, 14, 19, ...\} \to 4 \end{aligned} \tag{2.4}$$

After all this work, which has enabled us to arrive at a notation that we will use again later, we can recognise this strange set of sets $\mathbb{Z}_n$ as simply the set of remainders when $\mathbb{Z}$ is divided by a particular integer n. This particular 'mechanism' will be an essential intuitive and mnemonic device when we start working with polynomials.

We can now gear up to define homomorphisms. First, Fig. 2 provides a quick graphical reminder of the basic types of mappings. Homomorphisms are mappings that can be defined between algebraic objects of the same type. We will focus on rings. A **ring homomorphism** is a map

$$\theta \colon \langle R, +, \cdot \rangle \;\to\; \langle S, \oplus, \odot \rangle \tag{2.5}$$

between rings that satisfies the following conditions for any $a, b \in R$:

$$\theta(a + b) = \theta(a) \oplus \theta(b) \tag{2.6a}$$

$$\theta(a \cdot b) = \theta(a) \odot \theta(b) \tag{2.6b}$$

This is a fairly abstract set of conditions since the elements of rings, as we said, can be anything and addition or multiplication can likewise represent more general binary operations between the elements of a ring than their familiar arithmetical interpretations. A homomorphism does not have to be 1-1 and onto. If it *is* 1-1 and onto, then it is an **isomorphism**. To gain an intuitive understanding on how constraining an homomorphism is, we are going to use an example that is actually an isomorphism, simply because it is familiar and easier to grasp.

> Example. If the map is from $\mathbb{Z}$ to itself and is given by
>
> $$y(x) = \frac{1}{2}x,$$
>
> then we can see that
>
> $$y(2+4) = 3 = 1 + 2 = y(2) + y(4)$$
> $$y(2 \cdot 4) = 4 \neq 1 \cdot 2 = y(2) \cdot y(4).$$

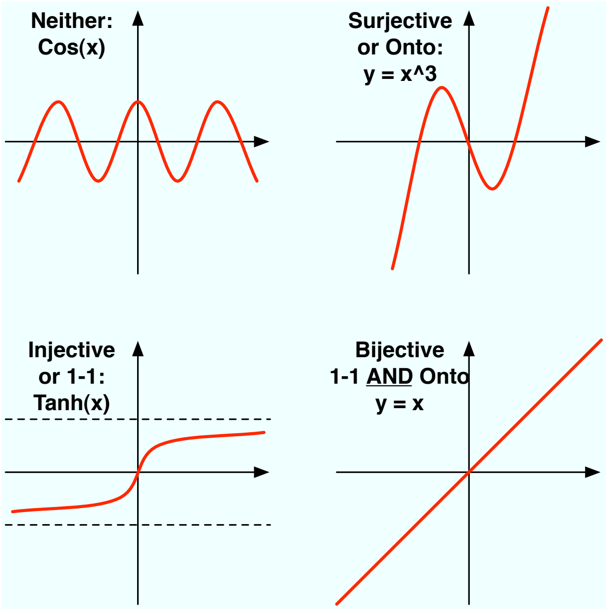


Fig. 2. **The four basic kinds of mappings**

Remarkably, the simple straight line equation $y = (1/2)x$ is not a homomorphism (and therefore it is not an isomorphism either)! To get a homomorphism we need to use $y(x) = x$. Because $y(x) = x$ is 1-1 and onto, it is also an isomorphism. In fact, an isomorphism from a ring to itself is called an **automorphism**, which in less technical terminology can be called a **symmetry**.

D. Kernel, Image, Ideals and Factor Rings

The above example has shown us that even if homomorphisms don't need to be 1-1 or onto, they are still fairly restrictive maps. Later we will have to come back to this discussion with an example that deals with somewhat more complicated elements than the integers. For now we can proceed with a couple of Venn diagrams, taken from Allenby.

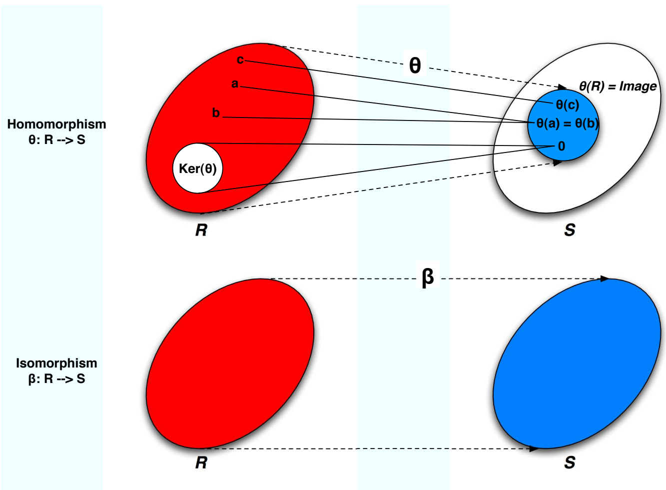


Fig. 3. **Showing how homorphisms are not 1-1 and Onto**

Fig. 3 shows two new objects, the kernel and the image of a homomorphism. The **kernel** of a ring homomorphism $\theta\colon R \to S$ is a subring of R defined by

$$\mathrm{Ker}(\theta) = \{k \in R\colon \theta(k) = 0\}. \tag{2.7}$$

The **image** is a subring of S defined by

$$\mathrm{Im}(\theta) = \{s \in S\colon s = \theta(r) \text{ for some } r \in R\}. \tag{2.8}$$

Allenby ([15], 142) discusses how homomorphisms have been likened to photographs in the algebra literature, a metaphor that is vaguely reminiscent of Plato's cave. The set R is the object being photographed, and the image $\theta(R)$ is the photograph, which is a subset of the 'real thing'. Furthermore, the kernel is the set of those points that lie behind the object relative to the camera, and that therefore do not appear at all in the photograph. In this sense they are being mapped to 'zero'.

There is a generalisation of the kernel that we are going to need. An **ideal** of a ring R is a subring I of R such that, for any $i \in I$ and $r \in R$, $ri, ir \in I$. To decide whether a subring is an ideal we should test this condition plus all the ring axioms. However, there is an equivalent and shorter 'ideal test'. A non-empty subset I of a ring R is an ideal of R iff

$$(a)\ \forall\ i_1, i_2 \in I,\ \ i_1 - i_2 \in I \tag{2.9a}$$
$$(b)\ \forall\ i \in I \text{ and } r \in R,\ \ ri, ir \in I. \tag{2.9b}$$

On the basis of this we can say that the kernel of a homomorphism $\theta\colon R \to S$ is an ideal of R. For example, all the subrings of $\mathbb{Z}$ of the form $n\mathbb{Z}$ are ideals. We can now see how the first of the sets in Eq. (2.4) is actually an ideal, which means that we can refer to the various cosets as $I, I+1, I+2$, etc. The 'set of sets' that we keep referring to is so important that it is given a special name, the factor ring. Additionally, its definition is made quite general, as follows.

Let I be an ideal of the ring R. The **factor ring** R/I is the set of cosets of I in R, with operations of addition and multiplication defined as

$$(I + x) + (I + y) = I + (x + y) \tag{2.10a}$$
$$(I + x)(I + y) = I + xy \tag{2.10b}$$

E. Fields

The research discussed in this article is not in mathematics, but in the application of algebra to logic, biology, and new forms and models of computing. Proofs are presented in an informal way when they can be particularly helpful for communicating the most important ideas, but much else is taken for granted. We are just covering the smallest possible number of essential concepts in order to provide a basis for coding theory and Galois theory that can expose the underlying algebraic machinery. We will endeavour to show how this algebraic machinery will then be relevant to biology, logic, and security. Furthermore, we hope that it will help us develop a model for interactive computing that should enable the replication of biological behaviour in software and in security applications in particular. Fields are the next algebraic structure that will help us in this direction.

Whereas the 'ring archetype' is the set of integers $\mathbb{Z}$, the 'field archetype' is the set of rational numbers, $\mathbb{Q}$. We saw in Section 2.2 that fields are more constrained than rings. In fact they are so constrained that there are only two methods to construct a field. The first method essentially replicates the construction of $\mathbb{Q}$ from $\mathbb{Z}$ and results in a field that is, loosely

speaking, *larger* than the ring it is based on. We proceed with a few definitions taken from [14].

A **zero divisor** in a ring R is a non-zero element $a \in R$: $\exists(b \neq 0) \in R$ with $ab = 0$. An **integral domain** is a commutative ring with identity that has no zero-divisors (e.g. $\mathbb{Z}$). Let R be an integral domain. A field F is a 'field of fractions' of R if

(a) R is a subring of F

(b) Any element of F can be written in the form ab^{-1} for some $a, b \in R$

The result of this kind of construction is an infinite field.

The second method generalises the construction of the integers modulo n from the integers, that is, the field is a factor ring R/I, and is *smaller* than the ring it is based on. Some conditions need to be satisfied for this to be true, but in the examples we will discuss it will be true, so we skip the finer points here. The result of this kind of construction is a finite field.

Something we have not yet stated is that, for factor rings generated as a set of integers modulo n, n must be a prime number for the factor ring to be also a field. More generally, Galois proved that any and all finite fields have order a prime power. This means that the number of elements of a finite field can **only** be either a prime (exponent = 1) or a prime raised to a positive integer power. For example, finite fields of size 5, or 25, or 256 exist, whereas no finite fields of size 18, or 30, or 100 exist. The essential part of the proof, which we skip here, is that in the former case each element has an inverse, whereas in the latter case it does not. It is also important to realise that, if the field is a prime power where the power is greater than 1, then such a field cannot be composed by integers as its elements. As we will soon see, in such cases its elements are more complex and can be represented as vectors, or as polynomials. Finite fields are often called Galois fields in honour of their discoverer. Thus, $\mathbb{Z}_5$ can also be called $GF(5)$ and $\mathbb{Z}_2$ is interchangeable with $GF(2)$. To denote a finite field whose size is a prime power, say m, we say $GF(2^m)$. Such fields, where $m > 1$, are called field extensions.

F. Field Extensions

We have now reached the central topic of interest in this chapter, field extensions. Field extensions are useful to give us a way to find the roots of polynomials. The kinds of problems we are interested in deal with finite fields and with polynomials defined over finite fields (although the theory can handle any kind of field). This means, for example, that if the finite field in question is $F = \{0, 1, 2, 3, 4\}$, the polynomials we can build have coefficients that can be taken *only* from this set of integers. Such polynomials are not constrained in their degree, they can be of arbitrarily high degree. The (infinite) set of all such polynomials is not a field, it is a ring with identity, denoted by $F[x]$. Since a ring needs to satisfy the closure condition under addition and multiplication, when any two polynomials are added or multiplied together the arithmetic operations on their coefficients are performed modulo (in this case) 5. This ensures closure with respect to F. If a polynomial defined in this way has no roots in the field over which it is defined, it is called **irreducible**. Polynomials whose leading coefficient (coefficient of the highest power of x) is 1 are called **monic**.

We finally come to the main theorem of interest, due to Kronecker. Let F be any field and let $f(x) = a_0 + a_1x + a_2x^2 + ... + a_mx^m \in F[x]$ be an irreducible polynomial over F. Then $\exists$ a field S containing F as a subfield such that $f(x)$ has a root in S. S is the extension field, F is the base field.

Surprising as it may seem, finding such a root is *not* our main goal! We will show why the above theorem is true by relying on most of the concepts discussed so far, which is in itself quite interesting. However, in the process we will discover that the roots of $f(x)$ in the extension field have a very unexpected representation, which makes them look rather useless. Our state of confusion will be rescued by a method that shows how the algegraic structures in the extension field (the roots) are related to the algebraic structures in the polynomial ring from which $f(x)$ is taken [17]. This relationship sheds light on network coding techniques, highlighting how they too do not need to find any roots explicitly. In the next section we will explore Gordon's technique as it makes it relatively easy to verify the abstract theory presented here with simpler and more applied examples. We will not have time or space in this article to delve into Galois theory, which requires another level of abstraction and a firm grasp of groups, in addition to rings and fields. The results we will discuss, however, should already provide a sufficiently rich context to begin thinking about how to map them to security through logic, and to interaction computing through biology.

Our first step is to note that we can define an infinite number of ideals as subrings of $F[x]$. Each ideal is generated by any one of the polynomials in the ring. For simplicity let's assume that we are dealing only with irreducible polynomials over F. We can take F to be the field $GF(2) = \{0, 1\}$ without loss of generality. In other words, to prepare ourselves for the network coding examples and the application to information systems in general we can use the binary 'alphabet' as our base or ground field.

Kronecker's theorem is proven by first establishing the following. Given a ring R and an ideal I of R, it is always possible to find a ring S and a surjective map $\theta\colon R \to S$ such that the ideal is mapped to 0 in S (i.e. $\text{Ker}(\theta) = I$). This means that the polynomial of interest (and all its possible multiples taken from $F[x]$) is mapped to 0 in S. This is quite ingenious since, if such a map can be found, then the variable x in R necessarily maps to the root of the polynomial of interest when they are both expressed in S.

The ideal generated by a polynomial $f(x)$ is the set of the possible products of all the elements of $F[x]$ with $f(x)$, and is denoted by $[f(x)]$. It turns out that the ring S is nothing more than the factor ring

$$S = \frac{R}{I} = \frac{F[x]}{[f(x)]}, \tag{2.11}$$

which is (conceptually) built by dividing all the polynomials in $F[x]$ by the polynomial $f(x)$ and retaining only the remainders. Each different remainder represents a different coset representative for a different coset of R. S is simply the set of these cosets. How many representatives are there? How big is S? It turns out it's not that big (when we list only the coset representatives, which is the normal way to represent it).

For example, if the irreducible polynomial in question is of degree 6, the remainders can, by definition,

be of degree 5 at most:

$$a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4 + a_5 x^5. \quad (2.12)$$

Because the coefficients are taken from $\mathbb{Z}_2 = GF(2)$ this is equivalent to a binary string of 6 bits. Thus, there is a total of $2^6 = 64$ elements in S for this example.

In other words, the required map is division of $F[x]$ by $f(x)$, modulo $f(x)$. Because this maps $F[x]$ to the ring of remainders obtained by dividing by $f(x)$, any multiple of $f(x)$ will give 0 remainder. But any multiple of $f(x)$ belongs to the ideal $I = [f(x)]$ generated by $f(x)$. Therefore, I is mapped to the 0 element of S. Therefore, Ker$(\theta) = I$ as required.

We can now prove Kronecker's theorem. This is done in two steps: (1) we need to show that $S = R/I$ is a field; (2) we need to show that S contains a root of $f(x)$. To prove the first part we first note that each element of S is usually written as

$$I + r(x), \quad (2.13)$$

where $r(x)$ is the remainder we have been talking about. Now suppose that $f(x)$ is irreducible over the ground field,

$$f(x) = b_0 + b_1 x + b_2 x^2 + ... + b_m x^m, \quad (2.14)$$

and that $I + r(x)$ is a non-zero element of S. Because $f(x)$ is irreducible and $r(x)$ is a remainder upon division by $f(x)$, necessarily the Greatest Common Divisor between $r(x)$ and $f(x)$ in $F[x]$ is 1: $GCD(r(x), f(x)) = 1$. One of the algebra basics we glossed over is the result that, in $F[x]$, $\exists\, s(x), t(x)$:

$$s(x)r(x) + t(x)f(x) = GCD.$$

Therefore,

$$s(x)r(x) + t(x)f(x) = 1 \qquad \text{in } F[x]. \quad (2.15)$$

Now the element 1 of R is mapped to the element $1 + I$ of S because clearly the number 1 can only be a remainder in a division by $f(x)$. Therefore, Eq. (2.15) becomes, in S,

$$s(x)r(x) + t(x)f(x) + I = 1 + I \qquad \text{in } S.$$

By the rules by which ideals are added and multiplied (Eq. (2.10)),

$$\begin{aligned} &[s(x)r(x) + I] \oplus [t(x)f(x) + I] = 1 + I && \text{in } S, \\ &[s(x) + I] \odot [r(x) + I] \oplus [t(x) + I] \odot [f(x) + I] \\ &\qquad\qquad = 1 + I && \text{in } S. \end{aligned}$$

But, by the definition of ideal,

$$f(x) + I = I = 0 \qquad \text{in } S!$$

As a consequence,

$$[s(x) + I] \odot [r(x) + I] = 1 + I \qquad \text{in } S, \quad (2.16)$$

which means that the element $[r(x) + I]$ has an inverse in S:

$$[r(x) + I]^{-1} = [s(x) + I] \qquad \text{in } S. \quad (2.17)$$

Hence, S is a field. Note that if $f(x)$ were not irreducible the GCD would not in general be 1, and the factor ring would not be a field because we could not prove the existence of an inverse for each of its elements. This is the exact analogue to the situation in $\mathbb{Z}$ where $\mathbb{Z}_p$ is a field only if p is prime.

Now we need to show that S contains a root of $f(x)$. To do this it is helpful to refer to Fig. 4, which is also taken from Allenby. In the figure we can see how the original field F is a subset of $F[x]$ since b_0 in Eq (2.14) spans over (can take on the value of) every element of F; of course this refers to those polynomials for which b_1 and all the higher coefficients are 0, i.e. to constant polynomials. The map θ then involves dividing all the polynomials in $F[x]$ by a particular polynomial $f(x)$, keeping only the remainders. Clearly, all the constant polynomials can only be remainders. Therefore, the field F is identically reconstructed in the factor ring S (which we have now established is a field). This is shown as $\theta(F) = F$ in the figure.

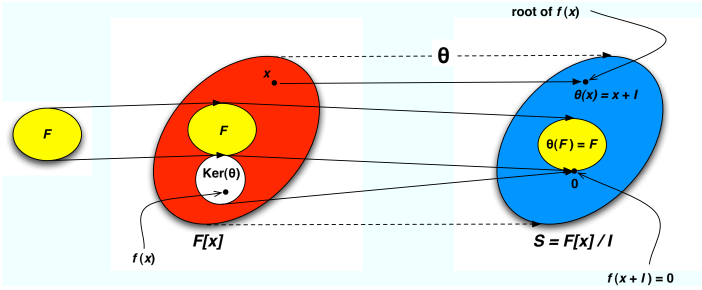


Fig. 4. **Finding the root of** $f(x)$

The final step in the proof is almost unremarkable and one struggles to see its significance at first. We notice that we have constructed the field S in such a way as to make $f(x)$ and its ideal map to 0. Then, necessarily, the variable x must map to the root of $f(x)$, which we will call α. How can we be sure? We know that x is of degree smaller than $f(x)$ because, if $f(x)$ were a first-order polynomial then its root would lie in F and we would have no need for all this work. So $f(x)$ is of degree 2 or higher (and it is irreducible in F). Because x is of degree smaller than $f(x)$ it is one of the remainders, and therefore it is an element of S:

$$\theta(x) = x + I = \alpha. \quad (2.18)$$

To prove that it is indeed a root of $f(x)$ we substitute it into Eq (2.14):

$$\begin{aligned} f(\alpha) &= b_0 + b_1\alpha + b_2\alpha^2 + ... + b_m\alpha^m \\ &= b_0 + b_1(x + I) + b_2(x + I)^2 + ... + b_m(x + I)^m \\ &= b_0 + (b_1 x + I) + (b_2 x^2 + I) + ... + (b_m x^m + I) \\ &= I + (b_0 + b_1 x + b_2 x^2 + ... + b_m x^m) \\ &= I + f(x) \\ &= I \\ &= 0 \end{aligned} \quad (2.19)$$

Here is where we notice for the first time that we have 'found' the root of $f(x)$ without having any idea at all about its numerical value. We understand what it is, but we do not really know it! Our understanding appears to be quite useless. To carry the point further, we have now found the extension field S but, whereas we have a very concrete idea of what F is, the nature of S escapes us. Be that as it may, we know that each element of S looks like Eq (2.13). Our only way forward is to map $r(x)$ to S the same way we have just done for $f(x)$, that is, substituting the image of x under θ into the

image of $r(x)$:

$$\begin{aligned}I + r(x) &= I + (a_0 + a_1x + a_2x^2 + ... + a_{m-1}x^{m-1}) \\ &= a_0 + (a_1x + I) + (a_2x^2 + I) + ... \\ &\quad + (a_{m-1}x^{m-1} + I) \\ &= a_0 + a_1(x + I) + a_2(x + I)^2 + ... \\ &\quad + a_{m-1}(x + I)^{m-1} \\ &= a_0 + a_1\alpha + a_2\alpha^2 + ... + a_{m-1} - \alpha^{m-1} \\ &= r(\alpha) \end{aligned} \tag{2.20}$$

The result tells us that each element of S is isomorphic to the remainders of $F[x]/[f(x)]$, but expressed in terms of the root we have 'found'. Interestingly, we have discovered quite a bit about the extension field of F without calculating any 'numbers'. The nature of the extension field S is in fact to be isomorphic to a vector space of dimension m (with 2^m elements) over the field F. In other words, since $F = GF(2)$, $S = GF(2^m)$. The different powers of the root α play the role of basis vectors. We recognise in this the same structure of the complex field $\mathbb{C}$, each of whose elements is of the form $a + ib$, where i is the root of the polynomial $x^2 + 1$, which is irreducible over the real field $\mathbb{R}$.

In the foregoing we have skipped a few technical details that we are going to need in the discussion that follows. Firstly, the process described above will work also for an $f(x)$ that is the product of smaller irreducible factors, meaning that each of its factors must necessarily be of degree 2 or higher. The process is simply applied to each of its factors in turn. As will be discussed more fully below, if it is applied to a polynomial that can be reduced into smaller factors S will not be a field but only a factor ring.

Secondly, it is helpful to introduce a special kind of irreducible polynomial, that we call **primitive** and that we can denote by $p(x)$ for convenience. A primitive polynomial will generate the whole of $GF(2^m)$ if its ideal $[p(x)]$ is used to generate the field $S = GF(q)[x]/[p(x)]$. This name is logical and matches the fact that a primitive element of a cyclic group generates the whole group through its successive powers. In the discussion above we took this for granted, but in fact this property does not apply to all irreducible polynomials. Irreducible polynomials that are not primitive and that are put through the process above will not generate the whole field extension, even though their roots will lie in the same extension. A polynomial of degree m over $GF(2)$ is primitive if the smallest value of n for which $p(x)$ divides $(x^n - 1)$ is $2^m - 1$ ([18], 44). Although this may seem a little arbitrary at first, hopefully the discussion that follows will show that it is actually quite natural.

Finally, another generic term for an irreducible polynomial (which may or may not be primitive) when discussing its roots in an extension field is **minimal polynomial** if its leading coefficient equals 1 (necessarily true over $GF(2)$).

G. Applications to Network Coding

Our purpose in this sub-section is not to summarise results that have been known for 40 years or more in order to apply them in the manner originally intended. For that, we need only provide bibliographical references. The point of developing all this algebra, from the point of view of our research in bio-inspired computing, is not functionalist (i.e. to improve performance) in the first instance; rather, it is to understand the algebraic structure itself and then see what this structure has in common with logic on the one hand and with the DNA on the other, as will be discussed in subsequent sections of this paper. There is of course an expectation that the biological insights will ultimately bring a number of advantages, including better performance, but the first objective is to develop a common formalism for these very different domains. To this end, network coding is a very useful 'case study' because it provides a very practical context against which the abstract concepts discussed in the previous sub-sections will hopefully become easier to understand. The following discussion relies on [18], [14], and [19].

We need to introduce some basic terminology of linear block codes, upon which we will build a powerful algebraic structure by progressively introducing additional constraints. We start with the notion of a vector space. Put simply (even if perhaps too simplistically), a **vector space** is a field in more than one dimension. Each element of a vector space of dimension n is a tuple of n objects, with each object an element of a field F. An n-dimensional vector space over F is denoted by F^n. If F is the binary field $\mathbb{Z}_2$, F^n has exactly 2^n elements.

Linear block codes operate by adding redundancy to a message in order to allow the original message to be reconstructed in the presence of transmission errors. Given a signal stream, we break it up into an unspecified number of message blocks of length k, taken from F^k. Thus, the message space is a vector space of dimension k and exactly 2^k elements (message blocks). To make the code unambiguous and invertible back to the original message, the mapping between the message space and the code space must be 1-1 and invertible. Therefore, the code space is a k-dimensional sub-space of F^n and also contains 2^k different elements, although each is of length $n > k$.[3] The positive consequence of setting things up in this manner is that the encoding map becomes a linear transformation (i.e. a matrix multiplication) ([14], 239). Thus, we multiply each message block (row vector) by a matrix of dimensions $(k \times n)$ (k rows by n columns). The result of this multiplication for each block is a code vector of length n, called a codeword. Each codeword is sent wirelessly by the transmitter and received by the receiver, possibly with some errors. The receiver must detect these errors, correct them, and then map each codeword back to the original message block to reconstruct the original signal.

Error detection is achieved by introducing the check matrix, which is another important plot line in the discussion. Drawing on elementary linear algebra, Theorem 3.4 in [18] states that, "For any $k \times n$ matrix G over $GF(q)$ with k linearly independent rows, there exists an $(n-k) \times n$ matrix H over $GF(q)$ with $(n-k)$ linearly independent rows such that for any row g_i in G and any h_j in H, $g_i \cdot h_j = 0$. The row space of G is the null (dual) space of H, and vice versa." Since the rows of G provide a basis for the code space, the same rule applies to any linear combination of its rows, i.e. to any codeword. Therefore, any error can be easily detected as a non-zero result when multiplying a received codeword by H^T. Appropriate error correction methods then can be applied, as discussed in

[3] This is analogous to a slanted 2D plane immersed in 3D space: each point on the plane is defined by 3 coordinates, but the plane itself remains a 2D subspace.

the literature. Fig. 5 provides a schematic summary of these concepts.

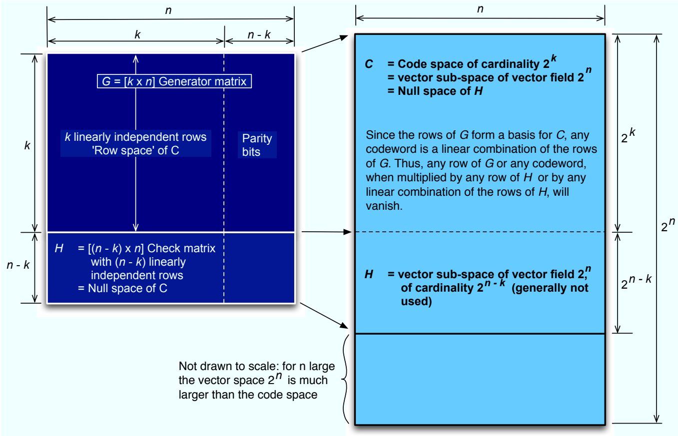


Fig. 5. **Schematic summary of linear block code**

We now start connecting to abstract algebra by noticing that in network coding applications the focus of attention is not a polynomial whose roots we need to find through the generation of a field extension as a factor ring (Galois's original problem) but, rather, the factor ring itself. Furthermore, the generation of a field extension is a necessary step in finding the roots of an *irreducible* polynomial. As it happens a polynomial does not need to be irreducible to form an ideal, and therefore it does not need to be irreducible to form a factor ring. An additional point that should help connect the work in the previous sub-section to this discussion is that the concept of ideal can be applied, in a sense, recursively. In other words, a factor ring is obtained by 'dividing' a ring $GF(2)[x]$ by an ideal, say $[(x^n - 1)]$. The resulting factor ring can also contain ideals. It is these latter ideals that we are mainly going to use. In the discussion of cyclic codes that follows we identify the vector space F^n with the factor ring, and the code space with one of its ideals (see Fig. 6).

A **cyclic code** is a linear block code with the characteristic that every cyclic shift of any of its codewords yields another codeword. A **cyclic shift** is the n-tuple obtained by shifting every component to the right (or left) and wrapping the last component back to the front of the list. When working with cyclic codes, it is useful to associate each codeword of length n with a polynomial of degree $n-1$ or less, whose coefficients are the coordinates of the code vector:

$$c(x) = c_0 + c_1 x + c_2 x^2 + ... + c_{n-1} x^{n-1}. \tag{2.21}$$

$c(x) \in F[x]$ is called the code polynomial. The set of 2^k code polynomials of a code C is a subset of the factor ring $GF(2)[x]/[(x^n - 1)]$, which is not a field since $(x^n - 1)$ is factorisable. Incidentally, from this fact we deduce that whereas every finite field contains a number of elements that is equal to an integer power of a prime number, the converse is not true: a set that contains a number of elements that is equal to an integer power of a prime number is not necessarily a field. To clarify, the factor ring $GF(2)[x]/[(x^n - 1)]$ is not a field because not every one of its elements has an inverse (due to the fact that the GCD of each element and $(x^n - 1)$ is not necessarily 1). From this point of view the factor ring is a one-dimensional object. But the same object can also be described as a 'vector field' of dimension n, F^n. From this second point of view the use of the term 'field' refers to the fact that the *ground* field $\{0, 1\}$ is indeed a field. We notice that in this factor ring, by construction, operations between its elements are performed modulo $(x^n - 1)$. The reason for creating the factor ring using $(x^n - 1)$, rather than an irreducible polynomial of the same degree, is that in the latter case the factor ring would be a field, and therefore its only ideals would be 0 and itself. Why this matters is shown next.

In the polynomial ring $GF(2)[x]/[(x^n - 1)]$ the cyclic shift of a codeword by k is equivalent to multiplication modulo $(x^n - 1)$ of the corresponding code polynomial by x^k. This fact in itself seems unremarkable until we realise that an arbitrary polynomial $a(x)$ can be considered as a linear combination of cyclic shifts which, according to the definition, yields another $c(x)$ that still belongs to the same code C. Because we already knew that C is a sub-space of F^n (or a subring of the factor ring), we know that it is closed under addition: adding two vectors yields a third vector in the same plane the two vectors define, even if this plane is immersed in a 3D space. As a consequence, the code C satisfies the ideal test and is in fact an ideal of the ring $GF(2)[x]/[(x^n - 1)]$. Because it is a subset, the Hamming distance between codewords is generally greater than zero and can be ensured to be greater than zero, as will be shown below, which is what makes error correction possible.

The ring of polynomials $GF(2)[x]$ is a Principal Ideal Domain, which means that every one of its ideals is principal. A **principal ideal** is an ideal that is entirely generated by a single one of its elements. For example, the element 2 of $\mathbb{Z}$ generates the ideal $[2]$ of $\mathbb{Z}$ (the even numbers). A useful theorem ([14], 245) states that if any ideal I in a ring R can be generated by a single element, then the same is true of any ideal of any of the factor rings R/I. This is enough for our purposes, but notice that it is a little weaker than to say that if R is a PID, then R/I is also a PID. This latter statement is true only if polynomial that generates I is irreducible, in which case as we saw the factor ring R/I is actually a field. If R is a PID but the polynomial that generates I is reducible (our case), then the factor ring R/I is not an integral domain; therefore, it cannot be a PID even though every one of its ideals is in fact principal. All of this matches exactly what happens with the ring of integers modulo an integer n.

In our case we are working with $(x^n - 1)$. The factor ring $GF(2)[x]/[(x^n - 1)]$ is not a field and is not a PID either. However, every one of its ideals *is* principal ([14], 245). This implies that our code C is entirely generated by a single polynomial $g(x)$, aptly called the generator polynomial. The **generator polynomial** is the monic polynomial of least degree (for a particular code) that belongs to C and from which every element of C can be generated through multiplication by elements of $GF(2)[x]$ modulo $(x^n - 1)$. There is a unique such polynomial for any ideal of $GF(2)[x]/[(x^n - 1)]$, and each such instance divides $(x^n - 1)$ ([19], 32). It follows that to generate all the possible cyclic codes for a given value of n we need to find all the irreducible factors of $(x^n - 1)$. The possible generator polynomials $g(x)$ are all the possible divisors of $(x^n - 1)$, formed as single irreducible factors or as products of these irreducible factors. If $g(x)$ is chosen of degree $n - k$, a linear code results and the generator matrix can be formed starting from the simple statement $m(x)g(x) = c(x)$, where $m(x)$ is a polynomial of degree $k - 1$ representing a message

block. Writing out $m(x)$ we see that each term will cause a cyclic shift in $g(x)$:

$$\begin{aligned} c(x) &= m(x)g(x) \\ &= (m_0 + m_1x + ... + m_{k-1}x^{k-1})g(x) \\ &= m_0g(x) + m_1xg(x) + ... + m_{k-1}x^{k-1}g(x) \end{aligned} \tag{2.22}$$

Thus, for $g(x) = g_0 + g_1x + g_2x^2 + ... + g_{n-k}x^{n-k}$, dropping the x, the generator matrix looks as follows:

$$c = [m_0m_1...m_{k-1}] \cdot \begin{bmatrix} g_0 & g_1 & \cdots & g_{n-k} & 0 & 0 & \cdots & 0 \\ 0 & g_0 & g_1 & \cdots & g_{n-k} & 0 & \cdots & 0 \\ 0 & 0 & g_0 & g_1 & \cdots & g_{n-k} & \cdots & 0 \\ \cdots & & & & & \cdots & & \\ 0 & 0 & 0 & \cdots & g_0 & g_1 & \cdots & g_{n-k} \end{bmatrix} \tag{2.23}$$

Furthermore, there exists a polynomial $h(x)$ such that $g(x)h(x) = (x^n - 1)$, which gives rise to the check matrix.

Fig. 6 summarises the concepts discussed so far. The width of the rectangles represents the dimension of the vector space or the largest possible number of terms of the polynomial representation of the code. $g(x)$ is shown in red to emphasise that it is one of the code polynomials, the one of least degree. The number of elements in the factor ring is 2^n. A cyclic code is an ideal of this ring, and therefore a subset. Before reaching the next kind of codes (BCH codes) we need to discuss a particular aspect of the check matrix.

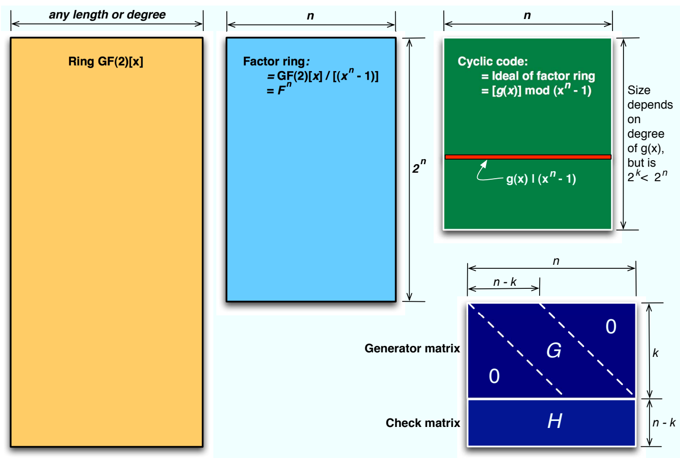


Fig. 6. **Schematic summary of cyclic code**

H. The Roots of $g(x)$

The discussion now enters a territory where abstract algebra becomes increasingly entangled with linear algebra. Although the immediate benefit is just a better understanding of network coding, it seems worthwhile to chart this territory as carefully as possible due to two apparently separate observations: the DNA code exhibits the same structure as a Lie algebra (i.e. a vector space with a commutator operation between its elements) [20], and some aspects of finite state automata can be formalised through group theory. Of course, a third reason is the algebraic structure of logic, which is explored in the second part of this paper, but we currently understand the connections between algebra and logic at a more abstract level. Network coding is a relatively specialised area whose relevance to logic is not immediately obvious. So, whereas the investigation of the connections between algebra and logic is a broader and longer-term endeavour in our research, the algebraic machinery of network coding that we are slowly unravelling seems closer and more relevant to the development of a mathematical modelling framework for the DNA and for interacting state machines. The goal of the discussion, therefore, remains to *understand*, rather than to utilise network coding results that have been known for decades and that are not, in and by themselves, likely to be directly applicable to biologically-inspired computing.

Purser ([19], 35) points out that since $g(x)$ is the generator polynomial of a cyclic code, it necessarily divides all the codewords, in addition to dividing $(x^n - 1)$. Therefore, the roots of $g(x)$ are also (some of) the roots of any codeword polynomial $c(x)$. Let α be one such root. Then, for any

$$c(x) = c_0 + c_1x + c_2x^2 + ... + c_{n-1}x^{n-1}, \tag{2.21}$$

$$c(\alpha) = c_0 + c_1\alpha + c_2\alpha^2 + ... + c_{n-1}\alpha^{n-1} = 0 \tag{2.24}$$

As a consequence, the row vector

$$(1, \alpha, \alpha^2, \alpha^3, ..., \alpha^{n-1}) \tag{2.25}$$

must belong to the null space of G, i.e. it must be a row of H. This argument is a little 'sneaky' because in Eq. (2.21) we are using the code polynomial as just one convenient way to represent a code vector, where the powers of x play the role of basis vectors, whereas in Eq. (2.24) we have switched completely to the algebraic perspective and are treating the polynomial as a function of x. None-the-less we have done nothing wrong and the conclusion (2.25) stands.

Notice also that, whereas when describing linear block codes and cyclic codes we worked mainly with factor rings and their ideals, we appear to have brought the roots of polynomials back into the discussion. To avoid possible confusion let's clarify what is happening. Focus for the moment on an irreducible $g(x)$. Since we know that it is of degree $m = n-k$, the roots that we are talking about must lie in a field extension $GF(2^m)$. We must emphasise that this field is different and additional to any other fields or rings we have been talking about until now. If $g(x)$ is irreducible its roots in $GF(q^m)$ are expressible as powers of the first one, as follows:

$$\alpha_i = \alpha_1^{q^{i-1}} \tag{2.26}$$

For $GF(2)$ this results in the sequence

$$\alpha, \alpha^2, \alpha^4, \alpha^8, ..., \alpha^{2^{m-1}} \tag{2.27}$$

Therefore, a check matrix of $n - k$ rows by n columns can be constructed as follows:

$$H = \begin{bmatrix} 1 & \alpha & \alpha^2 & \cdots & \alpha^{n-1} \\ 1 & (\alpha^2) & (\alpha^2)^2 & \cdots & (\alpha^2)^{n-1} \\ 1 & (\alpha^4) & (\alpha^4)^2 & \cdots & (\alpha^4)^{n-1} \\ & \cdots & & \cdots & \\ 1 & (\alpha^{2^{m-1}}) & (\alpha^{2^{m-1}})^2 & \cdots & (\alpha^{2^{m-1}})^{n-1} \end{bmatrix} \tag{2.28}$$

The check matrix also gives a way to determine the minimum distance of a code. In a linear code the minimum distance between any two codewords equals the minimum weight (the weight is the number of 1s in a codeword) among all the codewords. Furthermore, as indicated in Fig. 5, multiplying any codeword (or any row of G) by H^T equals zero. As a consequence, codewords of weight w correspond to

dependence relations among sets of w columns of H (or rows of H^T)([14], 243). Therefore, the minimum weight among all codewords, which equals the smallest distance between any two codewords of a code, is equal to the minimum number of linearly dependent columns of H.

In building cyclic codes it is easy to set their length and dimension. However, the spacing (in terms of Hamming distance) of the $c(x)$ codewords within the vector space F^n is not necessarily uniform, which means that the minimum distance is both uncertain and not easy to find, especially as n becomes large. BCH codes are also cyclic codes, but they introduce an additional constraint that makes it possible to specify the minimum distance d at the outset. The fact that, as already mentioned, the vector field F^n has the same cardinality as $GF(2^n)$ may at first seem to imply that F^n is a field, which would be potentially confusing since we just spent considerable effort to prove that F^n (in the present context) is a ring and not a field. Isomorphism, however, requires more than cardinality. It also requires that the elements map 1-1 and that the same map satisfy Eqs. 2.6. It is worthwhile explaining this point carefully because it will make it easier to understand the BCH codes, so let us introduce an example.

Assume that, using the notation of Eq. (2.5), $R = GF(2)[x]/[(x^4-1)]$ and $S = GF(2)[x]/[(x^4+x+1)]$. Since we are working with the ground field $GF(2)$ for the coefficients it does not really matter whether we use $+$ or $-$. The first of these polynomials is reducible, the second irreducible. They both give rise to a factor ring of the same cardinality as $GF(2^4)$, composed of all possible polynomials of degree 3 or less: there are 16 of them, including the ground field elements 0 and 1. More interestingly, these are necessarily *exactly the same* polynomials. Thus, we could envisage a 1-1 map θ between the elements of these two sets—in fact, the identity map. The point is that this map is not a homomorphism, and therefore not a isomorphism either, as we now show.

R is not a field while S is because these two sets are the remainders of different ideals. Therefore, if we take any two elements of R and multiply them together, for example, the result may be of degree higher than 3. Thus, it would need to be divided by (x^4-1), keeping the remainder, which is another element of R. If the *same* two elements are taken from S, now their product will need to be divided by (x^4+x+1), keeping the remainder. This second remainder will be an element of S that is *different* to the corresponding remainder we got in R. Therefore Eq. (2.6b) is not satisfied and we don't have a homomorphism. As a consequence the map is not an isomorphism even if it could be the identity map as in this example.

In general, therefore, because $GF(2^n)$ is a vector field of *remainders*, it cannot be considered independently of the ideal that generated it as a factor ring. When $GF(2^n)$ is the splitting field for (x^n-1), i.e. the field that contains all of the roots of all of the irreducible factors of (x^n-1), it will contain several ideals, each of which is generated by a single element (which could be the product of these irreducible factors) and each of which serves as a possible cyclic code.

But let us go back to the generator polynomial $g(x)$ and to its roots in the extension field $GF(2^m)$. Fig. 7 shows how the discussion of $g(x)$ and its roots concerns quite a different field, even if, in fact, these same roots will also solve any of the codeword polynomials. We know that if α is a primitive element of $GF(2^m)$ all the elements of this field extension can be represented as successive powers of this root. This represents none other than the multiplicative group of $GF(2^m)$, i.e. $GF(2^m)\setminus\{0\}$. We also know that each of the elements of $GF(2^m)$ can also be represented as an m-dimensional vector over $GF(2)$, or as a polynomial in α over the same ground field and of degree at most $m-1$. It is possible to continue the discussion in general terms, but the formalism becomes rather overbearing. It is more effective to drop down to a specific example in order to communicate ideas whose general applicability is in any case still visible and easily understood.

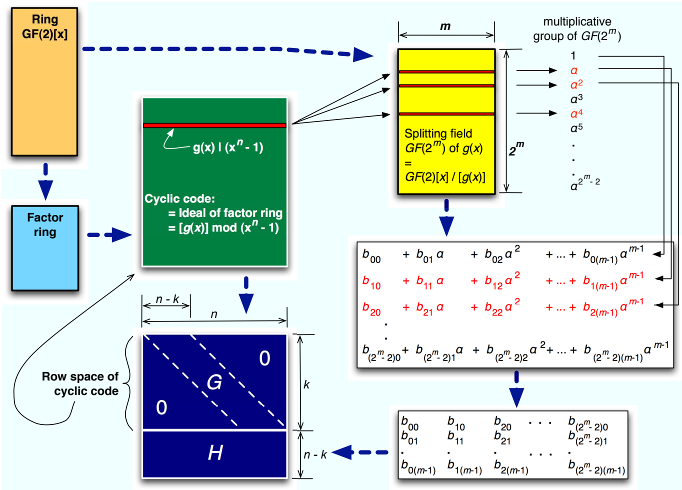


Fig. 7. **The roots of** $g(x)$

For example, take

$$g(x) = x^4 + x + 1, \tag{2.29}$$

as the generator polynomial for a code of length 15. Thus, $m=4$, $n=15$, and $k=11$. To say that α is a root of $g(x)$ is equivalent to saying that

$$\alpha^4 = \alpha + 1. \tag{2.30}$$

Applying this fact to generate the 14 roots of this field extension in polynomial form from the same roots expressed as successive powers of α is equivalent to constructing the factor ring modulo $g(x)$ (this is shown in general form on the right-hand side of Fig. 7):

$$\begin{array}{lcllllllll}
(\alpha^0 & = & 1) & & & & & & \\
\alpha^1 & = & & & \alpha & & & & \\
\alpha^2 & = & & & & & \alpha^2 & & \\
\alpha^3 & = & & & & & & & \alpha^3 \\
\alpha^4 & = & 1 & + & \alpha & & & & \\
\alpha^5 & = & & & \alpha & + & \alpha^2 & & \\
\alpha^6 & = & & & & & \alpha^2 & + & \alpha^3 \\
\alpha^7 & = & 1 & + & \alpha & + & & & \alpha^3 \\
\alpha^8 & = & 1 & + & & & \alpha^2 & & \\
\alpha^9 & = & & & \alpha & + & & & \alpha^3 \\
\alpha^{10} & = & 1 & + & \alpha & + & \alpha^2 & & \\
\alpha^{11} & = & & & \alpha & + & \alpha^2 & + & \alpha^3 \\
\alpha^{12} & = & 1 & + & \alpha & + & \alpha^2 & + & \alpha^3 \\
\alpha^{13} & = & 1 & + & & & \alpha^2 & + & \alpha^3 \\
\alpha^{14} & = & 1 & + & & & & & \alpha^3 \\
(\alpha^{15} & = & 1) & & & & & &
\end{array} \tag{2.31}$$

Since in this example $m = 4$, these roots can now be substituted into the 4 rows of H in Eq. (2.28), after reducing the powers modulo 15. This is done by treating each root as a 4×1 column vector. As a consequence H becomes a 16×15 matrix. As discussed by Purser ([19], 36), since for an irreducible $g(x)$ (actually primitive in this example) the higher roots are given as powers of the first (Eq. (2.27)), the second, third and fourth rows of H in Eq. (2.28) do not add any information. For this example, direct substitution does in fact show that of the 16 rows of H thus constructed only 4 are linearly independent. We might as well, therefore, only retain the first row of H in Eq. (2.28) which, since n was chosen to be 15 in this example, corresponds to all the vectors shown in Eq. (2.31). These can in fact be directly entered as tuples over $GF(2)$ as follows:

$$H = \begin{bmatrix} 100010011010111 \\ 010011010111100 \\ 001001101011110 \\ 000100110101111 \end{bmatrix} \begin{matrix} \alpha^0 \\ \alpha^1 \\ \alpha^2 \\ \alpha^3 \end{matrix} \qquad (2.32)$$

where the powers of α to the right of each row of the matrix are shown just to help recognise the patterns of 1s and 0s corresponding to Eq. (2.31). This matrix is shown in general form at the bottom-right of Fig. 7.

The choice of m such that $n = 2^m - 1$ makes this example a Hamming code, for which the minimum distance is $d = 3$ (and therefore this code can only correct 1-bit errors per codeword, at most). To see this we just need to note that all the elements of $GF(2^m)$ (except for 0) fit in H as distinct columns. The sum of any two, however, necessarily gives another column from the same set due to the closure of $GF(2^m)$ with respect to addition. Therefore, the largest number of linearly independent columns is 2 and the smallest number of linearly dependent ones is 3. As proven a few paragraphs above, the minimum distance is therefore 3. If we wanted a greater distance for the same length n, still using an irreducible $g(x)$ (necessarily of higher degree), the code rate k/n would suffer. It turns out that if, on the other hand, $g(x)$ is not irreducible but is composed of 2 or more irreducible factors, the rows of H will be formed by more than one root. As a consequence, the greatest number of linearly independent columns of H is more likely to be larger than 2, leading to a greater minimum distance. The code rate is not necessarily going to be the same, which highlights how optimisation and trade offs are required when mathematical results encounter engineering applications–which however is not our main concern here.

As explained by Purser ([19], 47), the above observation motivated the development of the BCH codes, for which the generator polynomial is the least common multiple of the minimal polynomials of $d - 1$ consecutive powers of a primitive n^{th} root of unity, for a desired minimum distance d. A **primitive** n^{th} root of unity in a field F is an element a whose order in the multiplicative group of F is precisely n. As a consequence, a and all of its powers are roots of the equation $x^n - 1 = 0$. By the definition of order of a group element, all the powers of a up to a^{n-1} are distinct. Therefore, all the n roots of $x^n - 1 = 0$ are generated by a and its powers. From these we can choose $d - 1$ consecutive ones.

Fig. 8 shows the steps of the process required to develop a BCH code. For a given code length n we need to find a primitive n^{th} root of unity, and the smallest field that contains $GF(q)$ and a primitive n^{th} root of unity is $GF(q^e)$, where e is the order of q mod n ([14], 248). If n and q are two coprime integers, the **order** of q mod n is the smallest positive integer e for which $q^e = 1$ mod n. In other words, e is the order of q in the multiplicative group $\mathbb{Z}_n$. For example, if $n = 10$ and $q = 3$, $e = 4$. In other words, the smallest field that contains a primitive 10^{th} root of unity over the base field $GF(3)$ is $GF(3^4)$. As another example, for $GF(2)$ and $n = 21$ $e = 6$, because $2^6 = 64 = 1 \mod 21$. In practice, one tends to use a value of $n = 2^e - 1$: $n = 127$ for $e = 7$, or $n = 255$ for $e = 8$. These codes are called, unsurprisingly, primitive BCH codes since the n^{th} root of unity β in such a case is a primitive element of $GF(2^e)$.

Fig. 8 shows how the construction of H is similar to what we just discussed above. To find $g(x)$ one first needs to find a primitive polynomial of $GF(2^e)$, and then several minimal polynomials for the $d-1$ roots. Finding a primitive polynomial is a task that needs to be done only once for each value of e. In addition, any one among the set of primitive polynomials for a particular value of e will do, since the others give rise to isomorphic fields. Finding the minimal polynomials, on the other hand, can be laborious. Here is where Gordon's method becomes useful. We will mention it briefly, mainly to highlight another interesting structural aspect of the algebra of network coding.

I. An Observation on Gordon's Method

Gordon's method [17] is relevant to BCH codes and is concerned with finding the minimal polynomials given the knowledge of their roots in the extension field $GF(2^e)$. These roots can be expressed as consecutive powers of a primitive element or as the remainders of $f(x)$ in the factor ring $GF(2)[x]/[f(x)]$, where $f(x)$ is the irreducible primitive polynomial that generates the extension field $GF(2^e)$. Because the degree of these minimal polynomials can be at most e, there are several in any given extension field with 2^e roots. Gordon realised that, for the same reason, each minimal polynomial can also be expressed modulo the same primitive polynomial. In other words, each minimal polynomial can be expressed as one of the elements from the set of its own roots. This makes the determination of the minimal polynomials trivially simple and ideally suited for low-power space probes, where every bit of memory and CPU cycle counts. Even more surprisingly, however, we cannot avoid the conclusion that a minimal polynomial and one of its roots may actually have the same polynomial form! It is not clear whether this fact has any significance, but such examples of structural invariance cannot help but stimulate our curiosity. This particular example is reminiscent of conformal invariance of differential equations under the action of a Lie group of transformations or, to a smaller extent, of the concepts of eigenvector in oscillatory systems or the renormalisation group of statistical physics.

In this section we have used network coding to give a flavour for the richness of algebraic structure, which may at first seem quite far removed from biology. As it happens, the work reported in [8] [20] indicates otherwise and motivates further investigation in Lie algebras. But let's turn to the structure of logic first.

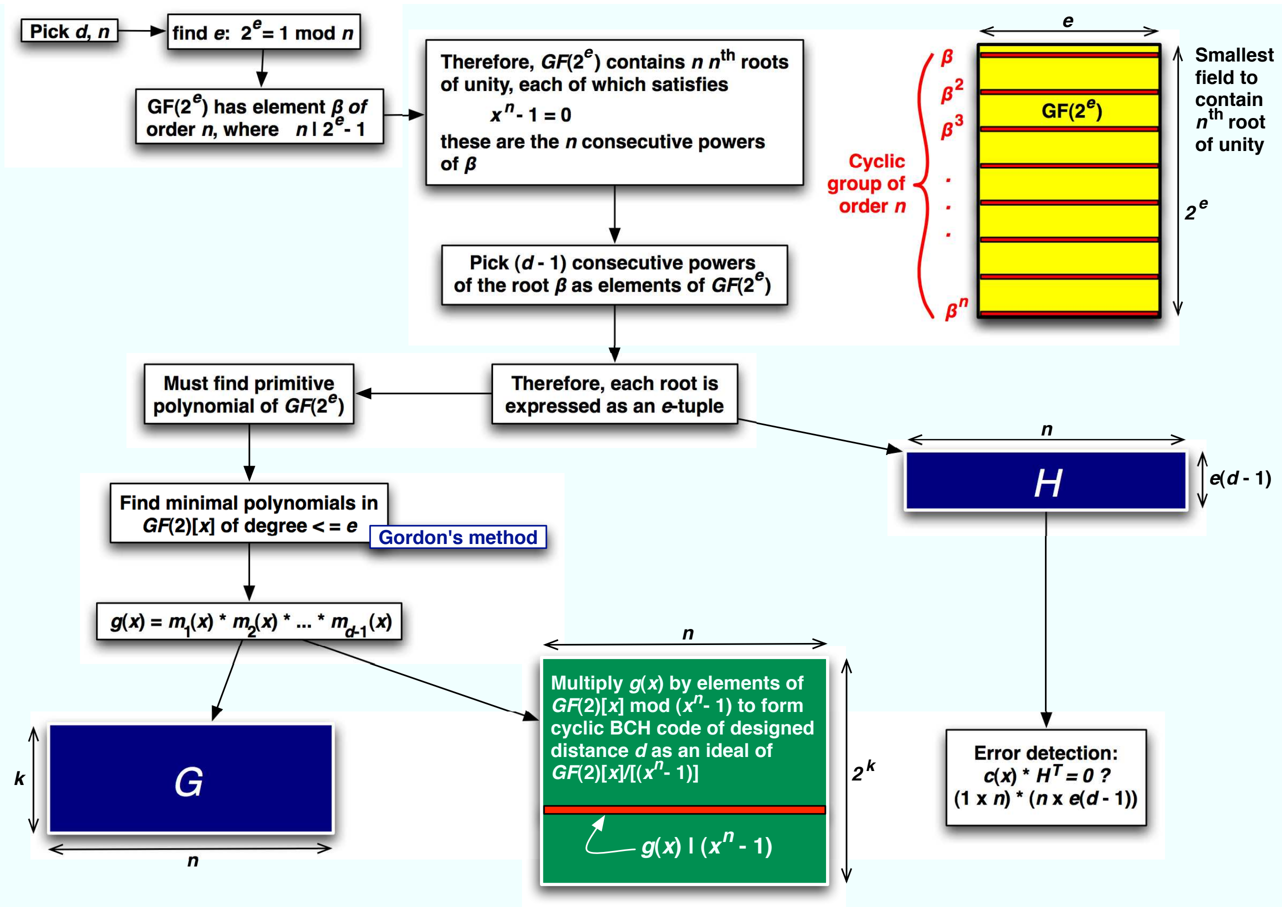


Fig. 8. **Schematic summary of BCH code**

III. Logic, Algebra, and Security

Having presented a general overview of abstract algebra and its very basic concepts, this section is going to establish a link between the disciplines of logic, algebra, and security. For this purpose we start by defining propositional and first-order logic. Section 2 should help to see the strong similarities between logic and algebra, especially when comparing Galois fields and propositional logic. We will attempt to show how the different logics find their counterparts in the realm of abstract algebra. With the informal extension of the basic concepts of logic and algebra to temporal logic and quantifier algebras, respectively, we will bridge from algebra to security. This not entirely obvious bridge is emphasised in the final sub-section, where we show that the intentional application of temporal logic and its wide use in the realm of computer science is process and program analysis and is therefore directly linked to security.

A. Propositional and First-Order Logic

In this section we want to give a short introduction to propositional and predicate logic, two important fields of the research area of symbolic logic. Of course, this introduction cannot be exhaustive and we will refer the reader to appropriate literature.

The following Section will be the basis for Sections III-B and III-C. However, for the experienced reader who is familiar with the basic concepts of these logics we recommend to proceed directly to Section III-B.

1) Propositional Logic: Propositional logic is often called sentential logic. Both names account for the nature of this logic since it uses sentences, thus **sentential**, which can be thought of as **propositions**. The classical propositional logic studies the logical values of sentences and the effects of propositional connectives, such as *and* and *or*, which can be used to form new sentences out of atomic sentences. To formalise this quantitative description textbooks usually start to define the syntax of the language used to describe these relations. In this paper we take another approach and define syntax and semantics concurrently, starting at the very basic building blocks of the language. This will help to expose the algebraic structures of propositional logic and facilitate the understanding of Section III-B.

The basic building blocks of sentences in propositional logic are so-called **atomic formulas**. An atomic formula is a propositional letter. The concept of atomic formula denotes the simplest functional (or semantic, or “signified”) unit of PL. The concept of propositional letter denotes the simplest structural (or syntactical, or “signifier”) unit of PL. There is an infinity of possible propositional letters, and each clearly can serve as an atomic formula.

As propositional logic studies classical two-valued logic, each atomic formula can assume one of the two possible truth values “true” and “false”. Truth values therefore introduce a third layer on top of semantics and syntax. Without loss of generality, it is mathematically convenient to consider these two truth values to be *also* atomic formulas. To represent these values in propositional formulas we use the propositional

letters (symbols) $\top$ and $\bot$, respectively. (A **propositional formula** can be: an atomic formula, the negation of an atomic formula, or the combination of atomic formulas according to operators to be defined below.) Please note that the choice of symbols is arbitrary and any other representation, such as 1 and 0 or t and f, would work just as well. In contrast, the interpretation of these symbols is not arbitrary and is defined using a so-called Boolean valuation function v. This function maps the two atomic formulas to a set $\mathcal{T}$. With the current atomic formulas, $\mathcal{T}$ could be defined arbitrarily. However, in reference to Section II-B we choose $\mathcal{T} = \mathbb{Z}_2$ and define $v'''(\bot) = 0$ and $v'''(\top) = 1$. Why $\mathbb{Z}_2$ is a perfect choice and why we introduce v in the first place will become clearer in the remainder of this section.

Propositional letters are sometimes called variables. Often, as a convenient shorthand, the concept of propositional letter is used interchangeably with atomic formula. Therefore the concept of truth value can also be applied to propositional letter. As an example, take the statement "viruses are the source of all evil". This proposition can be represented by the letter E. Its truth value depends on the valuation function which either maps $v'''(E) := 0$ or $v'''(E) := 1$. The fact that a propositional letter can stand for a proposition introduces a fourth layer to the structure of propositional logic. This fourth layer is the lowest and it is composed by propositions.

Propositions, however, can themselves have internal structure. And this structure could be represented using propositional logic as an atomic formula, or as a combination of atomic formulas, i.e. as a propositional formula. Therefore, we see that propositional logic has a recursively nested structure. Remarkably, the human mind does not seem to mind.

With these symbols, letters and their interpretation at hand we cannot do much and are far from constructing complex formulas in propositional logic. To do so, we need some operators which allow us to combine atomic formulas to create more complex formulas.

The first very basic and unary operator we introduce for this purpose is the so called "not" operator, henceforth referred to as **negation**. Using the symbol $\neg$ we can now extend our formula set with formulas such as $\neg\neg\neg\bot$ or $\neg\top$. The intuitive meaning of negation is the inversion of the truth value of the atomic formula it is applied to. That is, if we interpret $\top$ as "true", $\neg\top$ is interpreted to "false". Accordingly, $\neg\neg\neg\bot$ is evaluated to "true". Obviously, the interpretation of the latter example is not as easy anymore and several simple steps have to be taken to **evaluate** the formula. To put this evaluation on a formal basis we first define the mapping $\neg : \mathcal{T} \rightarrow \mathcal{T}$ with $\neg(0) = 1$ and $\neg(1) = 0$. With this mapping we can extend our very simple mapping v''' and obtain v'' defined by

1) $v''(\top) = 1; v''(\bot) = 0$
2) $v''(\neg X) = \neg v''(X)$ with $X = \top$ or $X = \bot$

Please note that in this definition $\neg$ has been used as a mathematical function as well as a symbol.

A major problem with this definition is its limitation to formulas which contain only one $\neg$ symbol. Thus, with this definition we cannot evaluate our formula $\neg\neg\neg\bot$. However, we can solve this problem by defining our valuation over formulas which we can currently construct (using atomic formulas and negation). Hence, we define the set $\mathcal{P}$ of currently defined formulas and the valuation v' through

1) $v'(\top) = 1; v'(\bot) = 0$
2) $v'(\neg X) = \neg v'(X)$ with $X \in \mathcal{P}$

This type of definition implicitly allows us to recursively define the formulas which we are currently allowed to construct. It can be transformed in a kind of grammar, which is a building plan, according to which we are able to construct the set $\mathcal{P}$ of all **well-formed formulas**:

1) If A is an atomic formula, $A \in \mathcal{P}$
2) If $X \in \mathcal{P}$ then $\neg X \in \mathcal{P}$

With this construction plan we can build formulas which consist of exactly one atomic formula and of none or a number of preceding negation symbols. To allow more than one atomic formula we extend our operator set by a binary operator denoted by the symbol $\vee$. With its help we obtain the set $\mathcal{P}$ of formulas constructed according to the following rules:

1) If A is an atomic formula, $A \in \mathcal{P}$,
2) If $X \in \mathcal{P}$ then $\neg X \in \mathcal{P}$, and
3) If $X, Y \in \mathcal{P}$ then $(X \vee Y) \in \mathcal{P}$.

Similarly, the evaluation function has to be adapted such that it is also defined for this new binary operator. This means that a feasible algebraic interpretation for $\vee$ has to be found. In natural language $\vee$ is often denoted by the word "or", i.e. if one of the propositions combined by $\vee$ is "true", the combination of the propositions is also "true". As we implicitly associated the truth value "true" with 1 we can easily extend our valuation function v' to:

1) $v(\top) = 1; v(\bot) = 0$
2) $v(\neg X) = \neg v(X)$ with $X \in \mathcal{P}$
3) $v(X \vee Y) = max(v(X), v(Y))$ with $X, Y \in \mathcal{P}$

The reader will already have noticed that the binary connective defined above is very specific and that there are logical "statements" which seem not to be expressible by this definition. As an example, suppose you want to state in propositional logic: "if propositions A and B hold then C holds". At first sight this sentence cannot be expressed with the definitions above as we are missing the operators "and" and "if ... then ...". However, together with our valuation we can actually express both operators. Look at the formula $F = \neg(\neg A \vee \neg B)$ with $A, B \in \mathcal{P}$. Obviously, this formula can be constructed with the rules above. Thus $F \in \mathcal{P}$. Evaluating this formula step by step yields that $v(F)$, iff $v(A) = 1$ *and* $v(B) = 1$. This mean F basically combines A and B in such a fashion that the valuation of F corresponds to the logical connective "and". As it is often burdensome and illegible to construct a simple "and" out of the two symbols $\neg$ and $\vee$, propositional abbreviations for binary operators are introduced. For an "and" we use the symbol $\wedge$. Accordingly, the "if ... then ..." operator, also called **implication**, can be expressed as $((\neg X) \vee C)$ but is abbreviated writing $X \Rightarrow C$.

Binary connectives have two input variables. As each of these variables can valuate to elements of a binary set, namely $\mathbb{Z}_2$, we have four possible input combinations. Each of these combinations is mapped to the same binary alphabet, the truth values of the combination. Thus, there are $2^{2^2} = 16$ such possible abbreviations, henceforth called operators. Fig. 9 shows this pictorially, and Table III lists these operators together with their valuation. It also includes trivial ones, i.e. $\top$ and $\bot$, and negation, i.e. $\neg A$ and $\neg B$.

It turns out that the construction rules above together with the valuation definition are equivalent to the definitions of propositional logic and its calculus. Our valuation rules are

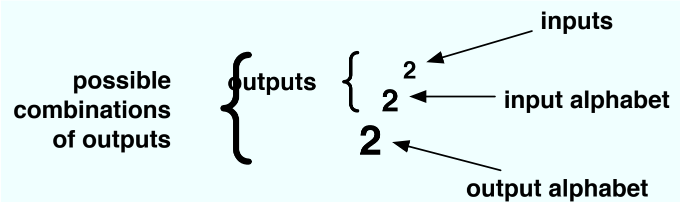


Fig. 9. **Number of binary connectives in propositional logic**

also referred to as **Boolean valuation** or **interpretation**. A special interpretation, which will play an important role in the next section, is a **model**. This is a valuation v for a formula F for which $v(F) = 1$ always. With the term model we can also define a **theory**. This is a set $\mathcal{F}$ of formulas for which all possible interpretations are also models of the formulas in $\mathcal{F}$.

With these definitions in hand we can now take an arbitrary formula of propositional logic and define a valuation for it. As an example, we consider the formula $\neg(\neg(P \subset Q) \wedge \neg R)$. To find the "truth value" of this formula we first define our valuation by $v(P) = 1$, $v(Q) = 1$, $v(R) = 0$. Finally, we apply our valuation rules to the formula, step-by-step:

$$\begin{aligned} v(\neg(\neg(P \subset Q) \wedge \neg R) &= \neg v(\neg(P \subset Q) \wedge \neg R) \\ &= \neg(\neg v((P \subset Q) \wedge \neg R)) \\ &= \neg(\neg(v(P \subset Q) \wedge v(\neg R))) \\ &= \neg(\neg((v(P) \subset v(Q)) \wedge \neg v(R))) \\ &= \neg(\neg((1 \subset 1) \wedge \neg 0)) \\ &= \neg(\neg(1 \wedge 1)) \\ &= \neg(\neg 1) \\ &= \neg 0 \\ &= 1. \end{aligned}$$

One can actually show that such a valuation exists and that it is unique by definition. Here, however, we only wanted to show which valuation steps had to be taken to find the truth value of the propositional formula.

The valuation of propositional formulas and the evaluation of polynomials are both important and help us relate the theoretical concepts to observable or experiential reality. However, as will be discussed below in Section III-B, our concern is less with the kind of value the elements of our set can assume, but more with the structural relationships within and between sets. This is the essence of algebra in its most abstract form, whose role as a unifying framework we are attempting to extend also to biology.

For example, Table III can be compared to Table II. Although at first these seem very different, it turns out that all the binary connectives shown in Table III can be also be expressed in terms of only two binary connectives, $\wedge$ (multiplication) and $\not\equiv$ (addition mod 2). This is equivalent to expressing all the connectives by only using $\neg$, $wedge$, and $\vee$. This is, the connective $\not\equiv$ in the propositional formula $(A \not\equiv B)$ can also be expressed as $((A \vee B) \wedge (\neg(A \wedge B)))$. This can easily be verified by constructing a valuation table comparable for the inputs A and B.

If we recollect the things we learnt about propositional logic we are now able to construct more complex formulas by combining atomic formulas or other formulas. There are various connectives which can be used to build new formulas but basically two binary and one unary operation are enough to yield the same expressiveness. Finally, all the elements we combine can be mapped to the field $\mathbb{Z}_2$ using a so-called valuation function. These general observations and the knowledge of Section II-E already suggest to interpret propositional formulas not only within the realm of logic but as polynomials over the field $GF(2)$.

To make this more explicit we define a translation Ψ which transforms every component of a well defined propositional formula into a polynomial whose value is equivalent to the valuation of this formula with an arbitrary valuation function v. Clearly, the easiest translation is the translation of atomic formulas A_i, namely

1) $\Psi(A_i) := v(A_i) := x_i$, with $A_i \in \mathcal{P}$ and $x_i \in \mathbb{Z}_2$

A little bit more complicated is the translation of the combination of these formulas. However, if we recall again the similarity between the multiplication and addition in the field $\mathbb{Z}_2$ with Table III we can easily define a translation for $\wedge$:

2) $\Psi(A \wedge B) := \Psi(A) \cdot \Psi(B)$, with $A, B \in \mathcal{P}$

To obtain an equivalent polynomial for negation we have to find an operation which turns every 0 or 1 of $\mathbb{Z}_2$ into 1 or 0 respectively. Addition of 1 modulo 2 is an operation which perfectly realises this requirement. Thus, we can define a rule for negation:

3) $\Psi(\neg A) := \Psi(A) + 1$, with $A \in \mathcal{P}$

Finally, it suffices to show that also $\vee$ can be expressed as a polynomial:

4) $\Psi(A \vee B) := \Psi(A) \cdot \Psi(B) + \Psi(A) + \Psi(B)$, with $A, B \in \mathcal{P}$

The equivalence of this polynomial with the valuation of $(A \vee B)$ can be shown by constructing a valuation table and compute the corresponding values for the above polynomial. For this purpose Table IV lists the polynomials over GF(2) which can be used to compute the valuation of the corresponding formula in propositional logic with the same input to output mapping. This table also supports the fact that two binary connectives and negation are sufficient to define all the possible 16 other connectives. Given the results from Section II-E we can directly claim that the set of the polynomials we can construct with the transformation function Psi, are elements of the ring $\mathbb{Z}_2[X]$. Additionally, the structure in Table IV is analogous to a field extension $GF(2^4)$ over the base field $GF(2)$ and the basis $\mathcal{B} = \{1, x_A, x_B, x_A x_B\}$.

Of course, we are not the first to discover the possible translation of formulas of propositional logic into polynomials over $GF(2)$ and their strong link to field extensions. In contrast, the "polynomial representation" or "algebraic transformation" of PL formulas have a long history and date back to the 1970's [21], [22]. Therefore, we refer the interested reader to the survey papers [23], [24]. They discuss the foundations of this theory, refer to appropriate papers and present applications. Especially, the variety and type of applications is an inspiring source for this work and supports our hypothesis. For instance, it has been shown in [25] that a uniform encoding of hash functions in propositional logic can reduce cryptoanalytical problems to well known and well addressed satisfiability problems. Another direction take [26], [27], [28]. The authors of these contributions start in the realm of logic and optimise the verification process of logical circuits

using a polynomial representation. Thus, our research can already build on strong links between algebra and logic which will be additionally emphasised in Section III-B. They will allow us to transfer observations or general principles from the structure of the DNA or other biological systems into the realm of logic. However, before we take a further step into this direction the next section is going to shed some light on the expressiveness of propositional logic and reveals some of its deficiencies.

2) First-Order Logic: As we have seen above, propositional logic can be used to compose formulas which represent simple propositions. These propositions can be used to declare other propositions. This implies that the meaning of a composed proposition is always derived from the most basic propositions, the atomic formulas. Atomic formulas in turn strongly depend on the valuation function, which is context-independent and which therefore becomes a characteristic of propositional logic. Generally speaking, propositional logic is not able to express facts about the environment. As an example, consider the following proposition: "all even integers are divisible by two". With propositional logic we would not be able to express this proposition except by writing one proposition for each number.

First-order logic (FOL) addresses this very simplistic and inflexible view. It replaces the pure atomic formulas by predicates which can have arguments. Thus, first order logic is often called predicate logic. Due to the introduction of variable arguments FOL also supports quantifiers to bind variables. In this way, the above example can be translated into $(\forall x)(E(x) \supset D(x,2))$, where $E(x)$ is the predicate which is only true if x is even. $D(x,2)$ on the other hand is the predicate which valuates to true if x is divisible by the constant 2.

As in the last subsection we will briefly introduce the syntax of first-order logic, also called first-order language. After introducing the general concept of a **model** we define the semantics of FOL.

FOL is defined over a first-order language $L(\mathcal{R},\mathcal{F},\mathcal{C})$. $\mathcal{R}$ is a finite, countable set of relation symbols (predicate symbols) with specific arity [4]. The finite or countable set $\mathcal{F}$ contains function symbols each of which also has associated an arity. Finally, $\mathcal{C}$ is also a finite countable set of constant symbols. This determines what kind of sets are used to construct all formulas. Comparable to propositional logic we have to define the rules according to which the elements of these sets can and must be combined in order to obtain formulas valid in first-order logic.

Obviously, we have finer granularity compared to propositional logic. Because of that we construct FOL **formulas** out of simpler building blocks, so-called **terms**. They are similar to propositional logic formulas with the major difference that they can contain **variables**. They represent elements of a set which we will later call **domain**. This is similar to propositional letters whose valuation can be either "true" or "false". However, the latter variability is fixed with the valuation function. In contrast, variables of FOL can adopt different values even during the valuation process and finally influence it at the same time.

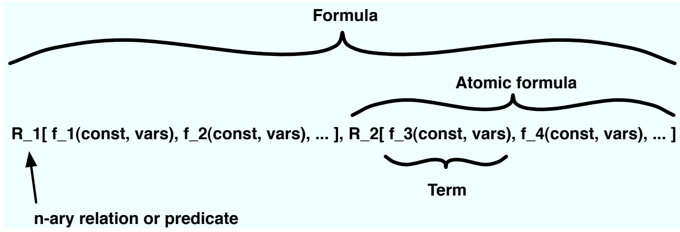


Fig. 10. **Abstract structure of FOL formulas**

This is also why variables and constants are considered to be terms of FOL. Additionally, functions are also considered to be terms if all of their arguments are also terms. Obviously, this definition is again recursive such that arguments of functions can be functions again. Henceforth, we will denote all the terms we can construct out of $L(\mathcal{R},\mathcal{F},\mathcal{C})$, as $\mathcal{T}_{L(\mathcal{R},\mathcal{F},\mathcal{C})}$.

Based on this definition of terms we now define formulas. You will recognise that the definition is very similar to the definition of formulas in propositional logic. First, we define atomic formulas and with them we construct more complicated formulas of FOL. The main difference is the use of terms instead of propositions and the introduction of the universal ($\forall$) and existential ($\exists$) quantifiers.

An **atomic formula** of FOL is any string of the form $R(t_1, t_2, ..., t_n)$ with $R \in \mathcal{R}$ with $t_1, t_2, ..., t_n \in \mathcal{T}_{L(\mathcal{R},\mathcal{F},\mathcal{C})}$, i.e. every term put into a relation using a symbol of $\mathcal{R}$ is an atomic formula. This makes perfect sense as terms do not have any logical meaning. Instead, as mentioned above, they are assigned elements from a domain D. These elements are then combined with other terms, variables, and constants. As an example, you may consider the binary function $f \in \mathcal{F}$, a variable x, the constants $3, 5 \in \mathcal{C}$, the binary equality relation $E \in \mathcal{R}$, and the domain $\mathbb{N}$. We could now interpret the function f as the addition of two natural numbers. Finally, we map the constants 3 and 5 to the natural numbers 3 and 5 respectively. The variable x represents any value in the domain $\mathbb{N}$. Accordingly, the relation E is informally defined as the set of all tuples whose elements are equal. Mathematically, E can be defined over $\mathbb{N}$ as the set $\{(y, y) : y \in \mathbb{N}\}$. This gives all terms and their combination a meaning over the domain $\mathbb{N}$. However as far as a FOL interpretation is concerned, it does not assign any truth value to them.

The process described above involved two steps. First, a so-called **model** $M = \langle \mathcal{D}, \mathcal{I} \rangle$ was defined. $\mathcal{D}$ denotes our domain, and $\mathcal{I}$ the interpretation of the constants, functions, and relations. Thus, with defining a model we know what kind of meaning is associated with each symbol, i.e. that f is a function $f : \mathcal{D}^2 \rightarrow \mathcal{D}$, E is a relation $E \subseteq \mathcal{D}^2$, a constant is associated with a specific element in the domain $\mathcal{D}$, and that variables are elements of domain $\mathcal{D}$. You will notice that with a given model and a term which does not contain any variables, e.g. $f(3, 5)$, we are already able to compute values for this term. These terms are called **closed** terms.

However, with a model alone we would not be able to compute values for terms which contain variables. This is why FOL also defines an auxiliary tool: **assignments** $\mathcal{A}$. Instead of only specifying that a variable is any element of a domain, an assignment assigns each variable a specific value within the

[4] In logic, mathematics, and computer science, the arity of a function or operation is the number of arguments or operands that the function takes. The arity of a relation is the number of domains in the corresponding Cartesian product. In this research we focus on the Cartesian product of a set with itself ($A \times A$). For example, the arity of the addition operation is 2, which means that addition is a binary operation, or that it takes 2 arguments. (http://en.wikipedia.org/wiki/Arity)

A	B	Operators															
		$\bot$	$\wedge$	$\not\supset$	A	$\not\subset$	B	$\not\equiv$	$\vee$	$\downarrow$	$\equiv$	$\neg B$	$\subset$	$\neg A$	$\supset$	$\uparrow$	$\top$
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

TABLE III

PROPOSITIONAL LOGIC OPERATORS AND THEIR VALUATION

x_A	x_B	Polynomials															
		0	x_Ax_B	$x_Ax_B + x_A$	x_A	$x_Ax_B + x_B$	x_B	$x_A + x_B$	$x_Ax_B + x_A + x_B$	$x_Ax_B + x_A + x_B + 1$	$x_A + x_B + 1$	$x_B + 1$	$x_Ax_B + x_B + 1$	$x_A + 1$	$x_Ax_B + x_A + 1$	$x_Ax_B + 1$	1
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

TABLE IV

POLYNOMIALS IN THE FIELD EXTENSION $GF(2^4)$ WITH BASIS $\mathcal{B} = \{1, x_A, x_B, x_Ax_B\}$

domain $\mathcal{D}$. As a consequence, as soon as a model and an assignment are defined each term is associated with a value. Consider an example we can construct from the terms defined above: $f(x, 3)$. If we consider the model $M = \langle \mathbb{N}, \mathcal{I} \rangle$, with $\mathcal{I}$ as defined above, and the assignment $\mathcal{A}$ which assigns x the value 2. This assignment will, by structural recursion, assign $f(x, 3)$ the value 5.

Now it becomes clear why relations applied to arbitrary terms are defined as atomic formulas in FOL. While the "values" of terms are only defined over a domain, a relation can and must be associated with a truth value to give it its meaning. As an example, consider the atomic formula $E(f(x, 3), 5)$, defined over the model $M = \langle \mathbb{N}, \mathcal{I} \rangle$ and the assignment $\mathcal{A}$ as defined above, short: $E(f(x, 3), 5)^{\mathcal{I},\mathcal{A}}$. It would make no sense to assign this construct a value of the domain. Intuitively a relation should indicate whether its arguments are part of the relation or not. In our case, the binary relation E should valuate to true iff its arguments are equal. In a more formal way, FOL associates with each atomic formula P of $L(\mathcal{R}, \mathcal{F}, \mathcal{C})$ a value $P^{\mathcal{I},\mathcal{A}} \in \{0, 1\}$ as follows:

$[P(t_1, t_2, ..., t_n)]^{\mathcal{I},\mathcal{A}} = 1 \Leftrightarrow \langle t_1^{\mathcal{I},\mathcal{A}}, t_2^{\mathcal{I},\mathcal{A}}, ..., t_n^{\mathcal{I},\mathcal{A}} \rangle \in P^{\mathcal{I}}$, $\top^{\mathcal{I},\mathcal{A}} = 1$, $\bot^{\mathcal{I},\mathcal{A}} = 0$, where $\top$ and $\bot$ are special symbols in $\mathcal{C}$ (also compare Section III-A.1).

With this definition we can now evaluate the term $E(f(x, 3), 5)$. Obviously, with the model defined above it only valuates to 1 iff the assignment $\mathcal{A}$ assigns $x = 2$, to 0 otherwise. This example again emphasises the difference to propositional logic and shows how FOL gains a higher expressiveness by introducing predicates (relations) which can have arguments and therefore give it finer granularity.

Similarly to propositional logic, FOL also needs to define rules that allow the combination of atomic formulas and their truth values. Thus, first order logic defines the following four combination rules for the language $L(\mathcal{R}, \mathcal{F}, \mathcal{C})$:

1) Any atomic formula is a formula.
2) If A is a formula, so is $\neg A$
3) If A is a formula and x is a variable, then $(\forall x)A$ and $(\exists x)A$ are formulas.

As it is very common to abbreviate binary functions by introducing special symbols, FOL redundantly defines binary connectives which can be used in infix notation:

4) For a binary connective $\circ$, if A and B are formulas, so is $(A \circ B)$.

By using the syntax defined above and defining $\mathcal{R}$, $\mathcal{F}$, and $\mathcal{C}$ appropriately, we can specify the same formulas that we find in popular theories such as set theory. Theories using this first-order language are also called first-order theories.

But with the rules alone which describe how formulas can syntactically be combined, the language is still rather useless since apart from the meaning of atomic formulas it has no meaning. Thus, we will have to take a step comparable to what we did in propositional logic. There we extended the valuation function to formulas in order to obtain the full expressiveness of propositional logic.

For FOL formulas we proceed similarly and extend the association of truth values with atomic formulas to all possible formulas. We are going to use the same propositional constants $\top$ and $\bot$ as well as the same connectives (denoted as $\circ$) as in propositional logic. Thus, let $\mathcal{M} = \langle \mathcal{D}, \mathcal{I} \rangle$ be a model for $L(\mathcal{R}, \mathcal{F}, \mathcal{C})$ and $\mathcal{A}$ an assignment in this model. To each formula Φ of $L(\mathcal{R}, \mathcal{F}, \mathcal{C})$ we associate a value $\Phi^{\mathcal{I},\mathcal{A}} \in \{0, 1\}$ as follows:

1) $[\neg X]^{\mathcal{I},\mathcal{A}} = \neg[X^{\mathcal{I},\mathcal{A}}]$.
2) $[X \circ Y]^{\mathcal{I},\mathcal{A}} = X^{\mathcal{I},\mathcal{A}} \circ Y^{\mathcal{I},\mathcal{A}}$
3) $[(\forall x)\Phi]^{\mathcal{I},\mathcal{A}} = 1 \Leftrightarrow \Phi^{\mathcal{I},\mathcal{B}} = 1$ for *every* assignment $\mathcal{B}$ in $\mathcal{M}$ that is an x-variant of $\mathcal{A}$.
4) $[(\exists x)\Phi]^{\mathcal{I},\mathcal{A}} = 1 \Leftrightarrow \Phi^{\mathcal{I},\mathcal{B}} = 1$ for *some* assignment $\mathcal{B}$ in $\mathcal{M}$ that is an x-variant of $\mathcal{A}$.

Here the assignment $\mathcal{B}$ in the model $\mathcal{M}$ is an **x-variant** of the assignment $\mathcal{A}$, if $\mathcal{A}$ and $\mathcal{B}$ only assign a possibly different value to x.

We want to finish this sub-section by giving an example to illustrate the expressiveness of first-order logic.

For this example we assume the language $L(\{R\}, \{\oplus\}, \emptyset)$ with variables x and y. We choose $\mathcal{M} = \langle \mathbb{N}, \mathcal{I} \rangle$ as the model with $\mathcal{I}$ defined as follows:

1) $\oplus^{\mathcal{I}}(a, b) = a + b$, with $a, b \in \mathbb{N}$, and
2) $R^{\mathcal{I}} = \{(x, y) : x, y \in \mathbb{N}, x > y\}$

Now, consider the sentence $(\forall x)(\forall y)(\exists z)R(x \oplus y, z)$. It is easy to see that this sentence always evaluates to truth value 1 in model $\mathcal{M}$. Note that this valuation does not depend on the assignment as all variables in this sentence are bound to a quantifier.

We could now start to introduce proof procedures for propositional and first-order logic which would show the full power of these languages. However, we skip these interesting details and refer the interested readers to [29].

B. Algebraic Logic

Algebraic Logic is the field of research which deals with studies of algebras that are relevant for logic. It additionally investigates the methodology of solving problems in one of the two domains, algebra or logic, and translating the solution back into its original domain.

This section is going to show how propositional logic as well as first order logic are connected to algebra.

1) Propositional Logic and Boolean Algebra: In subsection III-A.1 we introduced the basic syntax and semantics of propositional logic. We defined Boolean evaluation and showed how a simple formula in propositional logic is valuated. If we want to generalise the Boolean valuation for arbitrary formulas with n propositions we will have to investigate 2^n different **interpretations**.

This characteristic becomes a problem if we do not simply want to valuate propositional formulas but if we want to draw logical consequences from them. If we consider a set $\mathcal{S}$ of propositional formulas the logical (propositional) formula is a **logical consequence** if every model of $\mathcal{S}$ is a model for X. In this case we also write $\mathcal{S} \models_p X$. In common language we could state that a logical consequence is a statement which follows from some other statements. In mathematics, for example, the set of statements could be axioms.

So if we want to **prove** in propositional logic that a statement is a logical consequence of other propositional formulas one may use established proof procedures such as Hilbert systems or resolution. Logical consequences can also be interpreted as a means to find propositional formulas which are equivalent, this is, which have the same model. In the field of electronic circuits this is important as it allows to reduce cost and to find formulas with the least number of connectives in it. This may be achieved by simply guessing formulas and proving their logical equivalence by valuation.

In Section II we discussed the basic concepts of abstract algebra. We learnt that the basic language of abstract algebra is set theory and that algebra can be used to apply transformations to sets without destroying the relations between the elements within the set (homomorphism, isomorphism, etc). With this motivation we take a closer look at propositional logic.

Due to the definition of propositional logic any propositional formula has infinite equivalent formulas. As an example consider the atomic formula which consists of the proposition P. This formula can easily be extended with the propositional letter $\bot$ to $P \vee \bot$. Both formulas are equivalent as they have the same model. If we now merge all propositional formulas that we can build from the propositions P and R and that have the same model as P, we obtain a set that represents an equivalence class (see Section II) of propositional formulas with model M. Obviously, we can group all propositional formulas based on the propositions in $\mathcal{P} = \{P, R\}$ and obtain 16 equivalence classes. We call the set of these classes, set $\mathcal{E}$.

Now, according to the notion of algebra, we look at the interrelation between these equivalence classes. For this purpose we introduce a tautology and the concept of satisfiability. A propositional formula X is a **tautology** if $v(X) = 1$ for every interpretation v, i.e. no matter which valuation is chosen for the atomic formulas in X, the valuation of X is always "true". If the formula X is not a tautology there are valuation for which $v(X) = 0$. To satisfy a formula X is the process of finding a valuation for which $v(X) = 1$. If this process is successful the formula is also called **satisfiable**. Accordingly, it is possible to define that a set $\mathcal{S}$ of propositional formulas is satisfiable if there is some interpretation v: $v(s) = 1, \forall s \in \mathcal{S}$.

Thus, the propositional formula $P \vee \neg P$ is a tautology. Consequently, we also find an equivalence class that only contains tautologies. As $v(\top) = 1$ for all interpretations v, we call this equivalence class $[\top]$. Its counterpart is the equivalence class which is not satisfiable. We denote this equivalence class with $[\bot]$.

Intuitively, we now want to define a partial order on $\mathcal{E}$. We define the order $\leq$ as follows:

$$\mathcal{X} \leq \mathcal{Y} \Leftrightarrow v(P) = v(P \wedge R), \qquad \forall P \in \mathcal{X} \; and \; \forall R \in \mathcal{Y}, \\ with \; \mathcal{X}, \mathcal{Y} \in \mathcal{E} \; and \\ for \; all \; interpretations \; v.$$

Why is this an intuitive definition? Consider the valuation of formula $P \wedge R$ with the interpretation v with $v(P) = 1$ and $v(R) = 0$. This yields $v(\top \wedge \bot) = v(\bot)$ which also means that $\bot \leq \top$ which would be an intuitive definition of a relation on the set $\{\bot, \top\}$.

We draw this partial order in Figure 11[5]. Note that this figure is a **Hasse diagram** which is a graph representation for partially ordered sets (posets) $(\mathcal{H}, \leq)$. Vertices $a, b \in \mathcal{H}, a \neq b$ are only connected by an edge if $a \leq b$ and there is no $c \in \mathcal{H}$ such that $a \leq c \leq b$. Additionally, vertex b is drawn higher than vertex a if $a \leq b$. The use of a Hasse diagram is advantageous for our following discussions as the diagram directly shows infimum and supremum of two elements in set $\mathcal{H}$. The **infimum** of a subset $\mathcal{S}$ a partially ordered set $(\mathcal{H}, \leq)$ is usually defined as an element $h \in \mathcal{H}$ such that $h = inf(\mathcal{S}) = max(\{x \in \mathcal{H} | \forall y \in \mathcal{S}, x \leq y\})$. Informally, the infimum is the largest element in $\mathcal{H}$ which is smaller than any other element in $\mathcal{H}$. Therefore, the infimum is also often referred to as the **greatest lower bound**. By contrast the **supremum** describes the **least upper bound**, i.e. h is the supremum of the subset $\mathcal{S}$ of set $(\mathcal{H}, \leq)$, if $h \in \mathcal{H}$ with the characteristic $h = min(\{x \in \mathcal{H} | \forall y \in \mathcal{S}, y \leq x\})$. The supremum is often denoted by $h = sup(\mathcal{S})$.

[5] Inspired by the slides of the talk "A Refined Geometry of Logic" by David Miller given at the Department of Mathematics, University of Warwick, Nov 2005

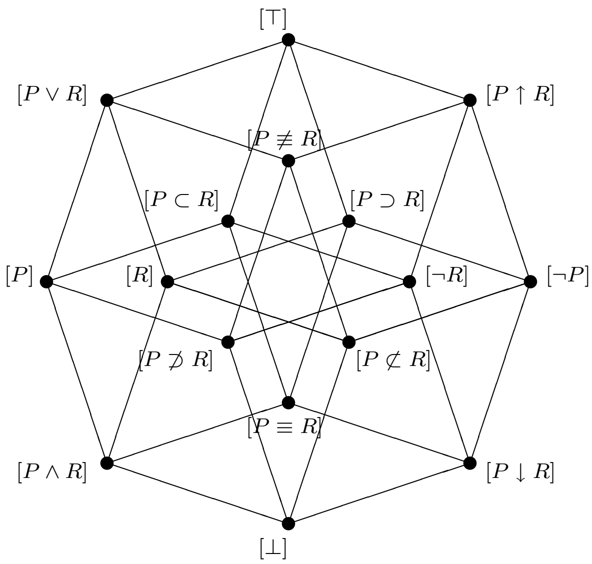


Fig. 11. Hasse diagram of Boolean lattice of PL generated by {P,R}

If we now take a closer look at this poset $(\mathcal{E}, \leq)$ by drawing its Hasse diagram representation we will recognise two things. First of all the poset is bounded by a least and a greatest element, $\bot$ and $\top$ respectively. The second observation is not that obvious and requires a trained eye for propositional logic formulas. Take representatives X and Y from two arbitrary equivalence classes $[X]$ and $[Y]$. The application of the propositional connectives $\vee$ and $\wedge$ to X and Y yields a propositional formula. This formula Z is a representative of an equivalence class $[Z]$. It is the supremum and the infimum respectively of $\{[X], [Y]\}$ and $[Z] \in \mathcal{E}$.

To illustrate this we consider the arbitrarily picked equivalence classes $[P \vee R]$ and $[\neg P]$ which form the new set $\{[P \vee R], [\neg P]\}$. We form the propositional formulas $V = (P \vee R) \vee \neg P$ and $W = (P \vee R) \wedge \neg P$.

Table V shows the possible interpretations of formulas V and W. If we compare the truth values with Table III we

$v(P)$	$v(R)$	$v(P \vee R)$	$v(\neg P)$	$v(V)$	$v(W)$
0	0	0	1	1	0
0	1	1	1	1	1
1	0	1	0	1	0
1	1	1	0	1	0

TABLE V

ALL INTERPRETATIONS OF SUPREMUM AND INFIMUM OF $v(P \vee R)$ AND $v(\neg P)$

can verify that V is equivalent to $\top$ and W is equivalent to $P \not\subset R$. Their equivalence classes are again part of $\mathcal{E}$. Due to the fact that we used a Hasse diagram to display our partial order $(\mathcal{E}, \leq)$ we can instantly see, that $[\top]$ is supremum and $[P \not\subset R]$ is infimum of set $\{[P \vee R], [\neg P]\}$.

Here, we already see, that the order theoretic interpretation based on the partial order $\leq$ finds its equivalence in an algebraic structure based on the two operators $\vee$ and $\wedge$ and vice versa. Based on this observation and using the above Hasse diagram we may additionally verify that any three elements of $\mathcal{E}$ of the presented structure $(\mathcal{E}, \leq)$ also have the characteristics of

1) associativity,
2) commutativity, and
3) absorption.

Thus the partial order $(\mathcal{E}, \leq)$ is a lattice. Additionally, we can show the

4) distributivity of any three elements and
5) existence of the complement to each representative of an equivalence class.

As we skipped the informal proof of characteristics one through three we would like to show the distributive characteristic by example, using the Hasse diagram in figure 11. Take three arbitrary elements from $\mathcal{E}$, e.g. the propositional formulas $A = \neg R$, $B = \neg P$, and $C = P \not\subset R$. The distributivity of lattice $(\mathcal{E}, \leq)$ or its correspondent $(\mathcal{E}, \vee, \wedge)$ requires the following equivalence

$$A \wedge (B \vee C) = (A \wedge B) \vee (A \wedge C),$$

or *translated* into order theory

$$inf(\{A, (sup(\{B, C\})\}) = sup(\{inf(\{A, B\}), inf(\{A, C\})\})$$

The latter can be easily verified using Hasse diagrams. In our case it is $sup(\{B, C\}) = P \uparrow R$, $inf(\{A, B\}) = P \downarrow R$, and $inf(\{A, C\}) = \bot$. Thus, it is $inf(\{A, P \uparrow R\}) = P \downarrow R$ and $sup(\{P \downarrow R, \bot\}) = P \downarrow R$ which proves distributivity for the chosen formulas.

The five characteristics listed above represent the axioms which have been set up by George Boole. Therefore, this distributive complemented lattice is also called a Boolean algebra. Tarski and Lindenbaum were the first to precisely discuss the set of propositional formulas as an algebra with operators which were induced by the connectives of the propositional language. The structural analysis we tried to sketch above by using intuition and geometrical representation in Hasse diagrams is discussed in more detail in [12], [30].

We generalise the result above and give the definition for a **Boolean algebra**. A Boolean algebra is a structure $\mathcal{B} = \langle B, +, \cdot, \bar{\ }, 0, 1\rangle$ where the following system of equations is valid, and where $x, y, z \in B$:

$$\begin{aligned}
x + (y + z) &= (x + y) + z && (associativity)\\
x \cdot (y \cdot z) &= (x \cdot y) \cdot z &&\\
x + y &= y + x && (commutativity)\\
x \cdot y &= y \cdot x &&\\
x + (x \cdot y) &= x && (absorption)\\
x \cdot (x + y) &= x &&\\
x + (y \cdot z) &= (x + y) \cdot (x + z) && (distributivity)\\
x \cdot (y + z) &= (x \cdot y) + (x \cdot z) &&\\
x + \overline{x} &= 1 && (existence\ of\\
x \cdot \overline{x} &= 0 && complement)
\end{aligned}$$

With this mathematical structure the Boolean algebra for propositional logic (PL) can be defined as the Boolean algebra model $\mathcal{B}_{PL} = \langle \mathcal{E}, \vee, \wedge, \neg, [\bot], [\top]\rangle$. You will realise that this algebra is only based on the operators $\vee$, $\wedge$, and $\neg$. As we have shown the equivalence with propositional logic is still valid. However, it becomes even more obvious if we reassure the reader that every propositional formula can be rewritten into an equivalent propositional formula in **normal form** which is only based on these connectives.

Summarising the results of this section we can state that with Boolean algebra we possess a very powerful tool which can be used to transform arbitrary propositional formulas into other propositional and equivalent formulas. These equivalence transformations have numerous areas of applications, such as integrated circuit optimisation or theorem proving to name just two possible domains.

However, of major importance for this contribution is the observation that there is a strong link between propositional logic and algebra. We saw that the definition and structure of propositional logic directly induces a Boolean algebra. Thus, we are now capable to choose a domain, propositional logic or Boolean algebra, which offers the most suitable tools and the best knowledge to analyse a structure. Consequentially, it would be good if the same correspondence between first-order logic and algebra held.

2) First-Order Logic and Quantifier Algebras: In this section we are going to show that first-order logic also possesses an algebraisation. As we can use the results from the last section we forgo an intuitive interpretation and graphical explanation and use a more mathematical approach. Nevertheless, this section is going to avoid complicated and highly mathematical algebraisations of first order logic and follows the spirit of Charles C. Printer who followed "... the most satisfactory way of introducing the [...] non-specialist to the ideas and methods of algebraic logic" [31].

From Section III-A.2 we already know that first-order logic is able to express formulas which are not expressible in propositional logic. This is basically due to the fact that propositional logic has been extended by quantifiers and that it supports n-ary relations as opposed to strict bi-nary relations. Now, one may assume that we simply extend the Boolean algebra, the algebraisation of propositional logic, and obtain a first-order logic algebra. In the next couple of paragraphs this is exactly what we are going to do. For this purpose we first introduce quantifier algebras for formulas. Their definition is very similar to the construction of a Boolean algebra out of propositional logic. Thus, from the last section we simply collect the elements which we need for a formal definition.

Let Γ be the first-order language $L(\mathcal{R}, \mathcal{F}, \mathcal{C})$ and let $\langle v_\kappa \rangle_{\kappa<\alpha}$ a ordered sequence of variables. Let Θ be a theory of Γ (see Section III-A.1). We define $\mathcal{F}^\Gamma$ as the set of all formulas of Γ. We currently have to restrict our considerations to the set of formulas which does not contain the formulas $F \equiv_\Theta G$. Here relation $\equiv_\Theta$ denotes the equality relation which can be deduced from Θ. This set is denoted by $\mathcal{F}^\Gamma / \equiv_\Theta$. We will account for this restriction later in this section.

Based on $L(\mathcal{R}, \mathcal{F}^\Gamma / \equiv_\Theta, \mathcal{C})$ and the general Boolean algebra $\mathcal{B} = \langle B, +, \cdot, \bar{}, 0, 1 \rangle$ we can define the following Boolean operations:

$$\begin{aligned} (F/\equiv) + (G/\equiv) &= F \vee G/\equiv \\ (F/\equiv) \cdot (G/\equiv) &= F \wedge G/\equiv \\ \overline{(F/\equiv)} &= \neg F/\equiv \end{aligned}$$

1 denotes all formulas of theory Θ. For simplicity and consistency with previous sections we write $\top$. 0, the negations of all formulas in $\top$ is denoted by $\bot$. We obtain the Boolean algebra $\langle \mathcal{F}^\Gamma / \equiv_\Theta, +, \cdot, \bar{}, \bot, \top \rangle$.

To define quantifier algebras we are only missing two more operations which find their analogy in the quantifiers in first-order logic. For $\exists$ we define the operation $\exists_\kappa$ with $\exists_\kappa(F/\equiv)$ denotes the equivalence class of all formula $(\exists v_\kappa)F$. Quantifier $\forall$ can not be defined directly. Instead we define a substitution operation S^κ_λ. $S^\kappa_\lambda(F/\equiv)$ denotes the equivalence class of the formula which results from F by replacing each free occurrence of v_κ by v_λ. Here, it also becomes obvious why we needed an ordered sequence of variables. Extending the Boolean algebra above we obtain the **quantifier algebra of formulas**: $\langle \mathcal{F}^\Gamma / \equiv_\Theta, +, \cdot, \bar{}, \bot, \top, S^\kappa_\lambda, \exists_\kappa \rangle_{\kappa,\lambda<\alpha}$ associated with Θ.

In analogy to Boolean algebras we can now formally define **Quantifier Algebras** of degree α (QA_α). This is a structure $\mathcal{U} = \langle A, +, \cdot, \bar{}, 0, 1, S^\kappa_\lambda, \exists_\kappa \rangle_{\kappa,\lambda<\alpha}$ where $\langle A, +, \cdot, \bar{}, 0, 1 \rangle$ is a Boolean Algebra with the unary operators S^κ_λ and $\exists_\kappa$ which have the following properties for all $x, y \in A$ and $\kappa, \gamma, \lambda < \alpha, \alpha \geq 2$:

$$\begin{array}{ll} (q_1) & S^\kappa_\lambda(\overline{x}) = \overline{S^\kappa_\lambda(x)} \\ (q_2) & S^\kappa_\lambda(x+y) = S^\kappa_\lambda(x) + S^\kappa_\lambda(y) \\ (q_3) & S^\kappa_\kappa(x) = x \\ (q_4) & S^\kappa_\lambda S^\gamma_\kappa \\ (q_5) & \exists_\kappa(x+y) = \exists_\kappa x + \exists_\kappa y \\ (q_6) & x \leq \exists_\kappa x \\ (q_7) & S^\kappa_\lambda \exists_\kappa = \exists_\kappa \\ (q_8) & \exists_\kappa S^\kappa_\lambda = S^\kappa_\lambda \ if \ \kappa \neq \lambda \\ (q_9) & S^\kappa_\lambda \exists_\gamma = \exists_\gamma S^\kappa_\lambda \ if \ \gamma \neq \kappa, \lambda \end{array}$$

Due to the many indexes these equations may at first look difficult but if you have a closer look you will realize that you can group $(q_1) - (q_4)$ into simple substitution properties, $(q_5) - (q_6)$ into quantifier properties, and $(q_7) - (q_9)$ into a group which relate substitutions to quantifiers. We assume here that the interpretation of these equations is obvious.

If U is a quantifier algebra as defined above we can define a so called **dimension set** of a formula $a \in A$:

$$\diamond x = \{\kappa < \alpha : \forall \lambda \neq \kappa, \, S^\kappa_\lambda x \neq x\}$$

Quantitatively this is the set of all indexes κ of variables v_κ which would change the valuation of formula x if substituted with another variable v_λ. We define $\mathcal{U}$ to be **locally finite** if $\diamond x$ is a finite set.

It can be shown that every quantifier algebra of formulas is a **locally finite quantifier algebra** as defined above. Accordingly, one can show that if $\mathcal{U}$ is a locally finite quantifier algebra then there is a theory Θ such that $\mathcal{U}$ is isomorphic to a quantifier algebra of formulas which could be derived from Θ. This result is already very important as it implies that we can express every theory in first-order logic without equality by using locally finite quantifier algebras and thus sets up a link between algebra and logic. Conversely we can take a locally finite quantifier algebra and translate it into a first-oder logic without equality.

To make this link even stronger we will need to remove the limitations from above which restricted our first-order formulas to $\mathcal{F}^\Gamma / \equiv_\Theta$. Intuitively we will extend the quantifier algebras by another equivalence class. This is equivalent to extending Boolean algebra with substitution and existence equivalence classes. We define the equivalence class

of equality as $e_{\kappa\lambda}$ which contains all formulas $v_\kappa \equiv v_\lambda$. Finally, this defines **quantifier algebras with equality** as an algebra $\langle A, +, \cdot, -, 0, 1, S^\kappa_\lambda, \exists_\kappa, e_{\kappa\lambda}\rangle_{\kappa,\lambda<\alpha}$ for which $\mathcal{U} = \langle A, +, \cdot, -, 0, 1, S^\kappa_\lambda, \exists_\kappa\rangle_{\kappa,\lambda<\alpha}$ is a QA_α and $e_{\kappa\lambda}$ are distinguished elements which satisfy

$$(q_{10}) \qquad S^\kappa_\lambda e_{\kappa\lambda} = 1$$
$$(q_{11}) \qquad x \cdot e_{\kappa\lambda} \leq S^\kappa_\lambda x$$

Why can we be sure that this quantifier algebra with equality is finally an algebra which represents our first-order logic? Common practice is to search for an algebra from which we know that it is an algebraisation of first-order logic and show that quantifier algebra with equality is isomorphic to this algebra.

For this purpose we look at so called **cylindric algebras** defined by Henkin, Monk and Tarski in [13]. Cylindric algebras of degree α are systems with structure $\langle A, +, \cdot, -, 0, 1, \exists_\kappa, e_{\kappa\lambda}\rangle_{\kappa,\lambda<\alpha}$ such that $\langle A, +, \cdot, -, 0, 1\rangle$ is a Boolean algebra and $\exists_\kappa$ and $e_{\kappa\lambda}$ satisfy the following conditions $\forall x \in A$ and $\kappa, \lambda < \alpha$:

$$(c_1) \qquad \exists_\kappa 0 = 0$$
$$(c_2) \qquad x \leq \exists_\kappa x$$
$$(c_3) \qquad \exists_\kappa(x \cdot \exists_\kappa y) = \exists_\kappa x \cdot \exists_\kappa y$$
$$(c_4) \qquad \exists_\kappa \exists_\lambda = \exists_\lambda \exists_\kappa$$
$$(c_5) \qquad e_{\kappa\kappa} = 1$$
$$(c_6) \qquad e_{\lambda\gamma} = \exists_\kappa(e_{\lambda\kappa} \cdot e_{\kappa\gamma}) \ if \ \kappa \neq \gamma, \lambda$$
$$(c_7) \qquad \exists_\kappa(e_{\kappa\lambda} \cdot x) \cdot \exists_\kappa(e_{\kappa\lambda} \cdot \overline{x}) = 0 \ if \ \kappa \neq \lambda$$

If we look at this definition we will recognise that the substitution operation has disappeared. Above, this operation accounted for the $\forall$ quantifier in quantifier algebras with equation. However, we can define S^κ_λ as follows:

$$S^\kappa_\lambda x = x \ if \ \kappa = \lambda \quad ; \quad S^\kappa_\lambda x = \exists_\kappa(x \cdot e_{\kappa\lambda}) \ if \ \kappa \neq \lambda$$

This definition appears to be obvious. Of course, a formula which replaces a variable by the same variable is equivalent to its original version. Substitution with a different value can be achieved by assigning v_λ to v_κ if v_κ actually exists. With this definition and the translation (isomorphism) informally described here we can conclude that a cylindric algebra is also a quantifier algebra with equality. From [31] we additionally know that definitions $(q_1) - (q_{11})$ are actually all theorems of the theory of cylindric algebras as shown in [13].

It turns out that we now have a full mapping from cylindric algebra to first-order logic with equality. If we only consider first-order logic without equality Galler also shows in [32] that we can map quantifier algebra to **polyadic algebra** by slightly modifying substitution and existence operators in QA_α. By doing this we obtain a polyadic algebra defined by Halmos in [33]. As in section III-B.1 we now have the evidence that we can directly map first-order logic into algebra and vice versa. Daigneault's interpretation [34] of Krasner's general theory on Galois Fields [35] (see also section II-E) as polyadic algebras gives even more evidence to the relation of first-order logic and algebra. This implies that we can analyse effects in the domains which we can map to algebra in powerful first-order logic. One of the probably most exciting results is the relatively recent work of Andréka, Madaraász and Németi which used the relation between cylindric algebra and first-order logic to formulate Einstein's general theory of relativity in first-order logic [36].

Finally we want to refer the interested reader to the survey "Tarskian Algebraic Logic" by T.S. Ahmed who gives a good overview on algebraic logic, summarising its history and its different relations to other fields. This survey also contains a nice discussion of (n-ary) Cylindric Algebras and their mapping to first-order logic. Andréka et. al. focus in [37] more on the Tarskian structuralist view to logic and thus can also be considered to be a good complement to this section. In this work the relevant branch of **universal algebra** is also discussed. As a matter of fact universal algebra offers a framework which provides powerful tools and theories to investigate the interconnections between different classes of algebras.

C. Temporal Logic

Sections III-A.1 and III-A.2 quantitatively introduced propositional and first-order logic. Generally speaking these logics support the reasoning based on propositions or terms and formulas. The truth values are fixed and constant over time, this is, no matter when you evaluate a proposition or a first-order formula, the truth value will always be the same only depending on the valuation function, the propositions, and variables used.

Temporal logic extends the classical concept and introduces the dimension of time. Thus, this notion of logic will extend propositions with a reference to time conditions. Consequentially, compared with classical logic which can describe states and properties of systems, temporal logic is able to express sequences of state changes and properties of behaviour.

As we have seen in classical logic also temporal logic comprises different logics. Thus, propositional and first-order logic find their correspondence in propositional and first-order temporal logic.

To introduce the general idea of temporal logic we will shortly introduce linear temporal logic (LTL). As we will not perform a similar algebraisation as for propositional and first-order logic we only give an informal definition for temporal logic and shortly compare it and its variants to first-order logic. Based on these informal definitions we outline the link between temporal logic and algebras. For this purpose we also establish a link to universal algebras. Finally, this section will explain the differences between linear and branching time logic and conclude with a short list of applications of temporal logics.

1) Linear Time Logic: For this purpose we first define two new temporal operators on a set $\mathcal{P}$ of regular propositional formulas.

1) $\bigcirc Q$ is a linear temporal formula if $Q \in \mathcal{P}$
2) $Q \cup R$ is a linear temporal formula if $Q, R \in \mathcal{P}$

To give the symbols defined above some semantics we extend the regular valuation function from section III-A.1 as follows.

1) $v(\bigcirc Q) = 1$, iff in the next time step $v(Q) = 1$.
2) $v(Q \cup R) = 1$, iff $v(Q \wedge \neg R) = 1$ until $v(R) = 1$.

To explain these rather abstract definitions we illustrate them in figure 12. Arrows in this figure represent the time line. Single nodes represent points in time at which a proposition changes. Above each node you find the valuation of the corresponding formulas depending on the time they are evaluated.

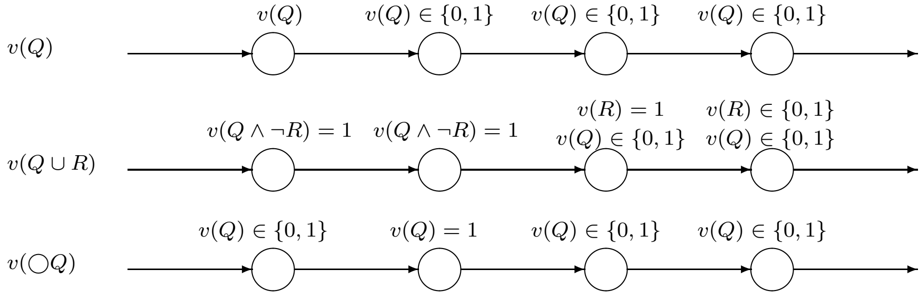


Fig. 12. Illustration of the semantics of LTL operators

Clearly, the choice of the type of illustration in Figure 12 was not arbitrary. By choosing a representation which resembles finite state machines we also wanted to emphasise the fact that Pnueli saw linear temporal logic as a tool to analyse computer programs [38].

The first linear sequence in this figure presents the semantics of a simple propositional formula Q. Its valuation is given for a specific point in time (here the first time step in the figure). If time is proceeding the valuation of this formula is arbitrary. For the second operation the formula $Q \vee \neg Q$ has to valuate to be true in the system. If this is true **until** $v(R) = 1$ the formula $Q \cup R$ valuates true for this sequence. The last formula $\bigcirc Q$ represents the simplest sequence as it only requires Q to valuate to $v(Q) = 1$ in the next time step of this sequence. With the given operators of propositional logic extended by $\cup$ (until) and $\bigcirc$ (next) and their semantics it is clear there are other temporal operators which can be derived. We give some example in the following list:

1) $\Diamond Q \equiv \top \cup Q, \ \ with \ Q \in \mathcal{P}$
2) $\Box Q \equiv \neg \Diamond \neg Q, \ \ with \ Q \in \mathcal{P}$

The semantics of these operators is already defined through the equivalence with formulas that use the operators we defined above. To clarify their meaning we again use the same illustration as in figure 12.

As $\Box$ depends on $\Diamond$ we start to look at the representation of $\Diamond Q$ as illustrated in Figure 13. Unfortunately this representation does not show that the valuation of Q does not have to turn to true after exactly three time steps. This can happen at any time such that $v(\Diamond Q) = 1$ if Q *eventually* evaluates to 1. If we look at the semantics of $\cup$ this behaviour becomes even more clear as the condition which has to hold has to hold before Q can come true is always true as it is the tautology $\top$. At the bottom of figure 13 the operator $\Box$ is illustrated. It turns out that this operator requires the formula it is associated to. With this knowledge we look back to our definition of $\Box Q$ and first consider the formula $\Box \neg Q$. As we learnt above this states that $\neg Q$ will eventually evaluate to true, this is, eventually Q evaluates to false. So if we negate this statement, as done in the definition above, we obtain that Q will not evaluate to false eventually which is equivalent to: “Proposition Q is *always* true”.

To show the power of this new approach we may use an example which is used in many lectures. Take a traffic light. With propositional logic you will be able to set up propositions about properties of is static characteristics, such as that the green and red light are not illuminated at the same time. Here you can statements about time because this characteristic is an invariant. However, you will not be able to describe that if the traffic light shows red light it will eventually turn into red light. In LTL you can express this state change with the formula $\Box(R \supset \Diamond G)$. Here R denotes the proposition “Red light” and G the proposition “Green light. Reading this formula in natural language yields: It is *always* true that if there is red light then *eventually* there will be green light.

Let us now look at the structure of linear temporal logic. To define it we took a classical propositional logic and extended it with two additional operators which introduced the time dimension. However, if we look at the definition of these operators then we could always formulate them with *there exists* a point in time or *for all* points in time. Thus, these definitions suggest themselves to ask whether we could model linear temporal logic in first order logic. In fact, it is possible but tedious because we have to model time in first-order logic and thus formulas become very complex and difficult to read. This is comparable to translating a higher programming language to assembler. However, the important result we should remember is that we can model all statements in propositional temporal logic using first-order logic. In [39] Etessami et al. go even one step further and prove that unary temporal logic (which is temporal logic with only unary temporal operators as defined above) can be expressed by first-order logic with only two variables. Consequentially, we can use the algebras developed in the last section to allow the algebraisation of unary-temporal logic.

One obvious question follows: Can formulas in first-order temporal logic be translated into first-order logic and thus is it possible to use the same algebras? The answer to this question would require the coverage of more theoretical concepts and we would need to extend our discussion of logic to completeness and other important theories. Therefore, for completeness we mention here that a lot of research has been conducted in this area. A good overview and introduction can be found in [40] and [41].

2) Branching Time Logic: So far linear temporal logics have been discussed. They get their name from the fact that they consider only behaviours which can be modelled as linear time sequences. This characteristic is nicely illustrated by figures 12 and 13. Every state, represented as a node, has exactly one successor. However, in communication systems or generally in concurrent systems a state in time will need to have several future states. To model such system branching time logics [42] have been proposed. They possess tree structure in which each state in time has more than one successor. One of the most popular of these logics is the computation tree logic (CTL) proposed in [43]. Although, it is usually easier to model concurrent systems using branching time logic. This is due to the fact that their additional path quantifiers

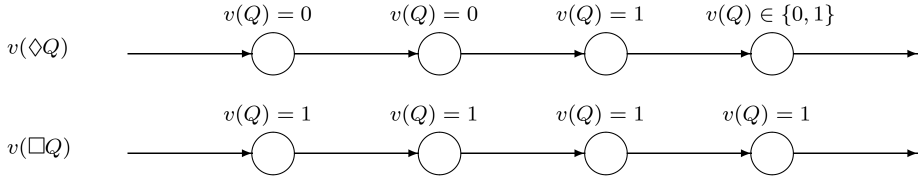


Fig. 13. Illustration of the semantics of derived LTL operators

usually support the navigation in their tree structure. However, sometimes it easier to use existing tools, proof techniques, and analytical methods which exist already in one domain. We can state here that branching temporal logics can be translated into linear temporal logic by simply modelling each branch in the branching logic as a linear sequence in linear temporal logic. This is common practice when, for example, translating non-deterministic machine models into deterministic ones.

D. Security

We described how temporal logic can extend classical PL to describe time-dependent system characteristics. This expressiveness can be used in many different ways. Very popular is the use of various types of temporal logic in the field of security. Here the logical formulas are used to describe expected behaviour. This behaviour could reflect certain security characteristics of a control flow of a computer program, of a security protocol, or a general access control mechanism. Accordingly, the number of applications of temporal logic in the field of security is huge.

Formal Specification has currently a strong focus in many areas of security research. Distributed computing systems, access control and software systems, security protocols, etc. may be subject to using logic for specifying their security characteristics [44] and was mainly induced by Pnueli [38], [45] and Lamport [46].

One important characteristic of classical and temporal logic is the existence of a proof calculus which is mainly based on the mathematical foundation of these logics in algebra. This calculus can be used to **formally verify** the correctness of system specifications based on logics [45], [46].

Specifying the security compliant operation of a system does not only require the thorough specification of its components. It is also required to thoroughly specify how the system is restricted and what the environment can or can not do. Formulas of (temporal) logic can be used to **describe** these **requirements**. However, even more important for the work this paper forms the basis for are the links between biology, algebra, and logic. As we intend to use algebra to study structures in biological systems with the special focus on genotypes we can exploit these links to conduct new and innovative research.

In BIONETS we are currently exploring characteristics and structure of and operations on **Fraglets** [47]. They represent a programming model which can be compared to the copying of DNA sequences. Their execution model is similar to interacting biological systems such as the DNA with the various enzymes that it generates. Due to this similarity, one may expect the implementation of genetic algorithms on these structures to be fairly easy. However, first experiments show that this is not the case [48]. Nevertheless, by choosing a programming model which is very similar to bio-chemical processes in a cell we hope to be able to transfer the observations made in the realm of biology to a programming language, i.e. we try to describe structures of Fraglets using algebra. These structures and their interaction with the environment correspond to specific program characteristics and behaviour. This process is comparable to specific structures of the DNA (genes) which in interaction with their environment yield different phenotypes of an organism. In this way, Fraglets and their similarity to bio-chemical processes form the exemplary bridge head of the application of our theory to security.

As explained in this sections algebras can be mapped into the realm of logic. Depending on the type of algebra we obtain from the analysis of our programming model we will have the possibility to analyse the corresponding characteristics in the realm of logic. As explained above, logic is a powerful means to investigate program properties, including security. Clearly, this process has its limitations as we will not be able to investigate any arbitrary program and thus security characteristic. However, it will be an important step towards understanding the complicated programming and execution model of Fraglets and possibly their counterpart in biology, the DNA and its proteins. Furthermore, if we invert this analytical process, it becomes clear how we could also guide program evolution. Being able to express specific program properties in logic, including some specific security properties, we will be able to express the same characteristic in algebra and thus as a structure of the programming language.

Consequently, the insights that we obtain from this bridge between biology and logic could also help to improve genetic operators and therewith basically any automated and autonomous code generation process. This is due to the fact that it would become easier to investigate the implications of a structural change (which corresponds to a genetic operation) on the program (security) properties. Finally, this would also have immediate effect on the design of fitness functions. Instead of evaluating program representations as a whole they could analyse their structure and evaluate only those parts that are relevant for 'survival'. This also allows for a better evaluation of security characteristics of evolved programs.

IV. Conclusion

The reason we are interested in the metabolism of the cell is that the cell can be considered an immensely complex parallel computer that executes a 'distributed algorithm'. This term arises from the fact that even though most of the instructions are coded in the DNA, a significant part of each metabolic cycle depends on the chemical composition of the cell moment-by-moment. The DNA instructions are propagated through the cell by diffusion mechanisms coupled with various reactions. The concentrations of the various chemical species are far from uniform. In addition, several kinds of membranes and structural elements separate areas of different chemical activities and make the internal topology of

the cell nested and extremely complex. There is however an aspect that greatly simplifies the conceptualisation of internal cell operations: dimensional reduction.

The most successful example of dimensional reduction is provided by the microcanonical ensemble of equilibrium statistical mechanics: an isolated system will approach equilibrium, which corresponds to the configuration of highest entropy. The configuration of highest entropy is, by definition, the most probable. Thus it can be easily identified by the peak in the frequency distribution of all possible configurations. For an isolated system in equilibrium this is nothing more than the familiar Gaussian, which has a very sharp peak indeed since it decays on both sides of the maximum like a square exponential. More generally, the Central Limit Theorem says that a sequence of random samples will converge to a Gaussian for equilibrium systems [49]. A more detailed discussion of these physics concepts and their relationship to self-organisation can be found in [5]. For our purposes here it is sufficient to recall how the CLT allows the derivation of stable macroscopic properties such as pressure from the random collisions of $O(10^{23})$ molecules in a litre of air. That's a dimensional reduction of 10^{23} to 1.

When a gene is activated and begins to signal to the cell machinery to fabricate a particular protein, it creates several thousand mRNA molecules that set an equal number of ribosomes to work in the cytoplasm (each cell has millions of ribosomes, or 'protein factories'). Such a large number of proteins will provide a high probability that the particular function the gene wants to execute will be executed. Therefore, we can regard the large numbers of molecules in the cell as a strategy to achieve a form of dimensional reduction that in computer science we generally call 'abstraction'. Several thousand proteins will participate in a relatively few biochemical reactions to advance one or more metabolic cycles one execution step. Even though the interior of the cell is never in equilibrium (it relies on its 'fall' toward equilibrium as the engine that drives all of its spontaneous self-organising processes—in fact, that's what 'spontaneous' means), its complex topology is divided into many areas in each of which a few reactions are active at any one point in time. From millions of elements we can therefore see how through a relatively small number of quasi-equilibrium regions of the cell several hundred metabolic cycles can be executed in parallel.

Dimensional reduction or abstraction working together with the fact that the DNA itself is composed of genes that can be ON or OFF makes it sound plausible that the internal working of the cell can be modelled through a discrete or digital framework. We can begin to recognise some of the concepts discussed in Section II. For example, the 4 DNA bases represent an alphabet with which the specification of proteins can be coded. The architecture of the DNA is such that it must not only carry the genetic code across generations but also support its expression through interactions with its environment. This has been achieved by replicating the same information along the two parallel strands of the DNA molecule, in such a way that the 4 bases are paired up two-by-two. In other words, of the four bases Thymine (T), Cytosine (C), Adenine (A), and Guanine(G) [50], only 2 kinds of pairs are possible: A-T and C-G, so that a binary base field might still be relevant in some way.

For example, as explained in [8] the DNA alphabet of 4 bases can be mapped to the finite field $GF(2^2)$. Since each codon of 3 bases can take on 4 values, the set of all possible codons, the DNA code, has 64 elements and can be represented as $GF(2^6)$. Although such a representation may seem arbitrary, by imposing a partial ordering based on the hydrophobicity of the amino acid each codon codes for, Sanchez et al. derive a Boolean lattice analogous to Fig. 11. As verified experimentally, this ordering reflects the robustness of the DNA code with respect to preservation of metabolic function in the presence of mutations. In mathematical terms, we could say that the partial ordering of the DNA code Boolean lattice approximates a continuous function of the most probable codon mutations. The most common mutations, in fact, have been observed to result in small steps along this lattice, which correspond to mutant amino acids with similar hydrophobicity. This, in turn, leads to similar protein folding characteristics. As a consequence, the resulting protein is in many cases able to perform the same function.

Sanchez et al. are AIDS researchers. Therefore, the robustness of the DNA code to mutations is obviously of great relevance (even if it is not very good news in this case) to the characteristics of the HIV virus. From our point of view, this provides an excellent example of how evolutionary processes have found a particular configuration of molecules that exhibit complex and robust behaviour at the same time–which can be formalised through algebra. In the case of the snowflake the mapping from physics to algebra is so strong as to be regarded as a cause-effect relationship. Even if in the case of the DNA code such a deterministic view does not seem appropriate, the fact that Evolution produced a working and stable system of unbelievable complexity some of whose structural and behavioural characteristics appear to be formalisable through algebra seems deeply significant and in our opinion warrants further scrutiny.

Assuming we can recognise an algebraic structure in the digital nature of cell biology, the same or similar structure would help us make a bridge to the structure of logic. So at this point we may ask the question: What can we do with logic and biology? As we have explained in Section III-B it is possible that by studying biological systems our research may yield interesting connections between these two, at first sight, completely different domains. For this purpose, we showed one of the numerous ways to map algebras into the realm of logic and vice versa. Logic, in turn, allows for the analysis and understanding of program properties, execution models and possibly their counterparts in biology, the DNA and its proteins. In this way, this document represents a first step in building a bridge from biological systems to security. Additionally, we briefly explained how this bridge could imply a significant contribution to the improvement of evolutionary computing paradigms, e.g. genetic programming or evolutionary programming.

The final goal is to be able to specify the security or other functional characteristics of a digital system, and have the specifications map to running code through a process analogous to gene expression, i.e. constructing order through interaction with the environment.

Acknowledgements

The authors wish to acknowledge valuable feedback from Luciana Pelusi and Marinella Petrocchi on the first draft of

this article, and the stimulating conversations with Sotiris Moschoyiannis and Christopher Alm. This research is funded by the EU OPAALS NoE (FP6-034824) and BIONETS IP (FP6-027748).

References

[1] E. Altman, P. Dini, D. Miorandi, and D. Schreckling, Eds., *D2.1.1: Paradigms and Foundations of BIONETS Research*. BIONETS, 2007. [Online]. Available: www.bionets.org

[2] P. Dini and D. Schreckling, "Notes on Abstract Algebra and Logic: Towards their Application to Cell Biology and Security," in *2nd International Conference on Digital Ecosystems and Technologies, IEEE-DEST 2008, 26-28 February*, 2008.

[3] S. Kauffman, *The Origins of Order: Self-Organisation and Selection in Evolution*. Oxford: Oxford University Press, 1993.

[4] P. Dini, *D4.1-DBE Science Vision*. www.digital-ecosystem.org, Deliverables: SP0: DBE Project, 2006.

[5] P. Dini and E. Berdou, *D18.1-Report on DBE-Specific Use Cases*. www.digital-ecosystem.org, Deliverables: SP5: DBE Project, 2004.

[6] P. Dini, *D18.4-Report on self-organisation from a dynamical systems and computer science viewpoint*. www.digital-ecosystem.org, Deliverables: SP5: DBE Project, 2007.

[7] ——, *D18.6-5-Year living roadmap for Digital Ecosystems research in biology-inspired computing*. www.digital-ecosystem.org, Deliverables: SP5: DBE Project, 2007.

[8] R. Sanchez, E. Morgado, and R. Grau, "Gene algebra from a genetic code algebraic structure," *Journal of Mathematical Biology*, vol. 51, pp. 431–457, 2005.

[9] N. Lynch, *Distributed Algorithms*. San Francisco, USA: Morgan Kaufmann, 1996.

[10] G. Boole, *The Mathematical Analysis of Logic*. Bristol: Thoemmes Press, 1998 (1847).

[11] P. R. Halmos, *Algebraic Logic*. AMS, 2007 (1962).

[12] A. Tarski, *Logic, Semantics, Metamathematics*. Oxford University Press, 1956, edited by J. H. Woodger.

[13] L. Henkin, J. D. Monk, and A. Tarski, *Cylindric Algebras*. Amsterdam: North-Holland, 1971.

[14] P. Cameron, *Introduction to Algebra*. Oxford: Oxford University Press, 1998.

[15] R. B. J. T. Allenby, *Rings, Fields and Groups: An Introduction to Abstract Algebra (2nd Ed.)*. Oxford: Butterworth-Heinemann, 1991.

[16] I. Stewart, *Galois Theory (2nd Ed.)*. London: Chapman and Hall, 1989.

[17] J. A. Gordon, "Very Simple Method to Find the Minimum Polynomial of an Arbitrary Non-Zero Element of a Finite Field," *Electronics Letters*, vol. 12, pp. 663–664, 1976.

[18] S. Moschoyiannis, *Group Theory and Error Detecting/Correcting Codes*. Guildford, UK: University of Surrey, Department of Computing, Technical Report SCOMP-TC-02-01, 2001.

[19] M. Purser, *Introduction to Error Correcting Codes*. Boston: Artech House, 1995.

[20] R. Sanchez, R. Grau, and E. Morgado, "A novel Lie algebra of the genetic code over the Galois field of four DNA bases," *Mathematical Biosciences*, vol. 202, pp. 156–174, 2006.

[21] Z. Tosic, "Analytical Representation of m-Valued Logical Functions over the Ring of Integers Modulo m," Ph.D. dissertation, University of Beograd, Beograd, Yugoslavia, 1972.

[22] W. G. Schneeweiss, *Zuverlässigkeitstheorie (Reliability Theory)*. Berlin: Springer, 1973.

[23] ——, "On the Polynomial Form of Boolean Functions: Derivations and Applications," *IEEE Trans. Comput.*, vol. 47, no. 2, pp. 217–221, 1998.

[24] B. J. Falkowski, "A Note on the Polynomial Form of Boolean Functions and Related Topics," *IEEE Trans. Comput.*, vol. 48, no. 8, pp. 860–864, 1999.

[25] D. Jovanovic and P. Janicic, "Logical analysis of hash functions." in *FroCos*, ser. Lecture Notes in Computer Science, B. Gramlich, Ed., vol. 3717. Springer, 2005, pp. 200–215.

[26] S. Morioka, Y. Katayama, and T. Yamane, "Towards Efficient Verification of Arithmetic Algorithms over Galois Fields GF(2m)," in *CAV '01: Proceedings of the 13th International Conference on Computer Aided Verification*. London, UK: Springer-Verlag, 2001, pp. 465–477.

[27] J. C. Agudelo and W. A. Carnielli, "Unconventional Models of Computation Through Non-standard Logic Circuits," in *UC*, ser. Lecture Notes in Computer Science, S. G. Akl, C. S. Calude, M. J. Dinneen, G. Rozenberg, and T. Wareham, Eds., vol. 4618. Springer, 2007, pp. 29–40.

[28] D. Mukhopadhyay, G. Sengar, and D. R. Chowdhury, "Hierarchical Verification of Galois Field Circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuites and Systems*, vol. 26, no. 10, pp. 1893–1898, 2007.

[29] M. Fitting, *First-order logic and automated theorem proving (2nd ed.)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1996.

[30] P. Halmos and S. Givant, *Logic as Algebra*. Washington, DC: The Mathematical Association of America, 1998, vol. 21 of Dolciani Mathematical Expositions.

[31] C. C. Pinter, "A Simple Algebra of First Order Logic," *Notre Dame Journal of Fomal Logic*, vol. XIV, no. 3, July 1973.

[32] B. A. Galler, "Cylindric and polyadic algebras," in *Proceedings of the American Mathematical Society*, vol. 8, no. 1, 1957, pp. 176–183.

[33] P. R. Halmos, "Algebraic logic, i. monadic boolean algebras," *Composito Mathematica*, vol. 12, pp. 217–249, 1956.

[34] A. Daigneault, "On automorphisms of polyadic algebras," *Transactions of the American Mathematical Society*, vol. 112, no. 1, pp. 84–130, 1964.

[35] M. Krasner, "Généralisation abstraite de la théorie de Galois," in *Algèbre et théorie des nombres*, ser. C.N.R.S. Paris: Centre national de la Recherche scientifique, 1950, no. 24, pp. 163–168.

[36] H. Andréka, I. Németic, and J. Madarász, "On the logical structure of relativity theories." Electronically available at http://www.math-inst.hu/pub/algebraic-logic/Contents.html, Tech. Rep., 2002.

[37] H. Andréka, J. X. Madarász, and I. Németi, "Algebras of relations of various ranks, some current trends and applications," *JoRMiCS*, vol. 1, pp. 27–49, 2004.

[38] A. Pnueli, "The temporal logic of programs," in *Proceedings of the 18th IEEE Symposium on Foundations of Computer Science*, 1977, pp. 46–57.

[39] K. Etessami, M. Y. Vardi, and T. Wilke, "First-order logic with two variables and unary temporal logic," *Information and Computation*, vol. 179, no. 2, pp. 279–295, December 2002. [Online]. Available: http://dx.doi.org/10.1006/inco.2001.2953

[40] M. Gehrke and Y. Venema, *Algebraic Tools for Modal Logic*. Electronically available at http://www.helsinki.fi/esslli/courses/readers/K15.pdf, Helsinki, Finland: European Summer School in Logic, Language and Information, 2001.

[41] P. Blackburn, M. de Rijke, and Y. Venema, *Modal Logic*. Cambridge University Press, 2001.

[42] M. Ben-Ari, Z. Manna, and A. Pnueli, "The temporal logic of branching time," in *POPL '81: Proceedings of the 8th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*. New York, NY, USA: ACM Press, 1981, pp. 164–176.

[43] E. M. Clarke, E. A. Emerson, and A. P. Sistla, "Automatic verification of finite state concurrent system using temporal logic specifications: a practical approach," in *POPL '83: Proceedings of the 10th ACM SIGACT-SIGPLAN symposium on Principles of programming languages*. New York, NY, USA: ACM Press, 1983, pp. 117–126.

[44] F. Kröger, *Temporal logic of programs*. New York, NY, USA: Springer-Verlag New York, Inc., 1987.

[45] Z. Manna and A. Pnueli, *The temporal logic of reactive and concurrent systems*. New York, NY, USA: Springer-Verlag New York, Inc., 1992.

[46] L. Lamport, "The Temporal Logic of Actions," *ACM Transactions on Programming Languages and Systems*, vol. 16, no. 3, pp. 872–923, May 1994. [Online]. Available: citeseer.ist.psu.edu/lamport94temporal.html

[47] C. Tschudin, "Fraglets - a metabolistic execution model for communication protocols," in *2nd Annual Symposium on Autonomous Intelligent Networks and Systems (AINS)*, Menlo Park, USA, July 2003.

[48] L. Yamamoto, D. Schreckling, and T. Meyer, "Self-Replication and Self-Modifying Programs in Fraglets," in *Proceedings of IEEE/ACM Bionetics 2007, Budapest, Hungary*, December 2007.

[49] D. Sornette, *Critical Phenomena in Natural Sciences: Chaos, Fractals, Self-organization and Disorder - Concepts and Tools*. Heidelberg: Springer, 2000.

[50] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter, *Molecular Biology of the Cell (4th Ed.)*. New York: Garland Science, 2002.

An efficient algorithm for conceptual subspace clustering

Huaiguo Fu
Telecommunications Software & Systems Group
Waterford Institute of Technology, Waterford, Ireland.
hfu@yahoo.com

Abstract

Cluster analysis is one of the primary tasks of data mining and Web mining. Due to large amounts of data on the Web continue to grow inexorably in size and complexity, the techniques and approaches of cluster analysis suffer from the challenges such as high-dimensional data clustering and description of too many clusters. In this paper, we propose a novel algorithm of cluster analysis to address these challenges. The algorithm extracts clusters in dense subspaces and the clusters can be described by overlapping concepts. The experimental results show the efficiency of this algorithm.

Keywords: *Algorithm, Clustering, Concept lattice, Conceptual clustering, Subspace clustering*

1 Introduction

Information searching on the Web is a basic service on the Web. But the searching results are often mass and confused for users. Users often get too much information by Web search engines and they have to sift through so long ordered list of information that they waste too much time and can not find the exact information that they need. Some techniques such as clustering can be used for information searching and post-searching to categorize information, clarify and represent knowledge to users, and users can easily locate the desired information. Especially, the techniques of hierarchical clustering can help us to generate the topic hierarchy for similar documents so that users are able to capture the "meaning" of a set of information or searching results.

Cluster analysis is unsupervised learning and one of the primary tasks of data mining. There are divers algorithms and approaches of cluster analysis. Due to large amounts of data continue to grow inexorably in size and complexity, the techniques and approaches of cluster analysis suffer from the challenges such as high-dimensional data clustering, complex data clustering and description of too many clusters. For example, it is hard for clustering very high-dimensional data of proteins or genes. The purpose of clustering of data is to extract the similarity between objects to gain better understanding of the data. However, description and interpretation of clusters of data is one of main issues of applications of the most clustering techniques, especially clustering often obtains large numbers of clusters. We investigate these problems and propose the integration of techniques of subspace clustering and conceptual clustering to address these challenges.

Subspace clustering [1, 5, 10]is a strategy to reduce the complexity of clustering high-dimensional data. Subspace clustering algorithms can divide data space or search space of clusters into subspaces and find clusters in different subspaces within a dataset. All clusters of whole dataset are included in all subspaces. Some subspaces include more interesting clusters but others that contain less interesting clusters or noisy data can be pruned.

Conceptual clustering [6, 8, 3] can seek clusters by concept structures so that the concept can describe the clusters. One approach of conceptual clustering is based on concept lattice [4]. The structure of concept lattice can be used to generate overlapping concepts. Theoretical foundation of concept lattice founds on the mathematical lattice theory [2, 7]. Lattice is a popular mathematical structure for modeling conceptual hierarchies. Concept lattice is a method for deriving conceptual structures out of data. It allows us to analyze and mine the complex data for such as classification, association rules mining, clustering, etc [9]. Furthermore, concept lattice also provides an effective tool of knowledge visualization that plays an important role in data mining. Concept lattice can facilitate pattern understanding, knowledge discovery, and interactive data exploration. The application of concept lattice has been an area of active and promising research in various fields such as knowledge discovery, information retrieval, software engineering and machine learning.

This paper integrates the techniques of subspace clustering and conceptual clustering and proposes a novel algorithm, called CSC, to extract conceptual clusters in dense

	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
1	×	×					×	
2	×	×					×	×
3	×	×	×				×	×
4	×		×				×	×
5	×	×		×		×		
6	×	×	×	×		×		
7	×		×	×	×			
8	×		×	×		×		

Figure 1. Example of formal context

subspaces and give the description of clusters by overlapping concepts. The algorithm CSC selects subspaces based on the density of dataset and exploits hierarchical overlapping clusters that can be described by concepts. The experimental results show the efficiency of this algorithm in this paper.

The rest of this paper is organized as follows. The basic concepts of concept lattice are presented in the next section. Section 3 analyzes the search space of conceptual clustering. The algorithm CSC is introduced in section 4. Section 5 presents the experimental results of the algorithm. The paper ends with a short conclusion in section 6.

2 Concept lattice

The core of Formal Concept Analysis (FCA) is concept lattice. FCA provides a natural platform for data analysis and knowledge representation. FCA is different from some of the traditional, statistical means of data analysis and knowledge representation because of its focus on human-centered approaches. Concept lattice facilitates exploring, searching, recognizing, identifying, analyzing, visualizing, restructuring and memorizing conceptual structures [11]. So we can take advantage of the features of concept lattice to extract and interpret the frequent patterns and clusters of data set.

In this section, we introduce some basic notions of concept lattice.

Definition 2.1 **Formal context** *is defined by a triple* (O, A, R)*, where O and A are two sets, and R is a relation between O and A. The elements of O are called objects or transactions, while the elements of A are called items or attributes.*

For example, Figure 1 represents a formal context (O, A, R). $O = \{1, 2, 3, 4, 5, 6, 7, 8\}$ is the set of objects, and $A = \{a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8\}$ is the set of attributes. The crosses in the table describe the relation R of O and A.

Definition 2.2 *Two* **closure operators** *are defined as* $O_1 \rightarrow O_1''$ *for set O and* $A_1 \rightarrow A_1''$ *for set A.*

$$O_1' := \{a \in A \mid oRa \textit{ for all } o \in O_1\}$$

$$A_1' := \{o \in O \mid oRa \textit{ for all } a \in A_1\}$$

These two operators are called the **Galois connection** for (O, A, R). These operators are used to determine a formal concept.

Definition 2.3 *A* **formal concept** *of* (O, A, R) *is a pair* (O_1, A_1) *with* $O_1 \subseteq O$*,* $A_1 \subseteq A, O_1 = A_1'$ *and* $A_1 = O_1'$*.* O_1 *is called extent,* A_1 *is called intent.*

Definition 2.4 *We say that there is a hierarchical order between two formal concepts* $(O_1,\ A_1)$ *and* $(O_2,\ A_2)$*, if* $O_1 \subset O_2$ *(or* $A_2 \subset A_1$*). And we say* $(O_1,\ A_1)$ *is the sub-concept of* $(O_2,\ A_2)$*, if there is no concept* $(O_3,\ A_3)$ *,* $A_2 \subset A_3 \subset A_1$ *or* $O_1 \subset O_3 \subset O_2$*.*

All formal concepts with the hierarchical order of concepts form a complete lattice called **concept lattice***.*

All concepts and concept lattice of formal context in Figure 1 are illustrated in Figure 2.

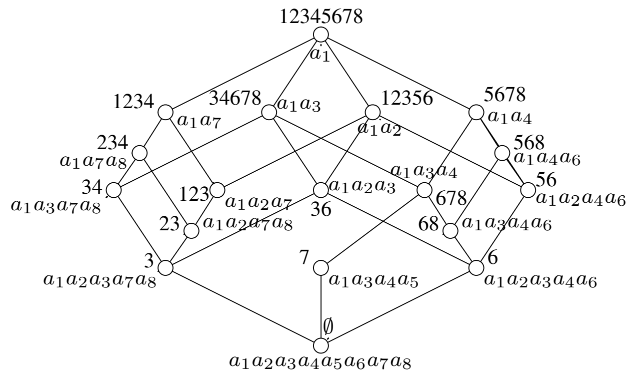


Figure 2. An example of concept lattice

3 Analysis of search space of conceptual clustering

We consider one formal concept as a cluster to analyze the search space of conceptual clustering. For one concept, the extent is an overlapping cluster, and the intent is the description of the cluster. Given a concept (O_i, A_i) of (O, A, R), we note sub-concept $(O_{i_{sub}}, A_{i_{sub}})$ of (O_i, A_i). We have $O_{i_{sub}} \subset O_i$ and $A_i \subset A_{i_{sub}}$ by the definition 2.4. (O_i, A_i) is more general than $(O_{i_{sub}}, A_{i_{sub}})$ in conceptual

category. O_i is more dense than $O_{i_{sub}}$. We observe that the hierarchical order of cluster based on lattice structure is also hierarchical order of density of clusters.

Lemma 3.1 *$(O_{i_{sub}}, A_{i_{sub}} - A_i)$ is a concept in the sub-context $(O_i, A - A_i, R)$*

Proof : $(O_{i_{sub}}, A_{i_{sub}})$ is a sub-concept of concept (O_i, A_i), then we have $O_{i_{sub}} = (A_{i_{sub}} - A_i)'$ and $O'_{i_{sub}} = (A_{i_{sub}} - A_i)$ in the sub-context $(O_i, A - A_i, R)$. Thus, $(O_{i_{sub}}, A_{i_{sub}} - A_i)$ is a concept in the sub-context $(O_i, A - A_i, R)$.

Lemma 3.2 *All sub-concepts of (O_i, A_i) can be generated from the sub-context $(O_i, A - A_i, R)$.*

Proof : Given a concept (O_l, A_l) of the sub-context $(O_i, A - A_i, R)$, $(O_l, A_l \cup A_i)$ is a sub-concept of (O_i, A_i). Thus all sub-concepts can be generated by the concepts of the sub-context $(O_i, A - A_i, R)$.

In fact, when we analyze the the hierarchical relation between one concept and its sub-concepts, we can simplify the intent of the concept. All the sub-concepts of (O_i, A_i) have common attribute set A_i, so we can consider A_i as one merging attribute, we note $a_i = A_i$ in this case. We give the following definitions to simplify the sub-context and reduce the complexity of searching clusters.

Definition 3.3 *Given an attribute a_i of context (O, A, R), $(\{a_i\}', A - \{a_i\}'', R)$ is called* **projective context** *of a_i.*

Definition 3.4 *Given a concept (O_i, A_i) of context (O, A, R), we can merge all attributes of A_i as a new attribute a_{new}, the projective context of a_{new} is called* **projective context** *of the concept (O_i, A_i).*

Definition 3.5 *Given a context (O, A, R), an attribute a_i is called* **trunk attribute***, if attribute $a_i \in A$, for all $a_j \in A$, $i \neq j$ and $\{a_i\}' \not\subset \{a_j\}'$.*

From lemma 3.1 and the definition 3.4 and 3.5 we can deduce the following lemma.

Lemma 3.6 *Given a concept (O_i, A_i) of context (O, A, R), the projective context of (O_i, A_i) is $(O_i, A - A_i, R)$. If attribute $a_i \in A - A_i$ and a_i is a trunk attribute, then $(\{a_i\}', \{a_i\}'' \cup A_i)$ is a sub-concept of (O_i, A_i).*

Therefore, the projective context of one concept is a subspace for searching cluster. From top concept or cluster, we can search its sub-concepts or sub-clusters in the subspace of clustering. All clusters can be generated in different subspaces of clustering by the same iterative approach.

In many cases, we need not generate all clusters but we are interested in the dense clusters. Thus, we give the following definition to reduce the subspaces of clustering and generate more general concepts or dense clusters.

Definition 3.7 *Given m = the minimum number of objects of cluster, a formal context is called* **ordered dense context** *if we only choose the attributes which number of objects is not less than m, and order these attributes of formal context by number of objects of each attribute from the smallest to the biggest one, and the attribute with the same objects are merged as one attribute. We note ordered dense context $(O, A^{\lhd}, R)_m$ of the formal context (O, A, R).*

Lemma 3.8 *Given ordered dense context $(O, A^{\lhd}, R)_m$, if attribute $a_i \in A^{\lhd}$ and a_i is a trunk attribute, then $(\{a_i\}', \{a_i\})$ is a concept.*

Lemma 3.8 shows us a new strategy to find concpets: we only need know which attribute is trunk attribute in an ordered dense context.

Ordered dense context can reduce the complexity of clustering high-dimensional data. For each subspace of clustering, we can deal with the projective context as a ordered dense context and then search for the clusters. By the hierarchical order, the clusters can be found from more dense subspaces to less dense subspaces of clustering. This is the key idea of the algorithm CSC.

4 Novel algorithm: CSC

In this section, we present the principle of the novel algorithm CSC (conceptual subspace clustering).

4.1 The principle of algorithm CSC

Given a formal context and minimum density of clustering, we propose the algorithm to generate dense hierarchical conceptual clusters by following steps:

1) First of all, the algorithm needs to generate the ordered dense context $(O, A^{\lhd}, R)_m$.

2) The second step: searching for trunk attributes in the ordered dense context, every trunk attribute a_i forms a concept or conceptual cluster: $(\{a_i\}', \{a_i\})$. $\{a_i\}'$ is cluster, a set of objects that have common attribute $\{a_i\}$ to describe the cluster.

3) For each trunk attribute, we generate the projective context of the trunk attribute, then an ordered dense context. For each ordered dense context, we use the same way as step 2 to find the clusters in the new context. We can continue to generate clusters until the number of objects for all trunk attributes is less than the minimum density of clustering.

4) At the end, we can get all dense hierarchical conceptual clusters. For example, given the minimum density of clustering 2, the algorithm generates all dense hierarchical conceptual clusters(see Figure 3) from the formal context of Figure 1.

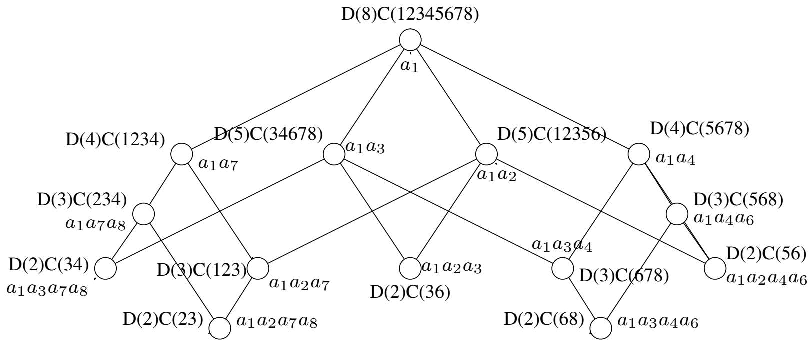


Figure 3. An example: dense hierarchical conceptual clusters mining with algorithm CSC. (minimum density of clustering is 2; for each node, D means density of cluster, C means cluster)

Algorithm 1 Algorithm of CSC (Conceptual Subspace Clustering)

input: context (O, A, R)
input: $m =$ the minimum density of clustering
output: hierarchical conceptual clusters
generate $ODC =$ ordered dense context $(O, A^{\lhd}, R)_m$
$D = ||O||$ //D is the density of cluter
if $O' \neq \emptyset$ **then**
  (O, O') is a conceptual cluster
  $ODC = (O, (A - O')^{\lhd}, R)_m$
end if
while $D >= m$ **do**
  for all levels of the projective context **do**
    for all trunk attribute a_i **do**
      $(\{a_i\}', \{a_i\})$ is a conceptual cluster //generate sub-concept in ODC
      generate projective context of $(\{a_i\}', \{a_i\})$
      $ODC =$ new ordered dense context
    end for
  end for
  $D = ||O||$ // in ODC
end while

4.2 An example

For the formal context of Figure 1, given the minimum density of clustering = 2, we show how to use the algorithm to generate all conceptual clusters.

For the first step, the algorithm generates the ordered dense context (see Figure 4) based on minimum density of clustering 2. a_5 is not included in the ordered frequent context because the number of objects of a_5 is less than 2.

	a_8	a_6	a_7	a_4	a_3	a_2	a_1
1			×			×	×
2	×		×			×	×
3	×		×		×	×	×
4	×		×		×		×
5		×		×		×	×
6		×		×	×	×	×
7				×	×		×
8		×		×	×		×

Figure 4. Ordered dense context of the formal context of Figure 1

From the ordered dense context, only a_1 is the trunk attribute. Hence, $\{a_1\}$ forms a concept $(\{1, 2, 3, 4, 5, 6, 7, 8)\}, \{a_1\})$. In figure 3, the first node is represented by $D(8)C(123\ 45678)$ and a_1, D(8) means the density of cluster is 8 and C(12345678) means the cluster is $\{1, 2, 3, 4, 5, 6, 7, 8\}$.

The density of a_1 is bigger than 2, so we consider the ordered dense context of the projective context of a_1 (see Figure 5). By the same way, we can generate all sub-concepts of $(\{1, 2, 3, 4, 5, 6, 7, 8)\}, \{a_1\})$: $\{D(4)C(1234)$, $a_1a_7\}$, $\{D(5)C($ 34678), $a_1a_3\}$, $\{D(5)C(12356)$, $a_1a_2\}$,

$\{D(4)C(\ 5678),\ a_1a_4\}$, as a_7, a_3, a_2 and a_4 are trunk attributes in the ordered dense context of Figure 5.

	a_8	a_6	a_7	a_4	a_3	a_2
1			×			×
2	×		×			×
3	×		×		×	×
4	×		×		×	
5		×		×		×
6		×		×	×	×
7				×	×	
8		×		×	×	

Figure 5. Ordered dense context of the projective context of a_1

If the density of the trunk attribute is 2, this attribute ends to generate sub-concepts of next level. Otherwise for the next level, we use the same approach to generate all concepts or dense hierarchical conceptual clusters (see Figure 3).

5 Experimental results

We have implemented the algorithm CSC in Java to generate dense hierarchical conceptual clusters. We test the algorithm on some real data and simulation data. In this paper we show the experimental results on the datasets of machine learning benchmarks: UCI repository (see table 1). In table 1, Objects means the number of objects, Attributes means the number of attributes and Clusters means the number of overlapping clusters (concepts) for each dataset.

Figure 6 shows the experimental results of the algorithm CSC on 12 datasets with different density threshold.

6 Conclusion and further work

Clustering can categorize information for searching and post-searching information on the Web to help users to find similar information. This paper presents the algorithm CSC for subspace conceptual clustering. The principle of the algorithm is to generate dense ordered context to reduce the complexity of subspace clustering. And then the clusters or concepts are easy to be mined. The results of mining include the overlapping conceptual clusters and the description of clusters. The experimental results illustrate the efficiency of this algorithm.

We will compare the performance of CSC with other clustering algorithms, and apply the algorithm to high-dimensional data clustering on the Web.

7 Acknowledgements

This work is supported by the project of EU IST Network of Excellence ”OPAALS”.

References

[1] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *ACM SIGMOD*, pages 94–105, 1998.

[2] G. Birkhoff. *Lattice Theory*. American Mathematical Society, Providence, RI, 3rd edition, 1967.

[3] G. Biswas, J. B. Weinberg, and D. H. Fisher. Iterate: A conceptual clustering algorithm for data mining. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, (28):100C111, 1998.

[4] C. Carpineto and G. Romano. Galois: An order-theoretic approach to conceptual clustering. In *Proceedings of ICML'93*, pages 33–40, Amherst, Juillet 1993.

[5] C. H. Cheng, A. W.-C. Fu, and Y. Zhang. Entropy-based subspace clustering for mining numerical data. In *Knowledge Discovery and Data Mining*, pages 84–93, 1999.

[6] D. H. Fisher. Knowledge acquisition via incremental conceptual clustering. *Machine Learning*, (2):139–172, 1987.

[7] B. Ganter and R. Wille. *Formal Concept Analysis. Mathematical Foundations*. Springer, 1999.

[8] M. Lebowitz. Experiments with incremental concept formation: Unimem. *Machine Learning*, (2):103–138, 1987.

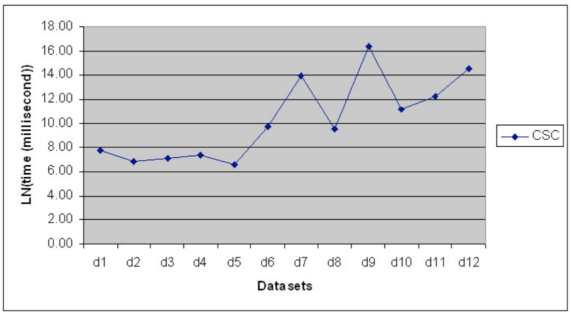
[9] E. Mephu Nguifo, M. Liquiere, and V. Duquenne. *JETAI Special Issue on Concept Lattice for KDD*. Taylor and Francis, 2002.

[10] L. Parsons, E. Haque, and H. Liu. Subspace clustering for high dimensional data: a review. *SIGKDD Explorations*, 6(1):90–105, 2004.

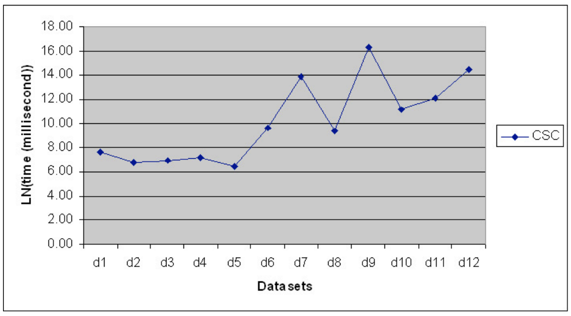
[11] U. Priss. Formal concept analysis in information science. *Cronin, Blaise (ed.),Annual Review of Information Science and Technology*, 40:521–543, 0062.

DataSet	ID	Objects	Attributes	Clusters
breast-cancer-wisconsin	d01	699	110	9860
house-votes-84	d02	435	18	10642
SPECT_test	d03	187	23	14532
SPECT_two	d04	267	23	21548
audiology.standardized	d05	26	110	30401
tic-tac-toe	d06	958	29	59503
nursery	d07	12960	31	147577
lung-cancer	d08	32	228	186092
agaricus-lepiota	d09	8124	124	227594
promoters	d10	106	228	304385
soybean-large	d11	307	133	806030
dermatogogy	d12	366	130	1484088

Table 1. The datasets of real data for experiments



The minimum density of clustering: 10%



The minimum density of clustering: 20%

Figure 6. Experimental results of CSC

An Examination of the Use of Inner Source in Multinational Corporations.

A Preliminary Framework to Understand Inner Source Software Development

[OPAALS Conference 2007]

Gary Gaughan, Brian Fitzgerald, Lorraine Morgan
Lero-The Irish Software Engineering Research Centre
University of Limerick,
Limerick, Ireland
Gary.Gaughan, Brian.Fitzgerald, Lorraine.Morgan
{@lero.ie}

Maha Shaikh
Lero-The Irish Software Engineering Research Centre
University of Limerick,
Limerick, Ireland
Maha.Shaikh@lero.ie
London School of Economics and Political Science
Information Systems and Innovation Group
London, United Kingdom
M.I.Shaikh@lse.ac.uk

Abstract — **This paper outlines the phenomenon of Inner Source software development and places Inner Source in the context of both existing open source literature and the ongoing work in the OPAALS community. Our study includes an analysis of three case studies of Inner Source in use in large scale global software development companies: IBM, Hewlett Packard and Bell Labs. The lessons learned from these case studies help us to contrast traditional open source principles with Inner Source principles, and we then gather the lessons to create our preliminary framework to make sense of when and how firms can adopt Inner Source. Our framework helps to make sense of the practical issues of managing and organizing Inner Source. These areas were chosen both for the background relevance to open source and emerging trends in Inner Source and also for the pertinence to the OPAALS research community. We have highlighted the emerging trends in the Inner Source phenomenon and surrounding areas. Awareness of this may be of great benefit to the work of researchers contributing to the OPAALS community both in terms of sustainable community building and in terms of collaboration methods within the community.**

Keywords – Open Source; Inner Source; OSS 2.0; Global Software Development

I. INTRODUCTION

In this paper we outline the principles of Inner Source extracted from our initial examination of emerging case studies and surrounding literature. Responding to the research challenges posed by Herbsleb (2007) we aim to contribute to the call to make better sense of when and how to orchestrate global development. Our detailed review of literature to date and in particular our focus on three case studies enable us to formulate a preliminary framework to help firms make sense of Inner Source, and whether it could be an appropriate choice for them in their global development scenario. The starting point for Inner Source is an understanding of open source and its many benefits and principles. The benefits of open source (Raymond, 1999) are varying and relative to each organization. More and more we see large organizations focus heavily on open source (Dinkelacker et al 2002). We can see this focus not only by the amount of large corporate investment directed into open source projects[1], (FLOSS Report, 2006) but also by the adoption of open source methods and practices behind corporate closed doors (Gurbani et al, 2005). These aspects will be examined further in Phase II of OPAALS. Phase II will include a more extensive examination based on the emerging questions and issues raised as a result of the work in this paper.

Wikipedia is often cited as an excellent business case where decentralized collaboration can be a very powerful tool. A word of caution obviously (Wikipedia is often held as a modern example of the inaccuracy of popular internet culture) is required as Wikipedia has been subject to vandalism and prejudice and is an example of how a tool can be powerful if used correctly but if used without question and control, it can almost destroy any benefits or credibility.

While open source may not represent a real paradigm shift in software development, it is an extremely successful exemplar of globally distributed development (GSD) and has many other benefits. In particular, in the current climate of outsourcing and off shoring, the open source development model is attracting considerable attention as organizations seek to emulate open source success in traditional development projects, through initiatives variously labeled as 'inner source', 'corporate source' (Dinkelacker et al, 2002) or 'community source'. It has been argued that open source communities can be considered in some cases to be virtual organizations (Crowston, Scozzi 2002; Gallivan 2001). If we

[1] "Firms have invested an estimated Euro 1.2 billion in developing FLOSS software that is made freely available. Such firms represent in total at least 565 000 jobs and Euro 263 billion in annual revenue." (FLOSS Report, 2006)

consider this assertion to be true then the possibility of being able to transfer the success and benefits of open source to corporate organizations becomes more real.

Inner source, as we use the term here, is any leveraging of open source development methods within a corporate environment. A variant of this term is progressive open source (and within it Inner Source) (Dinkelacker *et al* 2001). Inner Source is usually employed by companies to capitalize on the success that certain open source projects have enjoyed. However, there are some differences between open source and closed source development and their respective development communities. In traditional software development users and developers are typically located in separate departments and locations. This can lead to employees being unaware of development innovations and projects, and all too frequently to little mutual respect or voluntary interaction. The user-developer relationship has typically been quite different in open source. While early open source developers were users of actual products, as OSS has developed the situation has changed somewhat. In the absence of a traditional software development company, users need to become more intimately involved in the development process, as technical staff cannot simply send a checklist of requirements to the vendor to ascertain if needs will be met. It is a widely held belief that deploying open source can lead to a sense of shared adventure which is not a common scenario in the proprietary software arena (Raymond, 2001). The transfer of these lessons, where appropriate, to conventional in-house software development is a critical activity and is of much interest to these researchers and we are eager to leverage the lessons learned here for the OPAALS community.

We feel that just as the principles of open source (Raymond, 2001) have helped in understanding, defining and implementing open source, one of the most beneficial (to the OPAALS research community, the larger research community and to companies interested in Inner Source) places to start is with the contrast between open source principles and inner source principles.

This paper will outline a number of case studies (Bell Labs, HP and IBM) to give practical grounding in what initiatives are ongoing in the area. While the case studies have a lot of similarities in that they are for the most part positive experiences of Inner Source, this is probably due to firstly, a low adoption rate in a new emerging technology; and secondly, perhaps an unwillingness to publicize the failures of initiatives within companies. We then set about examining two particular fundamental areas of Inner Source managing and organizing Inner Source and the principles of Inner Source. We examine the important issues of managing and organizing an inner source solution which as we note require significant cultural change. Against this background we then present Inner Source principles gathered from the case studies and other relevant literature. The initial understanding and definition of Inner Source is examined and refined throughout the sections of this document. To conclude we present areas of future research we intend to undertake in Phase II.

II. Methodology

Our interpretive (Walsham 1993; 1995; Lee 1994) study includes three case studies (Yin 2002) of large global companies, Bell Labs, Hewlett Packard and IBM. Case study research is appropriate when a phenomenon is complex, when a holistic, in-depth investigation is needed, and when a phenomenon cannot be studied outside the context in which it occurs (Benbasat et al. 1987, Bonoma, 1985 and Yin, 2002). This paper is based on Deliverable 8.2 of the FP6 Network of Excellence Project OPAALS (Open Philosophies for Associative Autopoietic Digital Ecosystems). Similar to the aims of this deliverable, we too intend to get a better understanding of Inner Source use in large companies. In Phase II of OPAALS we will concentrate on collecting primary data from a number of companies but in order to do so we want to build a strong initial framework for analysis. This will then guide our data collection and analysis in Phase II.

We reviewed both academic and industrial literature to confirm which companies were using Inner Source ideas. A thorough review of the literature and our search brought a number of companies to our attention but we wanted to focus on only those firms that had implemented Inner Source for two years or more and where there was in-depth literature available on their experience. For a number of reasons most of the literature to date has focused on somewhat successful cases of Inner Source use. This then required us to study the documents and literature very carefully to reach an objective and balanced understanding.

Seminal work on each of the cases was identified and references followed up in academic work. Industrial reports and news articles on the same companies were scrutinized. Our data collection and data analysis stages were iterative. We built criteria for analyzing literature based on a number of categories like our definition of Inner Source, length of project duration in the company, domain area of the projects, and the size of the company (our focus is on multinational companies which tend to large in size). This list evolved over the early part of our data collection and analysis. New categories emerged over our study and we finally decided on three main cases based on our criteria and the recognition that only some cases had been academically analyzed with any detail. This provided us with a natural way to end our search and focus on three main cases.

Our analysis of the documents allowed us to immerse ourselves into the context of the company and we began to notice certain characteristics and issues for each case. There were similarities in all three cases yet each was also distinctive enough to allow our framework of principles to have wider appeal and use.

III. Case Studies

Examples of Inner Source deployed on a global scale lend themselves to a point of contrast with traditional software development, thus making Inner Source principles easier to extract and define. It is this contrast, while not always an absolute truth (there are always exceptions to this case (Cusumano and Selby 1997; Capiluppi and Michlmayr

2007; Garcia and Steinmueller 2003)), that we feel is very useful in extracting the principles of Inner Source.

A. ***Bell Labs*** *(Gurbani* et al*, 2005)*

This is the first case study in this paper and provides real insight into what can be achieved with Inner Source. Bell Labs provides indication that open source has overcome many GSD problems (Herbsleb and Mockus 2003) and has allowed distributed users to contribute to code with minimum conflict or overheads (Mockus, Fielding and Herbsleb 2002). Many companies are looking to capitalize on the success of open source in the hope that it will aid them overcome similar problems. The authors of the Bell Labs case study (Gurbani *et al*, 2005) focus on industry adoption of the open source development methodology. Their study contributes to the knowledge of Inner Source by "providing detailed analysis combed from interviews of multiple developers and quantitative analysis of data pertaining to a corporate open source project where multiple organizations contributed synergistically to further and use a common asset". The case study focuses on a development project in Bell Labs that uses open source software and practices to develop a commercial telecoms Internet telephony server. The software was available in Lucent Labs over a four-year period for anyone who wished to integrate it into their particular software product line. This example is a perfect fit for inner source as the reusability of the code developed is very high.

The differences between open source and commercial development as it is termed are outlined below in Table II, in terms of Requirements, Work Assignments, Software Architecture, Tool Compatibility, Software Processes and Incentive Structure. These categories of contrast emerged from the study and were observed in the implementation.

The software developed was a Session Initiation Protocol (SIP) Server (Rosenberg *et al* 2002) which is a text based protocol like FTP, HTTP or SMTP (Gurbani *et, al* 2005). The SIP stack was an emerging trend in communications technology at the time of development and is a very interesting technology both to developers and to the many areas of software development in the company. The code was written in C and was available on Concurrent Versions System (CVS). This was used for source control and version control. The code ran on Solaris and Linux, and was originally a SIP server and was later re-factored into a SIP library which hosts a SIP server.

The open source approach in this project arose in three phases; the initial development phase started in April 2000 and was developed and tested for compliance to the IETF standard by two developers. The second stage involved ad hoc distribution of the binary within Lucent Technologies. As maturity of the server grew so too did interest in the servers capabilities within a few select groups. As the maturity of the server grew further, it was no longer a research-only project, and became a full Lucent Technologies product offering. The increased interest in the SIP server binary did not translate into greater contributions beyond users reporting their experience to the developers. However, the use of the binary SIP server increased general use and the profile of the product. User requests and feedback reports were sent back to developers and prioritized based on business needs of the groups using the product, where fortunately a lot of these requests coincided. As SIP grew in profile in the industry and Internet telephony became a big part of emerging trends, the SIP server was viewed as an important resource by many groups within Lucent. By 2003 the source code was being studied extensively by many groups in Lucent.

Phase three involved more product-specific requests to the author that also coincided with actual code contributions and ideas. This was the needed breakthrough from the traditional to open source model in a corporate setting. The code then took an important change in that it was re-factored to include a transaction library, since all users of the code need a transaction manager. This enabled the creation of multiple user agents that executed on top of the transaction manager (Arlein and Gurbani 2004). Each group could now concentrate on the specific functionality they wanted to add to the user agent instead of having to also concentrate on the handling of SIP transactions. However, in this dynamic environment where open source methods were in use but without the traditional open source social structure and formatting, some issues began to arise.

Key Results: The centralized approach to development led to the creation of a high profile development project where many people experienced the process of emerging technologies development. The project was also used to groom highly motivated staff from other areas in the company who were contributing to the project in a hobbyist approach in their spare time. The project also gained from the classic open source benefit of many eyeballs making the problem look small (Raymond 2001), or even many eyeballs from different perspectives finding more bugs. In particular this company had the benefit of performance experts suggesting a list of changes to optimize the code; API experts suggested an appropriate API to make the SIP server into a more useable framework and others suggested changes to help port the code to different operating systems.

Bell Labs were very successful and could well be considered as a poster child for Inner Source. A summary of our understanding and some lessons learned from this case include:

- The project successfully implanted a number of changes in technology: a CVS, and some communication tools: for example a mailing list.
- The code quality greatly benefited from many experts having access to the code.
- The trusted lieutenant role was required and introduced.
- Use of an emerging technology that really took off when Internet telephony became more standardized and interest in the area grew. For this reason a research department or a department focused on future technologies is seen as a good place to foster growth in inner source development within a company.

- One of the code authors was a leading expert not only in the code but also in the IETF SIP standards. Experts in other software capability areas joined and made the product very strong.
- The code was heavily based on standards as a result of utilizing a resource where someone else was responsible for the code's compliance with the evolving standards. This was subsequently seen as a good way to capitalize on resources.
- The timing of the code's maturity coincided with industry interest.
- The company was liberal with key developer time and also the project seemed to be aligned with the company's business strategy.
- The company was willing to make significant changes to company structure and employee roles to leverage inner source as a useful company resource in the future.
- However, issues arose as groups forked code and subsequent code changes did not benefit the groups depending on the forked code.
- The cost of the solution was a problem in that it was seen to be under-valued as opposed to a bought solution by the accounting and costing departments.
- Communication was an issue that caused a duplication of effort by two different groups.

*B. **HP** (Dinkelacker* et al.*2002)*

Our second study examines the use of Inner Source within Hewlett Packard (Dinkelacker and Garg 2001). The authors use a number of terms almost interchangeably such as "Corporate Source", Progressive Open Source (Dinkelacker *et al.* 2001; Dinkelacker *et al.* 2002) and finally, Inner Source. All three share the same hypothesis, i.e. it is possible to migrate certain software engineering principles from the open source paradigm to the corporate software development environment. The three terms as defined in this study differ in terms of access or how open the source is. HP defines Inner Source access to be only within HP and behind the corporate firewall. Corporate Source is HP's implementation of Inner Source and is open to all HP employees. HP's Progressive Open Source (POS) has three distinct levels: Inner Source, Controlled Source and Open Source with respective users as: HP employees, HP partners and finally everyone with an internet connection.

HP understands the use of Inner Source as the application of open source concepts, perspectives and methodologies in a corporate environment. Open source is seen as being relevant to corporate environments due to its success in software maintenance, reusability and quality (Raymond 2001). Another reason given is that more eyeballs make problems look small. However fundamental conflicts exist. Businesses usually try to retain intellectual property (IP) rights on their software. IP rights are an important issue especially where the Inner Source boundary extends to include a corporate federation. The case study illustrates how Inner Source can be a good way to realize the benefits of open source while sidestepping possible risks. One risk associated with maintaining a fully open source solution includes the release of untested software. Some of the advantages observed by HP in open source projects include: community building, open discussions for requirements and features and evolvable and modular designs. However, open source and traditional corporate software development have vastly different goals, time constraints, motivations and environments. The biggest difference between the two, as seen by HP, is the pressure of time to market a product, thus making it very important to appreciate how the principles of open source of early release are transferred to the corporate environment. In an effort to make this transfer possible and attractive HP set out a list of expected benefits to be gained through an evaluation of required factors including:

- A toolkit of existing software from which to build new products
- Improved quality of code as many authors will have contributed and reviewed the code
- Community Debugging
- Leveraging of open source tools and methods
- Aid developer's movements within the company as existing developers are aware of the common code tree.
- Faster development schedules with the use of code common to multiple projects.

The challenges identified by Dinkelacker and Garg (2001) included both organizational and technical issues. In relation to organizational challenges, virtual organizations or group dynamics within a company may work together with various degrees of success, e.g. if one group always writes high quality code while another group capitalizes on this and uses resources elsewhere. Leadership is also required in all open source-like projects. However, two distinct problems arise from this in a corporate environment. Firstly, what happens when a code/module leader leaves the company? Secondly, what if no leader is responsible for a section of the code? Furthermore, task assignments are much easier to manage in smaller groups where a project manager is able to judge the skill set of each member in the group. It is much more difficult to do this on a large scale. Developer indoctrination is also possibly the most important but overlooked organizational challenge. If people do not accept the inner source model and really work with it (this includes code standards and general rules of operation), it will be very difficult to manage and run.

Concerning technology requirements, these include a repository for the code to be held with version support. Community support is also important and tools are required to enable community participation and communication. Security is a general concern of every company's code, but even more so in this case because as the access to code increases, so too might the risk in terms of IP rights and also by allowing remote repository access. Search and navigation are also other technical requirements to be considered, as are IS and IT support.

Key Results: It was envisaged that the results of implementing the above factors would include greater agility to address new opportunities, reduced time to market, higher quality code, increased consistency between HP products and decreased costs of product production.

The use of Inner Source in HP is continuously being improved with the focus for improvement being on organization and cultural issues. From a technical viewpoint the fundamental shift in focus is that instead of the binary code being the end-product, the source code is now the end-product. Adoption of inner source is more a social change than a technical change. While technical changes are needed, more important aspects include leadership and developer backing. There also exists a trade-off between choosing an area that is a concern or interest to many people and choosing an area that is of high value to a lot of people. Code module granularity is important in a corporate source code repository. The module must offer significant functionality to be worthy of potential inclusion and must not be too large and complicated with overheads and calling sequences. The latter would defeat the purpose of agility and reuse making it easier to write the required functionality from scratch. Corporate source has value beyond software product development, e.g. computing infrastructure and internal decision support tools. Some HP projects already make available their code on their own web service and in this case do not see the need to move over to the corporate source system.

*C. **IBM** (IBM Community Source, 2007)*

IBM, although a supporter of open source projects[2], has long been a traditional software development company, choosing to develop its software in a proprietary manner. It has now become a leader in adoption of OSS and the OSS model in the corporation (Capek *et al*, 2005). It is this fostering of open source outside the company that has afforded IBM the opportunity to weigh some of the attractive attributes of open source projects and bring them into the corporate environment (Capek *et al* 2005).

Community Source is the term preferred by IBM rather than Inner Source. From the initial data we know that Community Source methods fit into Inner Source but not all Inner Source principles seem to be present in the Community Source solution.

It is clear that IBM has invested significantly in Inner Source (IBM Community Source, 2007). IBM sees Inner Source as a good way to liberate the creativity of programmers, drive efficiency and bring products to the marketplace faster.

Inner Source is seen as a new development methodology for the globally distributed company. A strategic decision to decompose technologies into a number of components to aid reuse has been a driving factor in change. The use of Inner Source is seen as a good way to aid cross pollination of expertise, ideas and requirements between all the locations of IBM labs. IBM has essentially borrowed from open source philosophies, strategies, tools and culture to transform IBM's development practices to further support global component development. IBM sees Inner source as a means to an end. In the not too distant future every software company will need to componentize their code and Inner Source is a way to do this while giving much freedom to the developer and promoting innovation across the company.

IBM has over a hundred projects that are currently using the Community Source infrastructure and has over 2,000 users that are currently registered and using the Community Source tools and processes (Taft 2005). The company has tools that store the actual source logic using repositories very much like SourceForge, documentation and specs, news and bulletins, patches and fixes, and educational material and broader community tools and support. The company states that development is 30 percent faster using an open source model (Capek *et. al.* 2005). IBM sees Inner Source as a good way to liberate the creativity of programmers, drive efficiency and bring products to the marketplace faster.

As the IBM program involving Inner Source continues, the company is really trying to harvest the programming culture and practices that have been the product of decades of top class development at IBM. Along with these practices IBM are attempting to blend some of these new techniques that have become apparent through exposure to open source communities. Older projects will gradually adopt the new development methods and new projects will benefit from traditional IBM disciplines. This approach is seen as the best way to blend the two worlds but also poses the biggest challenge going forward.

IBM has a clear business vision that focuses on the components of their software and the gradual introduction of open source methods to the company wide development environment. IBM seems to concentrate on the software development methodology instead of the organizational structure or the people behind the software methods. However little is known about the extent of the programme and the detailed organizational changes needed to implement the system. The information we do know about IBM focuses on the technical changes to code architecture which forms only a part of Inner Source principles.

IV. DISCUSSION

In the previous section we examined three cases studies, Bell Labs, Hewlett Packard and IBM. Each case was discussed in terms of its background, how Inner Source was applied within each company, the implications and problems that arose as a rest of the implementation, the key results drawn from the case study and an analysis of each case study. It is common as can be seen from the cases above, that the introduction of Inner Source was typically a project that was not a product in itself but was a product to be offered to all project teams for use within their projects. The project chosen was typically an emerging area of interest in the

[2] IBM has played a large part in the success of open source projects such as the Linux operating system, Apache web server and the Eclipse software development environment. Each of these projects can be considered high profile, successful, OSS initiatives.

Identify applicable sponsor/s here. *(sponsors)*

industry with a dedicated domain expert as the champion of the project. As the project grew in maturity it would attract more attention and benefit from the snowball effect where attention would gain more developer participation and achieve a yet more mature stage.

The cases examined helped establish ongoing implementations of Inner Source and the issues surrounding such implementations. We now understand the:

- Differences between Inner Source and traditional software development.
- Differences between Inner Source and open source.
- Perceived benefits of Inner Source usage.
- Actual benefits of Inner Source usage.
- Problems with implementing inner source solutions and some of the possible factors causing these problems.
- Principles on Inner Source.
- The cultural changes required for a successful Inner Source solution.

This leads us to make some arguments concerning firstly, how inner source is managed and organized (Section V) and secondly, a step closer to characterizing Inner Source and formulating the main principles of Inner Source (Section VI).

V. Managing and Organising Inner Source

In this section we look at how traditional organizational and management structures have changed in order to accommodate and benefit from Inner Source adoption. The case studies indicate that there are two clear advantages of adopting Inner Source use within companies stemming from the increased visibility offered by open source principles, but also a number of problems but they are all related to the idea of visibility afforded by the open source aspect of Inner Source.

Better quality code through visibility: Visibility is an important issue to consider in Inner Source. As traditional projects move from away from traditional style development, where the code and communication is usually shared only within the project group and where face to face communication is common, to Inner Source style development visibility of the developer's work and textual communication increases significantly. The issues surrounding this may include an increase in quality of work as a result of many people being able to give helpful feedback but it also takes more time and effort for people to communicate in this environment.

Pride in their work through greater visibility: In HP, through increased visibility, the employees were conscious of posting to the Inner Source community as their messages were going out to a wider community. Managers felt that the use of the Inner Source solution would lead to an increase in quality due to the open nature of the Inner Source solution. People were insecure about posting their work for a company wide audience but eventually felt a lot of pride associated with contributing quality work (Melian, 2007).

Security issues surrounding visibility: However, as visibility increases so too must the companies' security efforts, especially in the case where various contractors are employed within different areas of the company. Where in the traditional development environment the access of people to code is localized within company departments and easier to manage, in contrast the very openness sought after in the use of inner source makes security and access control a lot more difficult. In HP digital badges were used to control access and vendor access to digital badges was something new that had to be managed as a result of the use of Inner Source. This creates an additional issue to be dealt with when bringing in a new vendor, as not only must the new vendor have a physical ID but also a digital badge for Inner Source access (Melian, 2007). This creates a certain level of trust within the company that the code is safe but when using external vendors the perception is that they have no control over what security the vendor is using once the vendor is given access.

Fear of job loss through visibility: Some of the fears encountered as a result in the increase in openness include fear of granting access to external access of the code to vendors and also by allowing other groups to see what we do, we run the risk of someone seeing an opportunity to downsize our department and take over our work. The other fear among developers is that by everyone in the company potentially having greater access to the inner workings of the company any employee that leaves permanently may release trade secrets to competitors and as a result threaten jobs in the company

Privacy and knowledge retention issues related to visibility: There is also the argument that in an environment where there exists a centralized repository where a lot of information is freely available on a company wide basis and if free to traverse and browse, that this environment could make people reluctant to use digital media for some forms of communication. This was certainly the case in HP where developers rejected the suggestion of the use of video equipment to record idea generation and brainstorming in a meeting direct because of the likelihood that the media would be made freely available in a company wide system where the media could live for a significant period of time.

Increased visibility leads to easier workplace monitoring: Introduction of an Inner Source system that is perceived to be open and a controlling factor by management causes serious undermining of developers. If Inner Source is to be used effectively and to promote innovation then introduction of an Inner Source system can aid decentralized and geographically dispersed development teams to work together on a project. The openness of the Inner Source solution deployed in HP caused developers to be a lot more careful about the quality of their work and comments they were contributing. The openness of the system also caused some users to become very concerned over job security. It is clear that management and organizational structure must

change and adapt to the use of Inner Source as must the users of the system.

Table 1 outlines the background difference between traditional software and Inner Source software development and the corresponding problems arising as a result of a change in environment. While some of these problems are a direct result of implementing inner source not all problems have a net negative effect. For example the problem of code forking, depending on the situation, is probably better than the previous situation where code was developed in parallel with little information about other ongoing project development. At least in the case of code forking the developers are aware of other code existing and possible solutions to existing bugs.

In this table we compare and contrast traditional software development to Inner Source software development based on the information gathered from our findings in the case studies examined in Section III and the expectations and perceptions of the people aiming to capitalize on open source. We use this contrast of traditional software development and Inner Source software development not to suggest that the two terms are polar opposites or by implication, that traditional software development and open source software development are polar opposites, but we instead use this comparison in order to make apparent the useful areas that aid in the identification of activities that typically occur in Inner Source. There are of course cases of cross-over between both types of development for example Garcia (2007) argues that most of the characteristics found in traditional development are found in open source development, and in particular Capiluppi and Michlmayr (2007) argue that many open source projects never migrate from centralized development to distributed development which are commonly held polar opposites between traditional development and open source development (Raymond 2001).

Another example of this cross-over is when companies attempt to build a community around a corporate product (West and O'Mahony 2004) where companies which released developed code to a public community much in the same way Nokia release code to the Maemo community. Maemo is Nokia's Linux based platform for handheld devices, where code is released to the community but no changes or applications developed by the community are accepted back into the Nokia product offering.

TABLE I. INITIAL FRAMEWORK FOR INNER SOURCE PRINCIPLES

	Traditional Development	Inner Source Development	Inner Source Problems and Issues as a result of change.
Process	Localized approach to development	Centralized approach to development.	Forking of code.
	Procedures and knowledge of how to account for software based on set formula assuming certain structures are present.	Absence of procedure of how to attribute value and cost when resources and input is company wide.	Inner source solutions not being valued or accounted for correctly.
	Local tools and standards based on each project preferences and closest development partners.	Company wide standards, tools and practices.	Pushing of company wide code practice, standards and tools seen as excessive and unnecessary.
	Breadth of ideas and thinking likely to be a lot smaller.	Breadth of ideas greatly increased.	Possible strain on resources as more interesting, less company centric ideas get developed.
	Defined timescale of development and testing, with complete control of the code being held by the local project group.	Constant development of code with no group having the ability to stop development for a testing cycle.	Code freezing not possible before project release.
	Communication and code access on a local level.	Open CVS, mailing lists available company wide.	Privacy of ideas and conversations an issue.
	Tight time constraints	No time constraints	Possible delay in time to market.
	Much face to face communication	More towards distance interaction	Developers spend more time on communication as it is perceived that they are communicating to the entire company and the discourse can be archived.
	Heavy hierarchical business organization structures	Structured to more loose roles	Concerns over project groups taking over the responsibilities of other groups.
	Tools and methods vary from project to project.	Corporate wide policy of tools and methods	Resistance to change is a real threat as tools and standards directly affect developers' day to day efficiency.
	Costing of a project is easily tracked with defined resources and costs being easily tracked with each project group.	Costing of inner source becomes much more difficult as code contributors can potentially come from any department or project	Implementing a new development model requires education of those responsible for costing as both savings and costs occur in new places and stages of development. In particular there is a cost incurred in any change applied in a company but specifically if a company wide adoption of tools and standards is required

Product	Code quality is dependent on a few key people within a project	Code quality is a concern of everyone across multiple projects, and is directly linked to the acceptance of the submitted code in the corporate repository.	The code quality required for Inner Source acceptance may be higher than is actually required and so this could have a negative effect on product development time.
	Code exists in an isolated manner with the possibility of major company resources being used to solve a problem multiple times.	Resources are less likely to be wasted on solving a problem many times as anyone can check if someone is attempting a solution with the correct communication structures in place.	There is increased dependency on technology as a medium of interaction. Allowing corporate wide access to Inner Source code means that employees who leave the company leave with more insider information than before.
	Focus on the binary shipping and release	Focus moves to producing source code for CVS submission.	Possible, delay in time to market.
	IP is a closely guarded secret	IP is released company wide	Every employee knowing the business of the entire company a threat to jobs when that employee leaves.
	Code held in a local context	CVS and associated tool usage	Issues related to tool-use like dependence, and even political debates on which tool to adopt.
	Code review takes place at key points	Constant code review	Possible duplication of effort and inefficient use of time.

VI. CHARACTERIZATION AND PRINCIPLES OF INNER SOURCE

In this section we characterize Inner Source (see Table II). Building on this characterization we then present the principles of Inner Source using Raymond's (1999) nineteen principles as a point of comparison and contrast (see Table III). This is not to suggest that we believe that Raymond's (1999) work is definitive in this area. We merely look back on his principles to indicate what is also true of Inner Source and to make apparent some areas of future research in Phase II of OPAALS.

A. *Characterizing Inner Source*

Our analysis of the case studies allowed us to organize the main arguments and issues emerging over the use of Inner Source ideas in companies. We identified these to fall within two broad categories of *product* and *process* (see Table II). Open source can be usefully analyzed through ideas of product and process (Fitzgerald 2006) and thus this categorization is also, to a degree, useful for Inner Source studies.

Product – We observed three aspects of a product that were questioned, domain area, knowledge issues and economic issues. Of these three the former two were for the most part considered unproblematic but economic issues such as the calculation of total cost of production and ownership is an area that needs further research as organizations are struggling with a way to come to an accurate estimate. Champions of Inner Source in organizations find it difficult to convince top management about the benefits of Inner Source development if they cannot justify to at least some degree the cost factor involved.

Knowledge diffusion and innovation in the product are a true selling argument in favour of Inner Source use. The case studies indicate quite clearly that all the companies that chose to take the Inner Source route did so to enlarge he developer and knowledge base. Domain experts, through the inclusive nature of Inner Source, were able to contribute and guide project development which lead to better quality code, and applications that were closer to user requirements.

Process – Communication, knowledge, community and structure were observed to be important factors in Inner Source use. Again, we have some factors that are treated as if they are unproblematic like communication. This is how communication has been presented in the literature on Inner Source but we have reason to believe that this aspect is closely related to governance and control issues and thus not as uncontroversial as appears.

The question of knowledge is raised again under process but the focus this time, as opposed to product knowledge where quality of code was the main concern, is on idea generation and linking this to idea dissemination and innovation. The process understanding of knowledge is much broader than the aspect looked under product. Here we show that code, knowledge and community are related and indeed inextricably linked.

The issue of community is an important one in Inner Source and was mentioned repeatedly in the case studies examined, however, how do we define "community"? Should the "community" have certain traits in order to be able to support Inner Source? Hillery (1955) found 94 different definitions of community with only one common element, that of people. Changes in what a community is today only serves to complicate any definition of community. The issue of community in Inner Source needs further examination and is an interesting area for future research. Questions such as how is Inner Source accepted in an organization, by whom, what is the governance structure of the organization and how does it adapt to Inner Source, and how are smaller sub-communities of developers assimilated if they are resistant to Inner Source are all pertinent and interesting avenues for future research.

TABLE II. CHARACTERIZING INNER SOURCE

Inner Source Characteristics		
Product	*Domain area*	A common area to choose for Inner Source projects was that of research or emerging technologies.
	Economic issues	Costing of code produced in the new method was not valued correctly.
		Structural changes often would lead to reduced costs in producing a product using inner source.
	Knowledge issues	The quality of the code increased as a result of the inner source implementation.
		The use of domain experts from across the company can increase the product's quality significantly.
Process	*Communication*	CVS and associated repository tools, along with community communication tools (mailing lists, web forums, etc) were part of a common solution.
		Communication issues were common.
		Success factors that play a role in open source include simple communication mechanisms with a low learning curve.
	Knowledge issues	Computer mediated groups generate more ideas (Nunamaker *et al*, 1991).
		The breadth of ideas is a lot greater in asynchronous communications media such as is found in open source projects (Benbunan-Fich *et al*, 1999).
		Formalization inhibits innovation (Klincewicz, 2005).
		Open source development is co-evolution of code, knowledge and community.
	Community	There needs to exist a large degree of overlap between community values and governance structure.
		Motivation of company and contributors needs to be clear but crystal clear for the company.
		Open source can be described as constellations of communities who can work on what they choose rather than a formal set of tasks mandated by business objectives.
		As with implementing any change in a company and particularly open source changes to a company there is evidence to suggest that employees will resist change (Jaaksi, 2007).
	Structure	Forked code was a problem that effected support of the common infrastructure.
		Significant organizational structural changes were required.
		Structural changes often lead to reduced time to market for products using inner source.
		Boundaries are especially important, where developers can work and where contribution is not really requested. Case example Maemo and Nokia has a one way code release (any open source developer contribution will not feed back into the Nokia code for a number of reasons).
		Open source can often be seen as an experimental approach to software development in that coordination takes place after action.

From our initial examination of the area and this issue in particular we have observed the community to include all developers and staff with at least code interaction duties and possibly all employees. The issue of suitable communities of Inner Source did not arise as the case studies had an inclusive approach where everyone in the company had access, and an experimental approach where they envisaged a better product would be where everyone had input. The point was to implement the project and then evaluate the experience.

The most problematic, though interesting area under process that we found in our study is structure. This aspect is linked to all others under process but very closely to the community one. Each issue that we found related to structure has a positive and negative angle which implies how finely tuned the governance strategy of an Inner Source policy and company will need to be. Boundaries of the project and the organization are relevant and yet often blurred because it is difficult to enclose a community that feeds (and provides food for thought) on ideas. Employees leave an organization but take a 'part' of the knowledge with them, and new people join that then need to be *included.* Structure and organizational change will form a key part of our study in Phase II.

B. *Principles of Inner Source and Future Research*

How do we determine if a software product, piece of code or any other entity is entitled to be labeled open source? There is more than one meaning of open source (Gacek, 2004) and there are of course different practices that are seen in many open source projects (Halloran *et al* 2002). One definite metric is if the code has a license recognized by the Open Source Initiative (OSI). Licenses in general are rather tricky to understand because some appear to be open source but contain clauses that negate the principles of OSI acceptable licenses. The case of Sugar CRM is one such example. It claims to be an open source product as its code is available but the license includes the stipulation that the Sugar logo must be used on every single page of any derivative work. Many argue that this clause invalidates the 'open sourceness' of Sugar CRM, indeed the Sugar CRM license is not recognized by OSI. Interestingly, recent changes have been made by both Sugar CRM and OSI that have seen the Sugar CRM product fully recognized by the OSI.

However, the emerging area of inner source does not have an organization such as the OSI to look to for an approved license under which they can release products and so we have chosen to look more clearly at the principle of open source and the contrast and similarities that exist with Inner Source.

Since Raymond (1999) wrote *The Cathedral and the Bazaar* a number of critiques have emerged which rightly make apparent the naïve aspects of Raymond's work (Bezroukov 1999a, Bezroukov 1999b) yet his principles provide us with a good starting point to understand the basic differences between open source and Inner Source. Table III shows Raymond's claim and then what we propose for Inner Source. There are many similarities but also some clear distinctions, e.g. how in Inner Source the code is kept only within the organization and not released into a larger open

source community. In this table we also indicate the areas of future research we intend to engage in and we encourage other researchers to take up some of these challenges.

Some observations on the principles of Inner Source have become apparent upon review of the areas and are summarized in Table III below.

TABLE III. OPEN SOURCE AND INNER SOURCE PRINCIPLES: *POSSIBLE FUTURE RESEARCH*

	Open Source Principles (Eric Raymond, 1999)	Inner Source Principles
1.	Every good work of software starts by scratching a developer's personal itch.	Partially true but this is also complemented by software demands and needs emerging from within the organization and is tested rigorously. *An area for further research here includes the process of requirements gathering for Inner Source projects which may be part of Phase II.*
2.	Good programmers know what to write. Great know what to rewrite (and reuse): Promotes parallelism and redundancy. Latter leads to knowledge sharing and transfer	This principle holds true for Inner Source, code reuse in multiple projects is a key benefit to any organization attempting to employ Inner Source. *How much code is reused in which projects and when, and why does duplication of effort occur in organizations is a possible avenue of future research.*
3.	``Plan to throw one away; you will, anyhow." (Fred Brooks, *The Mythical Man-Month*, Chapter 11).	*Time allocation in Inner source projects is an important issue as different groups have different timelines for products depend on the same code this can have the potential to cause issues within projects. Future work may include exploring the factors that possible misalignment of time planning might cause.*
4.	If you have the right attitude, interesting problems will find you.	Developers within an organization often have deadlines and so cannot work on intriguing problems as much as they would wish – however this point is somewhat true in inner source in that, if you wish you can find an interesting problem to work on. Exploring this in Phase II might lead us to examine how the interesting problems get solved and if it is based on merit or position.
5.	When you lose interest in a program, your last duty to it is to hand it off to a competent successor.	Finding a successor to champion your project when you leave is not easy for inner source so is true for both open and inner source, however company structure should mean that someone is responsible for the management of the project. *This is an interesting area and needs further research in Phase II.*
6.	Treating your users as co-developers is your least-hassle route to rapid code improvement and effective debugging.	This is the basis of Inner Source too, in that the company's developers can all be users and developers of any company built code. This is exactly the principle that companies are trying to engender within their organization. *The level of success they have achieved varies and matching their expectations to the actual result is an area that can be explored further.*
7.	Release early. Release often: reward feedback. Prompt feedback leads to tight development, debugging and testing (Maintenance problem is that you don't even recognize your own code 6 months later).	This principle is as true in Open Source as it is in Inner Source. *The reward system in Inner Source is an interesting area of future work. The contrast between recognition as a reward in open source versus monetary reward in companies is a major difference.*
8.	Given a large enough beta-tester and co-developer base, almost every problem will be characterized quickly and the fix obvious to someone. Given enough eyeballs, every bug is shallow: community feedback, author and community review again – double-loop learning. Self organization of community and knowledge.	This is a major benefit of Inner Source and the aspect of leveraging the community feedback extends in Inner Source to include bugs, community feedback and user requirements.
9.	Smart data structures and dumb code works a lot better than the other way around.	Code must be structured and abstract to ensure ease of use and readability. *This principle has yet to be examined in detail in the case studies but is one that is examinable in Phase II.*
10	If you treat your beta-testers as if they're your most valuable resource, they will respond by becoming your most valuable resource.	*This needs further research as it is not clear what level of feedback is given by other employees of the organization in Inner Source projects.*
11	The next best thing to having good ideas is recognizing good ideas from your users. Sometimes the latter is better.	*This needs further research as which ideas come to exist and are implemented is not something explored in the case studies.*
12	Often, the most striking and innovative solutions come from realizing that your concept of the problem was wrong.	A rather basic argument in open source which is probably true for Inner Source and software development in general (though organizations with stern deadlines may not allow employees such flexibility).
13	``Perfection (in design) is achieved not when there is nothing more to add, but rather when there is nothing more to take away."	True for Inner Source as well as all 'good' code development.
14	Any tool should be useful in the expected way, but a truly great tool lends itself to uses you never expected.	*Needs further research in order to make a comparison.*
15	When writing gateway software of any kind, take pains to disturb the data stream as little as possible—and *never* throw away information unless the recipient forces you to!	Organizations tend to archive code, communication and other related material so Inner Source should be able to avoid such pitfalls; *however there is no research to date that is able to say this is true for Inner Source.*
16	When your language is nowhere near Turing-complete, syntactic sugar can be your friend.	No evidence to prove or disprove this principle in current literature but maybe worth examining in Phase II.
17	A security system is only as secure as its secret. Beware of pseudo-secrets.	Visibility of code and work practices is clearly both an advantage and a disadvantage in Inner Source (see Section V).
18	To solve an interesting problem, start by finding a problem that is interesting to	This principle does not hold true from the developer's standpoint, as developers cannot in all situations

	you: knowledge generating. Intrinsic self-selection.	choose the most interesting problem to work on, *however there are exceptions to the rule which we will see later.*
19	Provided the development coordinator has a communications medium at least as good as the Internet, and knows how to lead without coercion, many heads are inevitably better than one. (Two principles – Internet and low coercion (Fitzgerald, 2006).	(A) The coordinator must be strongly connected to the group, and Inner source communication tools provided to aid this. (B) Low coercion is also a principle of Inner Source but in a very different way, in that high coercion is inherent in a corporate environment but the Inner Source environment should adopt a low coercion ideology. *Further research in this area will be planned for Phase II.*

We have begun the process of understanding Inner Source and refining the principles that are indicative of its practices. Organizing our ideas has made apparent to us the more relevant aspects of Inner Source which need to be studied in Phase II of OPAALS and beyond for researchers.

VII. Conclusion

In conclusion we have examined a number of case studies involving various Inner Source solutions. We looked at the expected results of the Inner Source solution and the motivations behind the deployment. We then looked at the actual benefits and the problems associated with the deployment. The issues of managing and organizing inner source were examined. These issues are especially important to Inner Source and as we have seen some critical changes were required in the successful implementation of Inner Source. In particular we have seen that to create innovation a flexible approach to managing is required and the organizational structure must undergo some key changes to foster growth of Inner Source. One particular finding is that organizational structure is the biggest issue that must be constantly examined when implementing an Inner Source solution.

Our paper also clarified what we understand to be Inner Source principles. We looked at emerging principles in terms of how they relate to open source and also how they relate to traditional software development. These principles help to characterize the Inner Source phenomenon more clearly; set out the scope of our future research; and also help us to explore the potential cross over of Inner Source principles beyond software. We have indicated future research opportunities in Inner Source clearly in Table III. The change occurring in the corporate world in order to leverage the successes of open source is a phenomenon not limited to software and this is another aspect we and other researchers are currently exploring.

Acknowledgment

This work has been funded by the OPAALS research project and the European Union Sixth framework.

References

Ågerfalk P J (2004). Investigating Actability Dimensions: A Language/Action Perspective on Criteria for Information Systems Evaluation, Interacting with Computers, 16(5), pp. 957–988.

Ågerfalk, P.J. (2005). Studying Persistent Conversations in an Open Source Context: A Conceptual Framework, In Proceedings of Promote IT 2005, (Eds, Bubenko jr., J et al.) Studentlitteratur, Lund.

Arlein, R. and Gurbani, V., An Extensible Framework for Constructing Session Initiation Protocol (SIP) User Agents. Bell Labs Technical Journal, 9, 3 (November 2004), p. 87-100.

Benbasat, I, Goldstein, D.K., and Mead, M. (1987) "The Case Research Strategy in Studies of Information Systems", MIS Quarterly, 11:3, pp. 369-385.

Bonoma, T.V. (1985) "Case Research in Marketing: Opportunities, Problems and a Process" Journal of Marketing Research (22), pp. 199-208

Benbunan-Fich, R., and Roxanne Hiltz, S. "Impacts of Asynchronous Learning Networks on Individual and Group Problem Solving: A Field Experiment," Group Decision and Negotiation (8) 1999, pp 409-426.

Bergquist, M. and Ljungberg, J. "The Power of Gifts: Organising Social Relationships in Open Source Communities" Information Systems Journal (11:4) December, 2004 pp.305-320.

Bezroukov, N. (1999a) "Open Source Software Development as a Special Type of Academic Research (Critique of Vulgar Raymondism)", *FirstMonday: Peer Reviewed Journal on the Internet,* **4 (10),** pp. 1-23.

Bezroukov, N. (1999b) "A Second Look at the Cathedral and the Bazaar", *FirstMonday: Peer Reviewed Journal on the Internet,* **4 (12),** pp. 1-29.

Capek, P.G., Frank, S.P., Gerdt, S. and Shields, D. (2005) A history of IBM's open-source involvement and strategy, IBM Systems Journal, 44, 2, 249-257.

Capiluppi, A. and M. Michlmayr (2007) 'From the Cathedral to the Bazaar: An Empirical Study of the Lifecycle of Volunteer Community Projects' in IFIP International Federation for Information Processing, Volume 234, Open Source Development. Adoption and Innovation, eds. J. Feller, Fitzgerald. B., Scacchi, W., Sillitti, A. Boston: Springer, pp. 31-44.

Carmel, E. and Agarwal, R. (2001). Tactical approaches for alleviating distance in global software development. IEEE Software, 18 (2), 22-29.

Carroll, J. M. (1996). "Becoming Social: expanding scenario-based approaches in HCI." Behaviour & Information Technology 15(4): 266–275.

Clark, H. H. (1996). Using Language. Cambridge, Cambridge University Press.

Conway, M.E., How Do Committees Invent? Datamation, 14, 4 (1968), p. 28-31.

Crowston, Kevin, Scozzi, Barbara, Open source software projects as virtual organizations: Competency rallying for software development. IEE Proceedings Software, 149(1), 3-17. (2002).

Cubranic, D. (1999). Open-Source Software Development. In 2nd Workshop on Software Engineering over the Internet, 7p.

Cusumano, M. and R. Selby (1997) "How Microsoft Builds Software", Communications of the ACM, 40 (6), pp. 53-61.

Gallivan, M.J. "Striking a Balance between Trust and Control in a Virtual Organization: A Content Analysis of Open Source Software Case Studies," Information Systems Journal (11:4) 2001, pp 277-304.

Gurbani, V., Garvert, A., and Herbsleb, J., A Case Study of Open Source Tools and Practices in a Commercial Setting, Proceedings of the 5th ACM Workshop on Open Source Software Engineering (WOSSE), (May 2005), pp. 24-29.

Dinh-Trong, T. and Bieman, J.M. (2004) Open Source Software Development: A Case Study of FreeBSD. In Proceedings of the 10th International Symposium on Software Metrics, IEEE Computer Society.

Dinkelacker, J., Garg, Corporate Source: Applying Open Source Concepts to a Corporate Environment. HP Technical Report 2001. http://www.hpl.hp.com/techreports/2001/HPL-2001135.html?jumpid=reg_R1002_USEN

Dinkelacker, J., Garg, P., Miller, R., and Nelson, D., Progressive Open Source. HP Technical Report 2001. http://www.hpl.hp.com/techreports/2001/HPL-2001-233.html.

Dinkelacker, J., Garg, P., Miller, R., and Nelson, D., Progressive Open Source. In Proceedings of the 2002 ACM International Conference on Software Engineering (ICSE'02), pp. 177-184, May 2002.

Erenkrantz, J.R. and Taylor, R.N. (2003). Supporting Distributed and Decentralized Projects: Drawing Lessons from the Open Source Community. In The 1st Workshop on Open Source in an Industrial Context, p. 21-30, <wwwbib.informatik.tu/muenchen.de/infberichte/2003/TUM-I0319.pdf>

Fielding, R.T and Kaiser, G. (1997). The Apache HTTP server project. IEEE Internet Computing, 1 (4), 88-90.

FLOSS Report EU http://ec.europa.eu/enterprise/ict/policy/doc/2006-11-20-flossimpact.pdf

Gacek, C. and Arief, B. (2004). The Many Meanings of Open Source. IEEE Software, 21 (1), 34-40.

Gallivan, M.J. “Striking a Balance between Trust and Control in a Virtual Organisation: A content Analysis of Open Source Software Case Studies” Information Systems Journal (11:4) December, 2004 pp.277-304.

Garcia, M.J. and E.W Steinmueller (2003) ‘The Open Source Way of Working: a New Paradigm for the division of labour in Software development?’ INK Open source working paper No.92, SPRU Science and Technology Policy Research, University of Sussex, available at: http://www.sussex.ac.uk/spru/1-6-1-2-1.html, last accessed 15/08/07.

German, D.M. (2003). GNOME, a case of open source global software development. In International Workshop on Global Software Development, p. 39-43, <gsd2003.cs.uvic.ca/gsd2003proceedings.pdf>

Goffman E (1959) “The Presentation of Self in Everyday Life,” Doubleday: Garden City, New York.

Goldkuhl, G. and P. J. Ågerfalk (2002). Actability: a way to understand information systems pragmatics. In Coordination and Communication Using Signs: Studies in Organisational Semiotics 2. K. Liu, R. J. Clarke, P. B. Andersen and R. K. Stamper. Boston, Kluwer Academic Publishers: 85–113.

Grand, S., G. von Krogh, D. Leonard, and W. Swap (2004). Resource allocation beyond firm boundaries: A multilevel model for open source innovation. Long Range Planning 37, 591–610.

Gurbani, V.K., Garvert, A. and Herbsleb, J.D. (2005). A Case Study of Open Source Tools and Practices in a Commercial Setting. In Proceedings of the 5th Workshop on Open Source Software Engineering, p. 24-29, ACM.

Halloran, T.J. and Scherlis, W.L. High Quality and Open Source Practices. in Meeting Challenges and Surviving Success: 2nd Workshop on Open Source Software Engineering. 2002. Orlando, FL.

Herbsleb, J. (2007) "Global Software Engineering: The Future of Socio-Technical Coordination". in *Future of Software Engineering FOSE '07,* pp. 188-198, IEEE Computer Society.

Herbsleb, J.D. and Grinter, R.E., Architectures, Coordination, and Distance: Conway's Law and Beyond. IEEE Software, Sept./Oct., (1999), p. 63-70.

Herbsleb, J.D. and Mockus, A., An Empirical Study of Speed and Communication in Globally-Distributed Software Development. IEEE Transactions on Software Engineering, 29, 3 (2003), p. 1-14.

Hillery, G. A. (1955) "Definitions of Community: Areas of Agreement", *Rural Sociology,* **20** pp. 111-123.

IBM Community Source Interview, http://www.betanews.com/article/IBM_Turns_to_Open_Source_Development/1118688437 2007.

Jaaksi, Ari (2007) ‘Experiences on Product Development on Open Source Software’ in IFIP International Federation for Information Processing, Volume 234, Open Source Development. Adoption and Innovation, eds. J. Feller, Fitzgerald. B., Scacchi, W., Sillitti, A. Boston: Springer, pp. 85-96.

Kirsch, L.J “Deploying Common Systems Globally: The Dynamics of Control” Information Systems Research (15:4), December 2004, pp. 375-395.

Klincewicz K. (2005), Innovativeness of Open Source software projects. MIT Working Papers.

Krishnamurthy, S. (2002). Cave or Community? An Empirical Examination of 100 Mature Open Source Projects. First Monday, 7 (6), <www.firstmonday.org/issues/issue7_6/krishnamurthy/>

Lee, A.S. "Electronic Mail as a Medium for Rich Communication: An Empirical Investigation Using Hermeneutic Interpretation," MIS Quarterly (18:2), June 1994, pp. 143-157.

MacCormack, A., Rusnak, J., and Baldwin, C., Exploring the Structure of Complex Software Designs: An Empirical Study of Open Source and Proprietary Code, in Harvard Business School Working Paper. 2004: Boston, MA 02163.

Melian, C. Progressive Open Source, PhD Thesis, Stockholm School of Economics 2007.

Michlmayr, M. Quality Improvement in Volunteer Free and Open Source Software Projects, PhD Thesis, University of Cambridge 2007.

Mockus, A. and Herbsleb, J.D. (2002). Why Not Improve Coordination in Distributed Software Development by Stealing Good Ideas from Open Source?. In Meeting Challenges and Surviving Success: The 2nd Workshop on Open Source Software Engineering, p. 19-25, <opensource.ucc.ie/icse2002/MockusGerbsleb.pdf>

Mockus, A., Fielding, R., and Herbsleb, J.D., Two Case Studies of Open Source Software Development: Apache and Mozilla. ACM Transactions on Software Engineering and Methodology, 11, 3 (2002), p. 309-346.

Moon, J.Y. and Sproull, L. (2000). Essence of Distributed Work: The Case of the Linux Kernel. First Monday, 5 (11), <www.firstmonday.org/issues/issue5_11/moon/>

Nunamaker, J. F. Dennis, Alan R. Valacich, Joseph S. Vogel, Douglas. George, Joey F. Electronic meeting systems, Communications of the ACM, v.34 n.7, p.40-61, July 1991.

Open Source Business, The Economist, March 16th 2006 http://www.economist.com/business/displayStory.cfm?story_id=5624944

Raymond, E. (1999) *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary,* O'Reilly & Associates, Sebastopol, California.

Rosenberg, J., et al., SIP: Session Initiation Protocol, IETF RFC 3261, July 2002, http://www.ietf.org/rfc/rfc3261.txt

Sagers, G "The Influence of Network Governance Factors on Success in Open Source Software Development Projects. Proceedings of the Twenty-Fifth International Conference on Information Systems. Agarwal, R and Kirsch, L. (eds.) Washington DC, December 2004, pp.427-438.

Sarma, A. (2005). A Survey of Collaborative Tools in Software Development, ISR Technical Report # UCI-ISR-05-3, Institute for Software Research, University of California, Irvine <www.isr.uci-edu/tech-report.html>

Scacchi, W., Understanding the requirements for developing open source software systems. IEE Proceedings on Software,149, 1 (February 2002), p. 24-39.

Shah, S. Understanding the Nature of Participation and Coordination in Open and Gated Source Software Development Communities. In Annual Meeting of the Academy of Management. 2004.

Walsham, G. Interpreting Information Systems in Organizations, Wiley, Chichester, 1993.

Walsham, G. "Interpretive case studies in IS research: nature and method," European Journal of Information Systems (4), 1995, pp. 74-81.

West, J. and C.S. O'Mahony (2004). 'Contrasting Community Building in Sponsored and Community Founded Open Source Projects', available at: http://opensource.mit.edu/papers/westomahony.pdf , last accessed 06/08/07.

Woods, D., and Guliani, G. Open Source for the Enterprise, O'Reilly Media, Sebastopol, CA, 2005.

Yin, R. K. Case Study Research, Design and Methods, 3rd ed. Newbury Park, Sage Publications, 2002.

An Evolutionary Framework for Language in a Digital Ecosystem

Stan Hendryx
Hendryx & Associates
stan@hendryxassoc.com

Abstract

The vocabulary of linguistic communities is organized to permit shared use of common terms by many communitites, while at the same time permitting a community to introduce new concepts or new terminilogy for existing concepts for its own use or for adoption by other communities.

1. Language and Communities

This paper concerns the organization and evolution of the language of linguistic communities as the communities themselves evolve, or as existing communities seek to formalize their vocabulary, with the aim of providing a model for an extensive information system that can adapt to accompanying changes in the language of evolving linguistic communities. For the purposes of this paper, we take the term "linguistic community," or simply "community," to denote a group of people who communicate with one another. The members of a linguistic community do not need to live or work in physical proximity to one another. A community may have as few as one person, or may be universal.

Our focus is with the lexical semantic aspects of language, not its syntax. Human language underlies human cognition, as we acquire knowledge and understanding through thought, experience, and the senses. Language is used when we think and as a means of expressing our thoughts. Our expressions are used to communicate. The issue we consider is the words used by a community to communicate meanings and how this stock of words, or vocabulary, changes as the community itself changes. The vocabulary used by a community changes when the communitiy uses different linguistic forms to communicate familiar meanings, and when the community expands or refines its repertoire of ideas. A given communication typically uses a vocabulary from one language, with occassional foreign words. We are concerned here with the structure of vocabulary within a community and among different communities.

Our interest is in situations that involve a machine intermediary in the communication, where such machine agents interact with humans and with other machines, and also create, reason about, and react or respond to the content of communications on behalf of their human principals. Although machine agents internally use a machine language, a natural human language, e.g. English, is used for machine interactions with humans and for describing machines' messages and behaviors as perceived by humans. Our interest includes situations where the machines may store and retrieve information, as well as perform synchronous or asynchronous interactions. This paper describes a metamodel for

1.1. Community Vocabulary

In a group of people who communicate with one another, each person has their own vocabulary. It is likely that no two people have exactly the same vocabulary. In a communicating group of people, the members share some vocabulary, consisting at least of the words they use to communicate. Each person in the group may have a different concept in mind that is represented by a particular word or phrase, and there may be some words or phrases that are not understood at all by some people in the group. These inconsistencies in the language of individuals are responsible for much miscommunication and misunderstanding.

Another source of misunderstanding is vagueness, use of abstract or general terms or an incomplete description to denote a more specific thing. Vagueness is often intentional. Our interest in this paper is in precise communication rather than vagueness. We are particularly interested in avoiding unintentional vagueness.

1.2. Semantic Communities and Speech Communities

To model linguistic communities, certain simplifications and idealizations are made. One such simplification is that the vocabulary of a community is considered to be an emergent property of the community

as a whole, not necessarily a property of any proper subset of the members of the community. This model means that each member of the community can expect to be able to communicate with any other member of the community using a shared vocabulary of the community. This model also provides for individuals to be members of several semantic communities. In this way, each person's linguistic capability can be modeled by the set of different communities in which they are members. A person can become a member of a community by acquiring the vocabulary and knowledge of that community, through education and experience.

It is recognized that in any group of people, different members have different expertise. Since each area of expertise tends to have some vocabulary that is distinctive to that area, it is natural to group the members of the community by their expertise, and to segregate the vocabulary of the community into vocabularies of different areas of expertise or subject matter. Thus, a community can be considered as unit that has other communities as its parts. The vocabulary of the community is the set union of the vocabularies of each of its part communities, plus vocabulary of its own. Note that it is possible for any person to be a member of more than one community unit or part. Having a community as a part means the members participate in that community.

Ordinarily, people communicate using a specified language. However, because of the diversity of natural languages, different languages an be used to verbalize and communicate a given set of concepts. Translation between languages is possible, with either a person or a machine as translator. Thus, it is useful to distinguish in our model between communities that share different concepts and communities that merely use different languages for a given set of concepts.

Two "community" concepts are useful in formalizing the language distinction, "semantic community" and "speech community". A semantic community is a community whose members nominally share a set of concepts and a knowledge domain, regardless of the language any member may use to represent what they know. The set of members of a "speech community" is a subset of the set of members of its semantic community that use a particular vocabulary, taken from a particular language. There is at least one speech community associated with each semantic community. The concepts represented by the vocabulary of each speech community is a subset of the concepts of the body of shared concepts of the semantic community.

In our community model, each semantic community has a body of shared meanings that comprises a body of shared concepts and a body of shared knowledge. The body of concepts includes noun concepts and verb concepts. The body of shared knowledge is a set of facts and non-factual propositions expressed using the body of shared concepts. The set of concepts associated with a vocabulary of a speech community is a subset of that of its semantic community's body of shared concepts. Translation of expressions is possible between the speech communities of a semantic community to the extent the speech communities share the concepts involved in an expression. To facilitate translation, at least one vocabulary should include the whole body of shared concepts, and this vocabulary needs to be based on the same language as used by the part semantic communities to represent all of their concepts. This common language is English; each concept should have an English definition. Other natural language vocabularies can be mapped to the same sets of concepts as the English vocabularies to identify the semantic links between vocabularies.

Semantic communities are structured around topics. Some topics are very broad, of general interest. Basic aspects of these topics are known by virtually the whole population. For example, the conceptualization of time, space, parts, and types, is generally applicable to any domain; the members of the semantic communities of these topics would each include virtually the whole population and every organization. The bodies of shared meanings of these universal semantic communities would be parts of virtually every other semantic community. Other examples of basic topics include buying and selling something, eating and drinking, or driving an automobile. Communities based on these kinds of topics may have a relatively small vocabulary that is considered to be universal. Other topics are more specialized and have a correspondingly smaller population in a semantic community devoted to these topics, for example, quantum mechanics or cellular biology or clearing and settlement of securities transactions or the car rental business; these may correspond to industries.

An enterprise semantic community might participate in its industry semantic community, i.e. have its industry semantic community as a part. The industry semantic community would participate in the universal semantic community. The enterprise semantic community would participate transitively in the universal community by virtue of its participation in the industry community. The body of shared concepts of the enterprise community may, if the industry model is well developed, consist mainly of individual concepts that are instances of industry concepts, e.g. "Acme Corporation" as an instance of industry concept "company", and "left handed end wrench, 20mm" as an instance of industry concept "product."

Considering the Universal Semantic Community as the root of a lattice of semantic communities, the set of all shared concepts available to a semantic community is the union of the body of shared concepts of each semantic

community from the semantic community to and including the root, via all paths.

1.3. Semantic Communities and Organizations

A semantic communitiy does not necessarily correspond to any organizational unit, such as an enterprise, or a division or department of an enterprise. A semantic community is a group of one or more people who have a common understanding of the things they have to deal with.

A semantic community may correspond to an organizational unit at some level of organizational hierarchy. Concepts in the body of shared meanings of an organizational semantic community would include individual concepts of things the organization owns or employs that are mentioned individually in the body of knowledge, specializations of more general concepts the organization deals with, and other concepts used by the organization but not included elsewhere. Well defined jargon that is peculiar to the organization would also be included. Different departments might be considered different semantic communities, each having its own special terminology. It is meaningful to consider what semantic communities an organizational unit participates in, what specialties it is involved in, when deciding how to structure the vocabulary of an organization, and to create specialized vocabularies for these that can be reused as parts of other organizations.

Refereneces

[1] Reference1

[2] Reference2

A Business Model of a Digital Business Ecosystem Negotiation Platform

[OPAALS Conference 2007]

Volker Hoyer
University of St. Gallen, MCM Institute
SAP Research CEC St. Gallen
St. Gallen, Switzerland
volker.hoyer@unisg.ch

Katariana Stanoevska-Slabeva
University of St. Gallen, MCM Institute
St. Gallen, Switzerland
katarina.stanoevska@unisg.ch

Abstract— **The process of business negotiations of complex products or services is an essential binding element in today's dynamic global markets in the age of the Internet of Services. The Digital Business Ecosystem paradigm provides a powerful concept for understanding such dynamic global business. In this paper, we present a business model of an envisioned Open Source decentralized negotiation platform focused on Small and Medium-sized Enterprises which is currently under development. In contrast to existing negotiation platforms such as Ebay in the B2C market or Ariba in the B2B market, the negotiation platform is based on a pure decentralized philosophy both on technical and business level following the Digital Business Ecosystems paradigm. The analysis of this new kind of B2B decentralized marketplaces is structure along seven major components according to the MCM Business Model Framework: features of the specific product or service, features for the specific medium (technology), potential customers, value chain, financial flow, flow of goods and services, societal environment.**

Keywords: Business Model, Business Negotiations, Digital Business Ecosystem, Internet of Services, Decentralization

I. Introduction and Motivation

Today's organizations are changing with respect to both organizational and technological perspective. Renowned scholars such as Thomas Malone argue that recent organizational forms have emerged from small, isolated businesses over centralized corporate hierarchies to loosely coupled networks [1]. Enabled by open and standardized information technologies, enterprises perform business transactions electronically and benefit by extending market reach, saving time, cutting costs and responding to customer queries agilely [2]. Especially, the next generation of the Internet, the so-called Internet of Services, serves as the technical foundation of decentralized organization units. By combining principles of Service-Oriented Architectures (composition of reusable building blocks) [3] and the Web 2.0 philosophy (user-self service and collective end-user intelligence) [4], it will facilitate the wide dissemination of many Internet resources provided by several service providers [5].

Already in 1958, Woodward argued in her work *"Management and Technology"* that technologies directly impact organizational parameters such as scope of management control and the degree of authority centralization [6]. To fully exploit the possibilities of new information technologies, Malone also argued that enterprises to expand our thinking to include *"radically decentralized organizations"* [7]. A concept combing both business and technical aspects in context with the trend towards decentralization is the Digital Business Ecosystem (DBE) paradigm. According to the vision of James Moore [8] the DBE concept has its root in biological ecosystems [9] which provide a powerful analogy for understanding a business network [10]. The key phenomena observed in nature such as competition, specialization, co-operation, exploitation, learning and growth, are also central to capitalist economy.

An essential element of such business ecosystems is the negotiation of alliances which enable companies to join core competences [11] as well as services and products into a complex offering. In the frame of the EU funded project Open Negotiation Environment (ONE), an Open Source self-learning negotiation platform focusing on Small and Medium-sized Enterprises (SMEs) is currently under development [12]. In contrast to existing negotiation platforms such as Ebay[1] in the B2C market or Ariba[2] in the B2B market, the ONE platform will be an open negotiation platform without a central governance cockpit where complex negotiation workflows and on going processes would be administrated. By representing such a new kind of decentralized marketplaces, the ONE project serves as a case to analyze the challenges implicated by following a pure decentralization philosophy both on organizational and technical level. To structure the analysis, we use the MCM Business Model framework [13] regarding along seven major components: features of the specific product or service, features for the specific medium (technology), potential customers, value chain, financial flow, flow of goods and services, societal environment.

The reminder of this work is organized as follows: In section 2, we define important terms such as DBE, SMEs and present the MCM Business Model Framework. In accordance with the seven component of the framework, we analyze in section 3 the challenges of the new kind of B2B marketplaces and elaborate how the envisioned ONE platform will handle

This work is partially funded under the IST program of the EU Commission by the STREP project "ONE" (Contract Number FP6-034744).

[1] http://www.ebay.com
[2] http://www.ariba.com

these issues. Finally, section 4 summarizes the findings and gives an overview of future work.

II. DEFINITION OF TERMS AND RESEARCH APPROACH

A. Digital Business Ecosystem

The synthesis of the concept of Digital Business Ecosystem (DBE) emerged in 2002 by adding "digital" in front of James Moore's business ecosystem vision [14]. He defined a business ecosystem as "*… an economic community supported by a foundation of interacting organizations and individuals - the organisms of the business world"* [8]. The generalization of the term to refer to a new interpretation of what socio-economic development catalysed by ICTs means was new, emphasising the co-evolution between the business ecosystem and its partial digital representation: the digital ecosystem. Thereby the digital ecosystem represents the technical infrastructure, based on a Peer-to-Peer (P2P) distributed software technology that transports, finds, and connects services and information over Internet links enabling networked transactions [14].

B. Small and Medium-sized Enterprises

Especially, Small- and Medium-sized Enterprises (SMEs) will benefit from online negotiations in business ecosystems. In the European Union – but in most areas of the world as well – SMEs are the predominant form of enterprises. Defined as companies that employ fewer than 250 persons and have an annual turnover not exceeding 50 million Euro and/ or an annual balance sheet total not exceeding 43 million Euro [15], SMEs are not necessary miniature versions of large enterprises. Related to their environment, structure, strategy and decision process, and psycho sociological context such as the dominant role of owner-manager, the following characteristics distinguish SMEs from large enterprises.

SMEs act on business markets that are not covered by large enterprises. Characterized by a *specialization and individuality,* many SMEs pursue a niche strategy that leads to certain strength in competition [16]. Compared with large enterprise, SMEs are strongly focused on their end-user allowing a high *proximity to markets*. Instead of focusing on exchangeable products of services for anonymous markets, SMEs provide service oriented at the customer's needs. Quickness to react and reorient themselves on business changes is a major characteristic of SMEs [17]. This *flexibility* in decision making and implementing organization changes is archived by preferring simplicity and flexibility regarding their processes and organizational structures [18]. SMEs are limited like all companies by tight resources, especially *missing IT literacy and financial resources* [17]. Also, smaller firms often lack coherent Information and Communication Technology (ICT) strategy or related skills. For instance, IT landscapes consist of *heterogeneous systems* reaching from Enterprise Resource Planning systems to spreadsheet-based island applications [18]. The growing internationalization of markets (*globalization*) since one decade affects the strategy of SMEs due to deregulation and liberalization of former market barriers. The change of new potential business partners involves an enormous adaptation pressure for SMEs which have less experience than large enterprises in global e-Business [19].

C. Research Approach

The Media and Communications Management (MCM) Business Model Framework has been established as a means for structuring and analyzing arbitrary business models on the basis of generic components and has been successfully applied to the field of online services [20]. A number of other definitions of the term business model exists [21, 22] which are different with respect to scope and structure and do not allow for through and systematic analysis. The MCM Business Model leverages the definition proposed by Timmers [23] and its business model components as a foundation and enriches it with additional relevant aspects. The resulting MCM Business Model Framework is presented in the figure below.

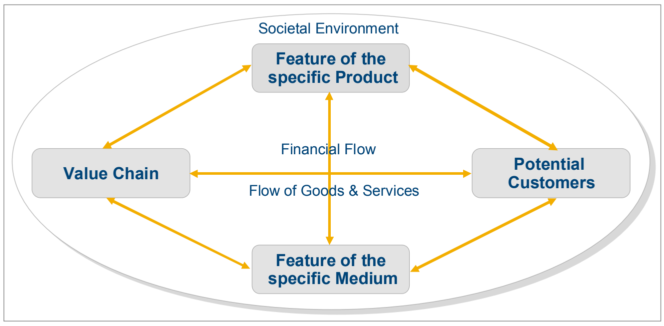


Figure 1. The MCM Business Model Framework [13]

The seven major components are described in the following: First the *Features of the Specific Product* comprises the actual design of a product or service, the way it is perceived and consumed by the customer and the value proposition for the customer. The component *Features of the Specific Medium* defines possibilities for transaction and interaction via certain media between the stakeholders of a business model from a technical point of view. The *Customers* component refers to the target groups of an offered product or service and explains their respective business needs. Fourth, the *Value Chain* component is devoted to reflecting all players that are involved in the production and delivery of a product and their respective interrelationships. The component *Financial Flow* identifies in which way the products and services are monetized and explain the roles different stakeholders play. *Flow of Goods and Services* describes the stakeholders' activities that are essential for the creation of the product or services. Last, the *Societal Environment* reflects relevant outside influences on a business model (e.g., legal aspects and competitive situation).

III. BUSINESS MODEL OF A DECENTRALIZED B2B NEGOTIATION PLATFORM

In the following paragraph, we analyze the envisioned decentralized ONE platform based on the MCM Business Model framework's seven major components.

A. Features of the Specific Product or Service

First, we will focus on the provided features from the viewpoint of a company. As depicted in the figure below, the so called ONE node can be deployed on participant's resources to keep the control in the company's borders. In case of the before mentioned SMEs characterized by a limited known-

how, it is also possible to use alternatively an external ONE node from a business partner to conduct electronic negotiations. This flexibility covers the special situation of SMEs.

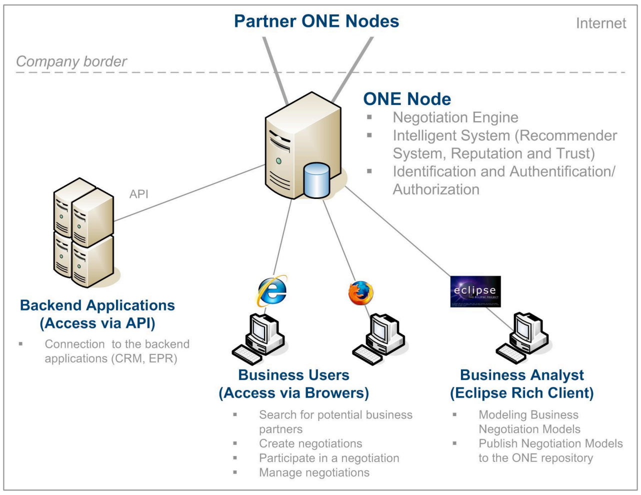


Figure 2. Features of the ONE B2B Negotiation Platform

1) Business Negotiation Modeling Environment

To provide a high flexibility, ONE will not be build around a set of predefined and fixed number of negotiation workflows as provided by Ebay or Ariba. Instead it makes use of a modeling language for describing negotiations process and formation of criteria like price, quantity or delivery conditions according to the business environment. Similar to business process modelling languages such as BPMN[3], a user can define a custom negotiation workflow taking into account, for example, negotiation rules, legal rules, pricing policy, reserver prices, logical process description and other individual specification. Based on the Eclipse Rich Client architecture, a so-called ONE Factory provides a user-friendly business negotiation modelling environment allowing per drag and drop the specification individual negotiation workflows according to the different business requirements. There are three main steps to accomplish a negotiation process in ONE:

- *Information Model* - the business domain (the items/ issues that are negotiated)
- *Protocol Model* – the type of negotiation (describing the negotiation workflow)
- *Negotiation Model* – the mix between Information Model and Protocol Model

2) Business Negotiation Engine

By publishing the modelled negotiation workflow any user can create a new negotiation based on the specified negotiation characteristics as modelled with the business negotiation modeling environment. The envisioned ONE Web portal provides an intuitive GUI to manage the business negotiation engine. It has no central governance cockpit or console where to administrate negotiation models and on going processes. The execution of the negotiation process will be hosted in participant's hardware resources, not in a central node. To interact with the ONE negotiation platform a user need only a browser. No local installation is required. Therewith, ONE considers the specific environments of SMEs which are characterized by limited IT infrastructures and know-how.

The ONE Business Negotiation Engine covers all activities related to the runtime of business negotiation. By *"setup a negotiation"* a user select a negotiation model as introduced in the previous section and configures the negotiation according to the specific requirements. After the announcement of the negotiation, the *"running phase"* coordinates the workflow meaning the interaction between the negotiation owner and the participants in terms of offer and counter-offer. Finally, in the "termination phase", the owner decides to terminate the negotiation based on the submitted offer of the participants. The result is an agreement serving as input for a legal binding contract.

3) Intelligent System

Additional to the traditional execution of business negotiations, ONE will actively support the human negotiators by exploiting automatic learning techniques applied to the goal of learning the best negotiation strategies. Consisting of a recommender system, a learner and a Distributed Knowledge Base (DKB), the user will be supported on how best to conduct electronic negotiations. The personal negotiation recommender that will support a user will be hybrid and conversational, and will compute the recommended actions exploiting a distributed knowledge base, which expands the personal knowledge base of an actor, and makes possible to speed-up the policy learning process, exploiting experiences gathered not only by the supported user but also by a community of trusted partners.

Methods for characterizing and analysing a large population of negotiating partners viewed as a complex ecosystem will be studied. The collective and emergent behaviour will be analysed, the heterogeneity and diversity of the business ecosystem will be characterised and self-stabilizing methods, such as mechanisms to help the system to get rid of harmful behaviours, will be applied to the specific context of the negotiation. ONE will also explore a novel research direction with respect to the State-of-the-Art of reputation-based models – combining reputation mechanisms with the learning and evolution aspects of the ONE platform. It will explore possible synergies between the reputation techniques and the learning and recommendation approaches used in ONE. Those synergies are expected to evolve the reputation rating scheme to new evolutionary criteria. Thus, the reputation-based trust will evolve SMEs in a new dimension of establishing trust relationships.

B. Features of the Specific Medium (Technology)

Based on Open Source technologies, a Model-Driven and a "pure" P2P architecture, ONE will follow the DBE paradigm as on open evolutionary platform.

1) Open Source

Open Source is a development method for software that harnesses the power of distributed peer review and transparency of process. The promise of Open Source is better quality, higher reliability, more flexibility, lower cost, and an

[3] Business Process Modeling Notation (BPMN)

end to predatory vendor lock-in. Quality, flexibility, independence from vendors and cost savings are driving the Open Source adoption in ONE. Open Source methodologies are a set of approaches for managing sources, or more generally contents, in an extremely agile manner. Contributors on a particular Open Source project often do not stay under the same roof and usually they communicate with the support of the Internet. The project coordination is managed using very simple textual communications, without any formal style. Developers have created a set of self-made tools used to manage their collaboration, and have demonstrated that they are able to achieve extremely valuable objectives with resources usually considered inadequate. Build tools (*Maven*), Wikis (MediaWiki), version control systems (*CVS*), integrated development environment (*Eclipse*) are common by-products of Open Source methodologies. Moreover there are a lot of Open Source projects adopted by ONE as basic components for the Business Negotiation Engine, the most commonly known are: *Apache Tomcat*, the leading open source JSP and Servlet Container from the Apache Foundation; *JBPM*, the leading Open Source workflow management system; *MySQL*, the leading Open Source relational data base management system; *Jackrabbit*, a fully conforming implementation of the content repository for Java technology API.

Furthermore, joining the SourceForge.net community is a way to share and promote the ONE's Digital Business Ecosystem system philosophy, potentially reaching thousand of IT managers, software developers and end-users.

2) Model-Driven Architecture

Model-driven architecture (MDA, software design approach launched by the Object management Group) provides a set of guidelines for structuring specifications expressed as models, where functional business needs are abstracted from the underling implementation. Models are a standard based common language to express information or knowledge or systems in a structure that effectively capture the rules governing a specific domain. Models can better support evolving business needs improving

- *Flexibility.* Because they can be changed easily, without the constraints of a specific programming language.
- *Quality.* Because they force developers to think ahead.
- *Agility.* Because they can be re-used and adapted to different architectures and platforms.

The Business Negotiation Modeling Environment is the environment provided by ONE to realize the implementation of MDA concepts. Based on the Eclipse framework, three Eclipse sub-projects are used to implement the ONE Factory:

- *Graphical Editing Framework* allows developers to take an existing application model and quickly create a rich graphical editor.
- *Eclipse Modeling Framework (EMF)* project is a modeling framework and code generation facility for building tools and other applications based on a structured data model.
- *Graphical Modeling Framework (GMF)* provides a generative component and runtime infrastructure for developing graphical editors based on GEF and EMF.

With the GMF framework it is possible to build a graphical editor which produces models that adhere to the specifications of the three main models Information Model, Protocol Model and Negotiation Model.

3) Peer-to-Peer Architecture

ONE will support the Business Negotiation Engine with a "pure" P2P network infrastructure. ONE does not have the notion of clients or servers, but only equal peer nodes that simultaneously function as both "clients" and "servers" to the other nodes on the network. It is fully decentralized and the peers are fully autonomous. Each ONE node can be subject or object of a request, provide services, be proactive, support a negotiation; the negotiation and infrastructure services are no more concentrated on a specific server but distributed between all the nodes of the network. Each peer can enter, join or leave the network anywhere anytime. It is capable of discover services exposed by other peers.

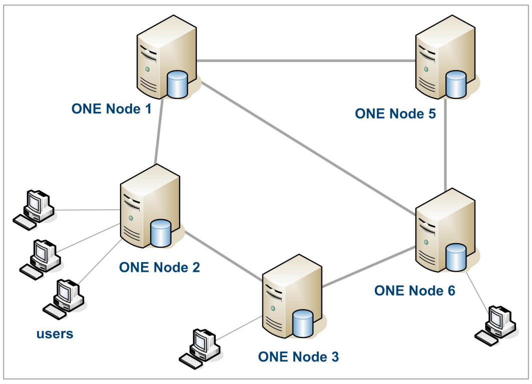


Figure 3. ONE P2P Architecture

ONE provides a technical framework with different features at different levels with the goal to design a scalable P2P service network: service creation and deployment; distributed service registry; several inter ONE and extra ONE communication protocols; replication of content repositories; application security (identity and authentication management). Services are the main software concepts that are used to provide functionalities in a distributed environment. They are deployed in ONE node and users can access these services with different applications that can be either rich-client, with the features and functionalities of a traditional desktop applications, or thin-client, usually implemented as Web application. It is a good practice that the design and the development of a service should decouple the business logic from the way that the service is accessed. For this reason the ONE environment provides features for supporting pluggable binding techniques and an easy configuration mechanism.

C. Potential customers

Potential customers of the ONE negotiation platform are basically any enterprise or individual, who is interested in participating in online B2B negotiations.

1) Business-to-Business (B2B)

Basically, B2B negotiations can be interpreted as electronic hubs that bring together a large number of buyers and suppliers and automate business transactions. In contrast to consumer negotiations focusing on simple negotiation processes as provided by Ebay (English auction, Dutch auction), business negotiations are characterized by *complex interactions* with any kind of variants. Nested negotiations spanning over several negotiation levels require a process control to handle such dynamic and flexible environments. Besides the standard negotiation issues price which is the predominant negotiation issue in consumer negotiations, B2B negotiations require individual *complex negotiation issues* (e.g, delivery conditions or quality aspects). Additionally, B2B negotiation marketplaces have to be *integrated into legacy systems*.. Additionally, B2B transactions required a high *reliability* meaning the access to negotiations details 24 hours a day, 7 days a week. Especially in the last moments of a negotiation process, the platform has to be able to support high volumes of contracts and bids. High response time or unavailability will jeopardize the business success of the platform. The last characteristic of business negotiation is *trust*. In case the community doesn't trust the ONE platform, no enterprise will join the ecosystem and will invest resources to refine the ONE platform.

2) Covering the "Long Tail"

In contrast to most commercial negotiation environments where some negotiation features are implemented and the where the network of intermediaries/ suppliers is static and centrally regulated, the ONE platform aims to provide an open negotiation platform integrating SMEs in the ecosystem. According to Chris Anderson the future of the media and communication markets is in the millions of niche markets at the shallow end of the bitstream [24]. In context of online negotiations this means that predefined negotiation workflows and negotiated issues don't cover the profitable "long tail". Instead of a traditional static concept, ONE will implement a dynamic negotiation platform.

The ONE Portal is designed as an application that can be adapted to various languages and regions easily, cost-effectively, and in particular without engineering changes (*internationalization*). The community doesn't distinguish between users across countries. When the product is available for free, it is amazingly easy to make a product global. In this sense, ONE aims to achieve a *global platform*. A personalized and *individual workspace*, so-called MyONE, allows the user to store information about negotiation he initiated and also his bids to negotiations initiated by other companies, to collect unstructured documentations related to products/ services that are involved in a negotiations, to store certificates, to store personal contacts, to store logs of past negotiation and aggregated data from the learner and recommender. The heterogeneous negotiation workflows require *individual negotiation workflows*. Any kind of enterprises is able to model his specific negotiation workflows according to the business environment.

D. Value Chain

Existing linear closely wired value chains [25] characterized by a one-to-one connection among business partners will be replaced by adaptive business networks to achieve seamless processes and real-time business across enterprises. React flexible and on demand both on business and technical environment changes is a new competitive differentiator with strategic importance.

1) Business Ecosystem

As introduced in the motivation, an understanding of the laws in nature will lead to a greater understanding of the workings of companies [26].

TABLE I. CHARACTERISTICS OF BUSINESS ECOSYSTEM

Characteristics	Support by the ONE negotiation platform
Large Number of participants	▪Not technical or organizational limitation for participating in the ONE business ecosystem ▪High scalability guaranteed
Interconnectedness	▪SOA-based P2P architecture
Interaction	▪No legal contract for participating ▪Loose coupling of business negotiation
Shared fate	▪ONE community will own the environment ▪Open Source strategy and Web 2.0 philosophy
Competition and Cooperation	▪Supporting private and public negotiations ▪Any kind of negotiation workflow (English auction, Dutch Auction, Multi-party negotiation)
Coupled with changing environment	▪Adaptive negotiation workflow ▪Model-Driven Approach (MDA)
Dynamics	▪Flexible reaction on technical and business changes ▪Easy join and leave process into the ecosystem
Conscious choice	▪ONE supports the user in the decision process.
Aims at innovations and commercial success	▪Free upload of negotiation models ▪Evolutionary platform covering the "long tail" ▪Extension continuously the ecosystem knowledge

Both the followed business ecosystem paradigm and the biological ecosystem are characterized by a *large number of loosely interconnected participants* who depend on each other for their mutual effectiveness and survival. Biological species in ecosystems *share their fate* with each other the same way that firm in a business ecosystem [10]. Failure or success of one organization may echo through the whole system. In addition, members of a business ecosystem work *co-operatively and competitively* to support new products, satisfy customer needs, and eventually incorporate the next round of *innovations* [27]. In contrast to networks which focus only on the cooperation, business ecosystems base their success on both competition and cooperation [8]. In addition to *changing environments*, this leads to unpredictable dynamics. The interaction between the business ecosystem and its environment is a source of unpredictable *dynamics*. Since business ecosystems assume constant changes the people make conscious decisions.

2) Intermediary

By abstracting from the ONE system architecture, there exist analogies to the concept of Service-Oriented Architectures (SOA). According to the widely accepted normative OASIS SOA Reference Model [3] the major components of a basis SOA and their possible interactions are: A *service provider* publishes his service via a *service intermediary* where a *service consumer* can find it and subsequently may bind to the service provider.

The concept of SOAs is limited mostly on technical discussion by today. The ONE platform will extend the service philosophy from a technical perspective to the business ecosystem. Thereby, ONE interacts as a neutral intermediary. In this function, ONE operates independently and does not favour buyers (service consumers) over sellers (service provider) or the opposite. ONE will be equally attractive to buyers and sellers. In such markets, neutral intermediaries add value by reducing transaction cost (aggregating) and improving matching (providing liquidity). If only one side of the market is fragmented, the benefits are greatly reduced for the non fragmented side [28]. In case of the envisioned ONE platform following an open and individual marketplace strategy, both market sectors are fragmented. In this context, this kind of neutral intermediary is the true market markers in Business-to-Business exchanges bringing buyers and sellers together [28].

E. Financial Flow

In the Open Source paradigm, multiple entities (individuals, companies) come together to develop a software. Generally, the initial development is done by a single entity as in the in-house. The software is released to the public as soon as it is useful to others before it would be considered a finished product. Only when the software becomes useful to others does the Open Source paradigm work fully because only in this case the community has an incentive to use it and to extend the software with additional features that are of interest to them.

1) Profit Potential of the ONE Open Source Paradigm

Not all Open Source products have the same profit potential. To analyse the profit potential of the ONE Open Source paradigm, we use two dimensions: customer applicability and relative product importance [29]. *Customer applicability* refers to the proportion of the market that can benefit from the software. If a product is being designed for a rarely used platform, only a small proportion of customers will be able to benefit from it meaning the level of applicability is small. *Relative product importance* refers to how important the product is to the business work. For example, the relevance of negotiation tools in the service economy will be increased and will play a major role in business transactions [29].

The products with the highest profit potential have high relative product importance and high customer applicability. These are the *stars*. They have large developer communities supporting them and are characterized by the greatest direct and indirect marketing support. In context of operating systems LINUX is such a star product. *Mainstream products* on the other side are products everybody can benefit from. However, they are not critical to the business work. On the other side, *low-profile nichers* products serve a specific niche. They are never going to be dominant products that will be used by many customers. Finally, the products with high relative product importance and low customer applicability are the *high-profile nichers*. By regarding very highly within a specific niche these products are limited to small customer segment even if they are important in the business environment.

ONE will cover a star product by focusing simultaneously on the “long tail” of online negotiations and high customer applicability. In future, cross-organizational business transactions as envisioned by Don Tapscott with the term *Wikinomics*, [30], require online negotiations as the foundation and starting point for doing global business. The dynamic business environment needs continuously online negotiations between the business partners to negotiate specific issues such as price, delivery conditions or quality aspects. That's why the relative importance of ONE is high. Additionally, millions of small contributions from users are opposed to a few significant contributions from “important” users. A large ecosystem has to build up around the ONE platform to cover this profitable “long tail”. The integration and identification of heterogeneous customers and business partners is the key of the economic success of ONE.

2) Web 2.0 Revenue Models

Besides the Open Source strategy, ONE will also generate revenue with the Web 2.0 philosophy [4]. Being successful in the Web 2.0 era is about more than just generating net revenue. It's about keeping your market share. To stay successful you have to maintain critical mass as described above in context with the Open Source profitability so that the continually contributed and enriched data stays better than the next person. It's about having the best content and functionality on the Web. In accordance with this definition, the ONE negotiation platform will combine both the net revenue and the shared content produced by the customers. Already in 1980, Alvin Toffler coined the term “prosumer” when he predicted that the role of producers and consumer would begin to blur and merge [31]. Toffler envisioned a highly saturated market as mass production of standardized products began to satisfy basic customer demands.

By taken this trend into account, the ONE platform will generate revenue with the several issues. In traditional marketplaces, you find complex *pay-per-use models*. Dependent on the executed business negotiation transaction, the user has to pay for the usage. An established strategy to generate revenue in Web 2.0 is to provide additional *premium services* which are not for free. The ONE platform and their members could extend the platform with analytical features or integration tools to existing legacy systems. In this sense, the decentralized and modularized architecture of ONE allows every service provider in the ONE ecosystem the offering of specific premium features. Furthermore, any provider of a ONE node is able to integrate banner *advertisements* on the ONE portal. Advertising revenue is in Web 2.0 the main revenue source. In addition, the ONE recommender system is a good starting point to analyze the Web users’ behavior. The gained knowledge can be the basis for new negotiation strategies. Established concepts well known in the world of grid computing serve as input for a paying system related to the compute power consumed. Any provider of a ONE node can redirect their cost for run the technical infrastructure to the users (*pay-per-compute power*). In context with the Open Source paradigm, the ONE business ecosystem will also generate revenue from related *support services* like system integration, product support, tutorials or user documentation. Empirical research results have proved that neutral intermediary have to transform into solution providers providing services to establish private exchange [32].

F. *Flow of Goods and Services*

As explained in context of the value chain investigation, the ONE platform acts as a neutral intermediary. It serves as technical platform bringing the two parties sellers and buyers together. The P2P architecture and the Distributed Knowledge Database (DKB) as well as the common ownership of the platform leads to distributed content and service location. Additionally, ONE follows the Web 2.0 philosophy of a perpetual beta meaning applications are no longer software artefacts. They are ongoing services [4]

1) *Decentralized Information Flow*

The information flow during the negotiation process is decentralized by default. Both the source code and the user-generated content are shared within the ONE business ecosystem without *any central control* allowing an innovative negotiation environment for both seller and buyers.

According to the business ecosystem paradigm the ONE platform will interact as a *neutral intermediary*. Every enterprise can download the source code of the ONE platform and can setup a ONE node. This independent system architecture and philosophy is necessary to support a "pure" decentralized information flow. No restrictions concerning control functions exist on technical layer. Further on, the key of successful marketplace is the provided content. The new role of customers as content producers (so called prosumer) implicates shared user-generated. According to Surowiecki *"groups are remarkably intelligent, and are often smarter than the smartest people in team"* [33]. He argues that a crowd's collective intelligence will produce better outcomes than a small group of experts if four basic conditions are met: Diversity of opinion, independence of members from one another, decentralization and a good method for aggregating opinions. The diversity brings in different negotiation workflows. Independence keeps people from being swayed by a single opinion leader as followed by Ebay or Ariba. The including all opinions guarantee that the ONE negotiation platform is smarter than if experts had been in charge.

2) *Perpetual Beta*

What version of Web 2.0 platforms such as Google or Ebay is currently running? In the Web 2.0 Internet area, users think in terms of services not package software. They expect these services to just be there and to improve over time. No versions, no installations, no upgrades needed. The traditional design-develop-test-ship-install cycle of packaged software is ending. Software has become a service – a service that is always on, always improving [4]. The success of the envisioned ONE platform relies on adoption of the perpetual beta development model in which software is continuously refined and improved. According to the DBE paradigm, the evolution of the platform is the key to handle the heterogeneous technical and business environment. The interpretation of customers (both buyers and sellers) as active participants in the evolution of the ONE platform leads to a *closed relationship* to the business ecosystem. According to the Wikinomics principles [30], the customers have to be integrated as co-innovators [34]. In this sense, they can upload new and innovative negotiation workflows to the ONE platform.

Required business changes caused by a growing dynamic environment are handled by a high flexibility. The closed relationship to customers allows reacting in *real-time on business changes*. Because there will not be a central controlled process to upload new negotiation models defining a negotiation workflow, the heterogeneous market requirements can be covered by the ONE platform. The active integration of the customers *reduces the risks* of new product development [35]. Innovations and new services will not cause a high risk because the potential customers influence directly the development of services at an early stage. By providing open APIs and by orienting on standards, a seamless interoperability is the foundation to make *quantifiable decisions* in the negotiation process. Besides the provided real-time data, the recommender system supports the ONE user in an evolutionary manner.

G. *Societal Environment*

Finally, we investigate the societal environment of ONE spanning from legal aspects to a competitive situation.

1) *Legal Aspects*

In order to support electronic transactions in a similar way as conventional transactions, legal aspects have to be considered. Even if the ONE platform will not negotiate a legal binding contract, the evolutionary approach according to the DBE paradigm requires the integration of legal issues. In context with electronic contracts, the following legal requirements have been identified: a clear identification of the negotiating parties, a clear indication of the subject of the negotiation, a clear indication of the time period of validity, a valid signature of the involved parties and non-repudiation meaning nobody should be able to change the content of the negotiation [36].

The ONE platform will meet the legal requirements by developing and integrating an identity management component in the platform supporting the management of partners' identities in a decentralized and P2P fashion [12]. It will include three security primitives: authentication, integrity and confidentiality.

2) *Competitive Situation*

Last but not least, we regard the competitive situation. To structure the analysis, we use the Porter Five Forces framework [37]. It is used to make an analysis of the attractiveness (value) of an industry structure by the identification of five fundamental competitive forces: entry of competitors, threat of substitutes, bargaining power of buyers and sellers, rivalry among existing players.

The threat of new entry is quite high. If anyone looks as if the ONE platform is making a sustained profit, new competitors can come into the market easily. Negative effect is the reduction of profits. Potential competitors such as Ebay or Ariba could change their strategy and could follow a combined Open Source and Web 2.0 strategy as well. Due to the fact that these competitors can refer to experience in online negotiations and there exist no technical entry barriers, the threat of new entry is quite high. Further on, Ebay or Ariba have a strong customer base. Both the supplier and buyer power is neutral. By following the Digital Business Ecosystem paradigm, ONE

focuses explicitly on SMEs characterized by their limited resources (financial and technical) and know-how. Combined with the changing role of customers to prosumers [31], the number of suppliers will be grown. On the other hand, the open platform allows replacing easily suppliers. To cover the financial profitable "long tail" [23], ONE provides in contrast to existing negotiation platforms a flexible and individual negotiation environment to handle dynamic negotiation workflows. This flexible environment could not provided by any other services or products. Summering up, the competitive rivalry will be extremely high due to the strong traditional players of Ebay in the B2C market and Ariba in the B2B market. By focusing more and more on core competences [11], enterprises outsource products and services to its business ecosystem.

IV. Conclusion and Future Work

In the frame of the work, we have analyzed a new kind of decentralized B2B marketplaces following the Digital Business Ecosystem paradigm. In accordance with the seven major components of the MCM Business Model Framework, the challenges of a pure decentralization approach on both business and technical level are identified. By means of the ONE project, we have presented solutions to handle the trend towards decentralization in the age of the Internet of Services. Future work will deal with the evaluating of the open negotiation platform implementing the approaches in real-word scenarios which are part of the ONE project.

V. References

[1] Malone, T. (2004). *The Future of Work: How the New Order of Business will Shape your Organization, your Management*. Harvard Business School Press. Boston, MA.

[2] M. Porter (2001). *Strategy and the Internet*. In: Harvard Business Review, Vol. 79(3): 63-78.

[3] McKenzie, C.M.;Laskey, K.; McCabe, F.; Brown, P.F.; Metz, R. (2006). *Reference Model for Service-Oriented Architecture 1.0*.

[4] O'Reilly, T.. (2005). *What is Web 2.0*. Available at http://www.oreillynet.com/pub/a/oreilly/tim/new/2005/09/30/what-is-web-20.html.

[5] Schroth, C.; Janner, T. (2007). *Web 2.0 and SOA: Converging Concepts Enabling the Internet of Serives*. In: IT Professional, Vol. 9(3): 36-41.

[6] Woodward, J. (1958). *Management and Technology*. H.M.S.O., London.

[7] Malone, T. W. (1999). *Is Empowerment just a Fade? Control, Decision-Making and Information Technology*. In: BT Technology Journal, Vol. 74(4): 141-144.

[8] Moore, J.F. (1993). *Predators and Prey – A new Ecology of Competition*. In: Harvard Business Review, Vol. 71(3): 94-102.

[9] Rothschild, M. (1990). *Bionomics Economy as Ecosystem*. Henry Holt and Company. New York.

[10] Iansiti, M.; Levien R. (2004). *The Keystone Advantage: What the new Dynamics of Business Ecosystems Mean for Strategy, Innovation and Sustainability*. Harvard Business School Press.

[11] Hamel, G.; Parahalad, C.K. (1990). *The Core Competence of the Corporation.*. In: Harvard Business Review, Vol 68(3): 79-91.

[12] Telesca, L.; Finnegan, J.; Ferronato, P.; Malone, P.; Ricci, F.; Stanoevska-Slabeva, K. (2007). *Open Negotiation Environment: An Open Source Self-Learning Decentralized Negotiation Framework for Digital Ecosystems*. In: Proceedings of the IEEE Digital Ecosystems and Technologies Conference: 186-191.

[13] Hoeeg, R.; Meckel, M.; Stanoevska-Slabeva, K.; Martignoni, R. (2006). *Overview of Business Models for Web 2.0*. In: Proceedings of GeNeMe 2006, Dresden: 23-37.

[14] Nachira, F.; Dini, P.; Nicolai, A. (2007). *A Network of Digital Business Ecosystems for Europe: Roots, Processes and Perspectives*. Available at http://www.digital-ecosystems.org/doc/discussionpaper.pdf.

[15] European Commission (2003). *The new SME definition – User Guide and Model Declaration*. Available at http://ec.europa.eu/enterprise.

[16] Porter, M. E. (1980*). Competitive Strategy*. The Free Press. New York.

[17] The Economist Intelligence Unit (2006). *Thinking big: Midsize Companies and the Challenges of Growth*. Available at http://eiu.com.

[18] European Commission (2006). The European e-Business Report 2006 – A Portrait of e-Business in 10 Sectors of the EU Economy. Available at http://ec.europa.eu/enterprise.

[19] Ibielski, D. (2002). *SMEs on the Move to Globalization*. Available at http://www.unece.org/indust/sme.ibielski.htm.

[20] Stanoevska-Slabeva, K.; Hoegg, R. (2005). *Towards Guidelines for the Design of Mobile Services*. In: Proceedings of the ACIS 2005 conference.

[21] Afuah, A.; Tucci, C.L. (2001). *Internet Business Models and Strategies*. McGraw-Hill International Editions. New York.

[22] Osterwalter, A. (2004). *The Business Model Onotoly*. HEC Lausanne.

[23] Timmers, P. (1998). *Business Models for Electronic Markets*. In: International Journal on Electronic Markets and Business Media, Vol. 8(2): 3-8.

[24] Anderson, C. (2006). *The Long Tail: How endless choice is creating unlimited demand*. Random House Business Books, London.

[25] Porter, M.E. (1985). *Competitive Advantage. Creating and Sustaining Superior Performace*. The Free Press, New Yrok.

[26] Lewin, R. (1999). *Complexity. Life a the Edge of Chaos*. The University of Chicago Press, Chicago.

[27] Adner, R. (2006). *Match Your Innovation Strategy to Your Innovation Ecosystem*. In: Harvard Business Review, Vol. 84 (4): 98-107.

[28] Kaplan, S.; Sawhney, M. (2000). *E-Hubs: The new B2B Marketplaces*. In: Harvard Business Review, Vol. 78 (3): 97-104.

[29] Krishnamurthy, S. (2005). *An Analysis of Open Source Business Models*. In: Feller J. et al. (eds). Perspectives on Free and Open Source Software. MIT Press, pp. 279-296.

[30] Tapscott, D. Williams, A. C. (2006). *Wikinomics: How Mass Collaboration Changes Everythink*. Portfolio, New York.

[31] Toffler, A. (1980). *The Third Wave*. Bantam Books, Toronto.

[32] Yoo, B.; Choudhary, V.; Mukhopadhyay, T. (2007). *Electronic B2B Marketplaces with Different Ownership Structures*. In: Management Science, Vol. 53 (6): 952-961.

[33] Surowiecki, J. (2004). *The Wisdom of Crowds. Why the Many are Smarter Than the Few*. Doubleday, New York.

[34] Thomke, S.; von Hippe, E. (2002). *Customers as Innovators. A New Way to Create Value*. In: Harvard Business Review, Vol. 80 (2): 74-81.

[35] Ogawa, S.; Piller, F. (2006). *Collective Customer Commitment: Reducing the Risks of New Product Development*. In: MIT Sloan Management Review, Vol 47 (2): 65-72.

[36] Gisler, M.; Stanoevska-Slabeva, K.; Grenz, M. (2000). *Legal Aspects of Electronic Contracts*. In: Proceedings of Infrastructures for Dynamic Business-to-Business Service Outsourcing (IDSO'00), Stockholm.

[37] Porter, M. E. (1979). *How Competitive Forces Shape Strategy*. In: Harvard Business Review, Vol. 57: 86-93.

Community visualisations in Open Knowledge Space

Uncovering rabbit holes in a digital ecosystem

[OPAALS Conference 2007]

Jukka Huhtamäki
Hypermedia Laboratory
Tampere University of Technology
Tampere, Finland
jukka.huhtamaki@tut.fi

Abstract*—Visualisations have a long tradition in the research of social networks. The increase in the popularity of various social networking and community services in modern Web has emphasised the need for high-end visualisation targeted for the end users of the services. We see the community visualisations as means for a member of the community to uncover "rabbit holes" leading into new and unexpected information landscapes, to both discover unknown members with complementing interests and find unknown interests of already known members, and, in all, to become a working part of a community. In this paper, we lay the foundations for community visualisations related to Open Knowledge Space, a collaborative knowledge-creation platform currently developed in OPAALS, a European Network of Excellence. In addition, we discuss the connections between community visualisation, social network analysis, Semantic Web and Web 2.0 and introduce a set of community visualisation prototypes designed to support the community building and knowledge-creation processes in Open Knowledge Space.

community visualisation; digital ecosystems; social network analysis; data processing pipelines

I. Introduction

Finding relevant information and people with similar interests is a major issue in any digital community or, in general, any *digital ecosystem* (see http://www.digital-ecosystems.org/). The information content in a digital community is always, at least partly, informal and vague and therefore hard to manage with machines. Also finding the proper location to input new information may be challenging - not to mention becoming a working part of the community. We argue that *community visualisations* can provide a member of an ecosystem means to solve these problems.

The overall objective of *information visualisations* can be characterised as to serve as an amplifier of user's cognition by giving insight to the data visualised [1]. The *visual information seeking mantra* by Shneiderman distils the usage process of a visualisation in the following way: "overview first, zoom and filter, then details-on-demand" [2]. The mantra suggests that visualisation is a process rather than a static image.

At best, the visualisations can act as "rabbit holes" in a digital ecosystem through which users are able to find and situate themselves into new information landscapes (cf. [3]). The rabbit hole is a metaphor inspired by Lewis Carroll that "has also come to signify any event which triggers a completely unexpected, bizarre situation or paradigm" [4]. McGonigal has introduced the metaphor in the context of game design and provokes by stating that rabbit holes create community, build a learning culture, increase sensitivity, produce vertigo and virtualise everyday life [5].

Creating a personal connection between the user of the visualisation and the visualised data can be seen as a motivation factor for using the visualisation and, more importantly, the creation and refinement of the data that is visualised. Heer even claims that when visualisation users are "successfully able to 'write themselves into the data', the hope is that they might then continue on to more fully explore their data environs" [1].

Social networking is an uprising phenomena in the state-of the-art Web or *Web 2.0* [6]. The social network data, essentially the member profiles and the articulated relationships between the profiles, is often quite formal, thus enabling the mechanical data analysis (see [7], [8]). Once the data can be analysed, interactive visualisations providing means to navigate the social network data and gain overall insight on the community structure can be created (see [9]).

In this article, we start by describing the context we are working in, continue with an overview of our evolving theoretical framework, discuss the social networking in general and particularly in the context of Open Knowledge Space (OKS). We also introduce two early community visualisation prototypes and discuss their design and implementation. We finish the paper with a general discussion, some early conclusions and directions mapping our work in the future.

A. What is OKS?

Open Philosophies for Associative Autopoietic Digital Ecosystems (OPAALS) is a multidisciplinary Network of Excellence for developing the science and technology behind digital ecosystems (http://opaals.org) coordinated by London School of Economics and Political Science (see http://www.lse.ac.uk/). To support the multidisciplinary research work done across Europe, Africa, Asia, and South-America as well as the efficient dissemination of the research

results, the OPAALS community is designing and constructing Open Knowledge Space (OKS). The work is done in an evolutionary manner, largely by tailoring and integrating existing tools for creating, refining, and accessing the information in OKS related to the OPAALS research. Interestingly, OKS is a digital ecosystem about digital ecosystems.

While OKS can not be easily formalised, we see that it includes at least the following web-based tools: 1) *OKS wiki* offers a platform for collaborative writing. 2) *OKS blogs* are informal journals written by the members of OKS community. While not being able to edit each others' blogs, members may add comments to any blog writing. 3) *OKS members profiles*. Since OKS does not currently provide any formal tools for member profile creation, we have used the OKS wiki member pages for capturing the profile information.

B. *Some concepts*

Working in a multidisciplinary domain is difficult for many reasons, the lack of shared concepts is one of the main issues. In the following, some of the concepts used in this paper are introduced and discussed.

In general, social science data can be categorised as follows. A*ttribute data* is related to the properties of an agent in the network including the properties, qualities and or characteristics defining the agents as individuals in a community. The connections, contacts and ties between the network agents is referred as *relational data*. Third type of data, *ideational data* does "describe the meanings, motives, definitions and typifications themselves". [10]

In this paper, *member* is used as a synonym for agent, user, actor, and (human) entity. Connections are called *relations* or *relationships.* In 1930's, Jacob Moreno introduced a concept of *sociogram* as means to visually represent "social configurations" (see [10]). We use the concept to refer to a logical description of a social network that is processed from the raw data in order to be visualised. By *community visualisation*, we refer to visualisations of the social networks in a community.

II. Background and related work

Both social network analysis (SNA) and social network visualisation have a long history in sociological research. From the originally hand-drawn images to the current interactive visualisations created with and running in computers, the social networks have been visualised in particular with diagrams, or sociograms, based on points representing social actors and lines representing the connections between actors. With the visualisations, the researchers have been able to gain and share insight on social networks. [11]

Traditionally, SNA has been applied by expert users and researchers when analysing a particular real-life social phenomena. As the popularity of social networking applications such as LinkedIn, Facebook, Orkut, and Friendster and Last.fm, Flickr, and other web communities has increased, the potential possibilities of end-user driven visualisation of the are becoming more evident to the service developers. There are various reported attempts to apply SNA in the context of social networking for research purposes (see e.g.[7], [12]).

Exploring the social network in the state-of-the-art social networking services is, in most cases, only possible through a linear list of friends presented as a part of a member profile. This makes it difficult to grasp the overall insight of the social network (cf. [9]). To support the exploration of social network in *Friendster*, a popular social networking service, Heer and boyd designed and built a social network visualisation tool *Vizster*. Vizster follows the tradition of sociograms by using nodes to represent the Friendster members. The nodes are interlinked by the articulated friendships between members with lines representing the relationships.

The member profiles play a significant role in a social networking service. On basis of their research on Friendster, boyd and Heer state that people use their profiles as means of creating and interpreting the context of online performance [13]. Heer and boyd point out that in Friendster, the existing profiles act as norms for new members defining their profile and that users tend to refine their tastes according to the ones of their friends [9].

III. Open Knowledge Space as a medium for social networking

A. *Social networking in online communities*

User profiles and other common social networking features are implemented into existing Wed services in a fashion sometimes resembling the enthusiasm related to the gold rush. Despite the obvious hype, the social networking feature come with distinct advantages to the members of a community.

The member profiles are the core of social networking. The profiles include attribute data of a member such as name, nickname, image, contact information, online accounts, interests and preferences. In addition to the attribute data, the members are often able to define relationships to other members of the community. The relationships are, in many services including Last.fm, Friendster, and Orkut, referred as "friendship" or sometimes simply "connections" like e.g. in LinkedIn.

In addition to the explicitly stated member profiles, the recorded member actions and the actual content of the service can be used to mine weak links between members. In Last.fm, for example, the music listened by a member is tracked and mined in order to find member-to-member, artist-to-artist, and member-to-artist relationships and create recommendations for the members [14]. In addition, the members are able to formally associate their published writings with artists, thus creating more annotated relations into the contents of Last.fm.

B. *OKS as a social networking platform*

Open Knowledge Space can be used as a social networking platform for researchers and other actors interested in the research of open digital ecosystems. The members are, already, able to define their profiles. The possibility to track

OKS users and mine the contents of OKS in order to induce new facts on users and new connections between users is appealing but needs further analysis and research. For example, on basis of the fact that user is mostly reading and editing wiki pages in two categories, **Visualisation** and **SocialNetworkAnalysis**, the members could be offered the possibility to easily define the two areas as interests into the profile (cf. [15]).

In Open Knowledge Space, currently, the member profiles are managed with *FactsAbout cards,* simple wiki pages following a specified structure. A page template is provided for the members to input their information. The OKS member profile include name, image, email, organisation, interests, the OPAALS tasks that the member is participating in, a free description, and links to homepage, blog and publications in systems external to OKS. The values are either free text or URIs pointing to wiki pages or other Web resources. Fig. 1 shows an example of a personal wiki page including the FactsAbout card.

Figure1. Example of a personal wiki page including the FactsAbout card

The FactsAbout card carrying the profile information of a OKS member is on the right side of the Fig. 1. The profile attributes are key-value pair, multiple values can be defined as needed. Music listening habits and other, more personal content, are not included in the profile data although it could be meaningful information to some of the OKS members.

C. Expressing researcher profiles in OKS

Friend of a Friend (FOAF, see [16]) is, along *RDF Site Summary* (RSS)[1], one of the first Semantic Web applications gaining wider popularity – largely due to the fact that both are in the core of the Web 2.0 phenomena. In short, FOAF is a vocabulary or a simple ontology for representing profile data including personal information, contact details, online accounts, publications, and interests. In addition, since FOAF enables the expression of relationships between individuals with the **foaf:knows** property, it can be used to represent a social network as a whole.

FOAF is an application of *Resource Description Framework* (RDF), the default mechanism for publishing metadata in *Semantic Web*. Semantic Web is an initiative of *World Wide Web Consortium* (W3C) aiming for a Web of machine interpretable meanings [17]. As RDF statements form logically a graph, the analogy between sociogram and FOAF might seems quite straightforward. However, as the example visualisations will show, several data processing steps are needed to transform the initial FOAF description to the final sociogram.

IV. Early examples of community visualisations

The first community visualisations prototypes of OKS are based on wiki data. The two prototypes to be introduced are *Community of Interest* and *Actor Network*, first visualising the relationships between OKS members on basis of their articulated interests and the latter presenting relationships between OKS members based on their recorded wiki editing behaviour. For now, the sociograms are created with ad hoc methods. In the future, more sophisticated SNA methods will be applied as needed (see e.g. [18], [19]). The two visualisations are introduced and discussed in the following chapters. As both of the prototypes use a tailored version of Vizster as a visualisation player, the fundamentals of Vizster are introduced first.

A. Vizster fundamentals

Vizster is designed to support the free exploration of a social network in a playful manner (see [9]). The playfulness is supported by the highly interactive interface. Originally developed to support the exploration of the the data collected from a social networking service Friendster, Vizster is able to represent both the structure of the social network and the details of individual user profiles, or in SNA terms, the sociogram including both the attribute data and relationships.

The social network is visualised from an egocentric point-of-view. Heer and boyd state that instead of following the Shneiderman visualisation mantra, the design philosophy of Vizster is “start with what you know, then grow” [9]. As the connections between the members are explicitly articulated by the members themselves, the immediate friends of a member act as landmarks helping the user to navigate and situate themselves into the visualisation. Nodes representing Friendster members display the name and a representative image of the member. Once a member node is selected, the profile information is presented in the visualisation. [9]

Vizster uses a spring-embedded layout algorithm where each node repel each other and the edges act as springs. The tension of the springs is adjusted according the connectivity

1 Other names for RSS include Really Simple Syndication and Rich Site Summary. In this article, RSS refers to syndication in general so also Atom is logically included in the list.

between nodes: the less connected the node is, the higher tension is given to the node thus forcing loosely connected nodes to form sub-communities or *cliques*. The possibilities to interact with Vizster include extending and collapsing the nodes by double-clicking a node, highlighting the nodes representing the friends and friends-of-friends of a member by pointing a member node, panning and zooming the visualisation, and a keyword search that can be used either by entering free text or by clicking the profile details. The X-Ray mode can be used to classify the profiles according their attribute data. Finally, a community highlighting mechanism can be used for automatic determination of the community structure. [9]

During informal experiments of the visualisation, the users were found to appreciate the playful aspects of the visualisation [9], [3]. The playfulness has been noticed also during the first experiments with the OPAALS visualisations.

To support a tight integration of Vizster to various OKS tools and views, a deep linking mechanism was implemented into Vizster. As the current OKS tools are Web-based, the Vizster is delivered as a full-featured application, instead of an Applet, by using Java Web Start launch technology (see http://java.sun.com/products/javawebstart/). The node that is selected by default can be specified as a parameter during the launch. This enables the creation of context-sensitive hyperlinks and special kind of bookmarks able to dynamically parametrise the bookmark, *bookmarklets*, for launching the visualisation directly into a specific state with a regular Web browser, assuming Java is supported by the environment in use. The up-right part of Fig. 1 shows a set of links placed on a wiki page enabling the launch of community visualisation with a particular OKS member selected.

A drop-down menu enabling the selection of a particular OKS member present in the visualisation data was added to Vizster on basis of the preliminary experiments with the visualisation. This is possible since there are, currently, tens of members in OKS. By design, Vizster is able to handle large networks of members so new ways to select a particular member have to be considered as more OKS members appear.

B. Community of Interest

Community of Interest is an example of a sociogram that is based on inferred rather than articulated relationships between members. The relationships are formed through shared interest between members based on the profile data. The connections between actors are, for now, processed in a simplistic manner: a relationship between two members is formed if they share at least one interest. The interests are defined in the FactsAbout cards either as free text or by adding hyperlinks to wiki pages defining or containing information on the subjects that the member is interested in. Fig. 2 includes an example view of Community of Interest and Fig. 3 shows an example of the profile display. In Vizster, the two displays are placed next to each other.

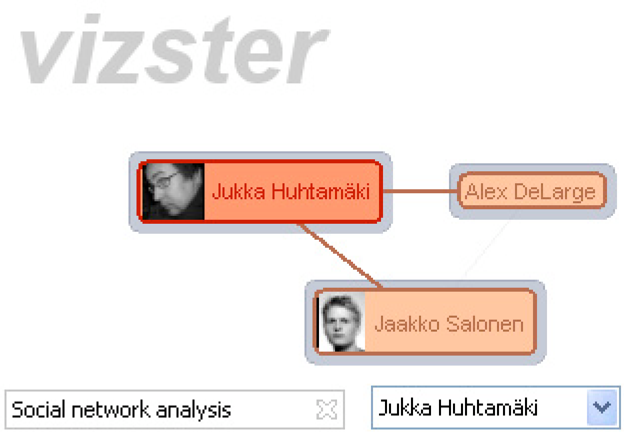


Figure2. Example view of Community of Interest in Vizster

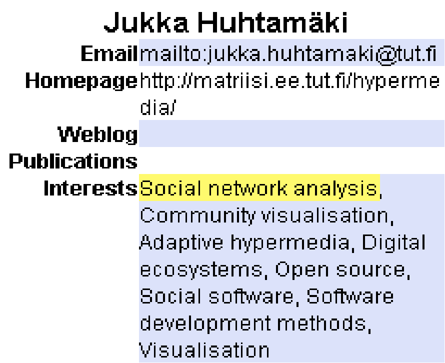


Figure3. Example view of the profile display in Vizster

On basis of Fig. 2, we can conclude that since all three OKS members are interconnected, they share at least one interest. The details of a profile can be used to find out the interests of each member in detail (see Fig. 3). Another approach is to use the in-built search mechanism of Vizster: in Fig. 2, the state of the visualisation shows that all three profiles include the text “Social network analysis” that the user has entered. The same text is highlighted in the profile display in Fig. 3.

A major problem with the current relationship-creation method is that it is *crisp* (as opposed to *fuzzy*): a connection between two members is established only if the two free texts or hyperlink URIs compared are equal. The means to improve the current mechanism include, among all, the following:

First, a *tagging* mechanism following the conventions of *folksonomies* might catalyse the evolution of a shared vocabulary for defining the interests. A folksonomy is a vocabulary that is not controlled by third parties, but is rather defined by the members of a community (see e.g. [20]). Tagging refers to a functionality enabling adding the individual folksonomy concepts into the contents of a Web community (cf. [21]), in this case to the profile of a member.

Second, an *interpretation ontology* specifying the

connections between individual concepts can be developed. Nykänen defines an interpretation ontology as a special case of interpretation logics that, in general, enable the shared use of incompatible logics [22]. Examples of the types of connections between concepts include is-a and synonymous relationships.

Third, a more organic integration of the visualisations and the OKS content management tools would - together with defined processes, practices and guidelines - enable easier adjustment of the interest descriptions, thus promoting the use of shared concepts. The latter approach is already partly supported by a scenario description available for the users of OKS. The scenario describes a usage pattern of the visualisation in defining the interests in a uniform manner.

C. Actor Network

Actor Network is the first attempt to visualise a social network of OKS members on basis of their recorded actions instead of data explicitly created by the members. For now, the Actor Network visualisation is based purely on the editing data collected from the wiki. In the future, more OKS tools will be connected to the visualisation. Fig. 4 shows an example of Actor Network in action.

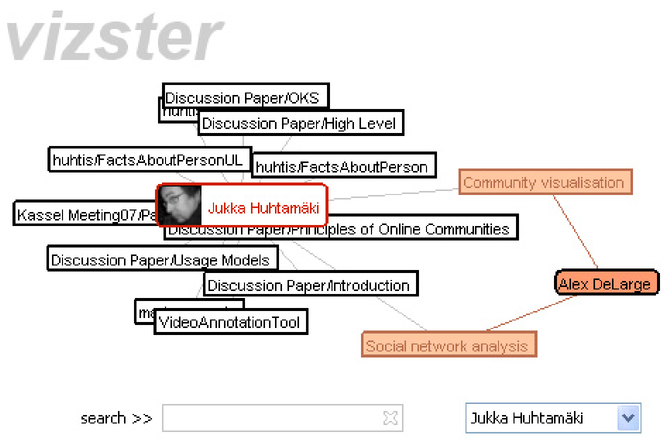


Figure4. Example view of Actor Network in Vizster

In Fig. 4, there are two member profiles present in Actor Network sociogram, namely *Jukka* and *Alex*. As seen, Alex has not yet added his picture into his FactsAbout card. From the visualisation, we can see that there are two pages, *Community visualisation* and *Social network analysis*, jointly edited by the two OKS member. In addition, Jukka has edited a few other pages. The node representing Alex is collapsed, so he might have edited some other pages as well. In the live visualisation, a profile display similar to Fig. 3 appears on the right side of the sociogram.

Unlike in the original version of Vizster where only the member profiles are present, in Actor Network also the targets of action are present as nodes in the visualisation. This makes the relational data visible and supports finding new targets of possible interest. A connection between a member and a target of action, currently a wiki page, is established if the actor has edited the page at least once. The approach is, of course, again quite naïve but nevertheless does enable the implementation of the first prototype of the visualisation for comments and critique.

As the name of the visualisation suggest, we are aiming for visualising the OKS actors and targets-of-action in general instead of wiki pages and editors. In order to do this, we have to integrate more data sources into the visualisation data processing pipeline and, in addition, generalise the data processing methods and the visualisation player.

D. Design and implementation of the visualisations

The sociogram data for the visualisations is processed with *VizPipe*, a general framework for building *data processing pipelines* with arbitrary components, both open source or proprietary. VizPipe is currently being developed in the Hypermedia Laboratory of Tampere University of Technology (see [23]). Data processing pipelines are chains of parametrised data transformations refining raw data into a visualisation (see [24], [25]). Although being technically able to handle various data formats also internally, the VizPipe pipelines deliver data on Extensible Markup Language (XML) format in order to gain high interoperability. VizPipe already includes components for fetching data from different systems in various formats, data aggregators supporting both XML and RDF, transformation engines capable of mapping data between different XML vocabularies, engine for querying RDF data with SPARQL (see [26]) *et cetera*. New components are introduced to the environment as needed.

Both Community of Interest and Actor Network are based on FOAF data representing the OKS member profiles. A pipeline for creating the FOAF data is structured as follows. First, FactsAbout pages are collected from the wiki, in *DocBook* XML (http://www.docbook.org/) format, with a spider component utilising the export functionality of *MoinMoin* (see http://moinmoin.wikiwikiweb.de/), the wiki engine currently in use in OKS. Next, the individual FactsAbout pages are aggregated into a single XML document and profile data are distilled from the pages with a simple transformation. The result, a document following a general vocabulary for facts is, finally, transformed into a FOAF document. In addition, a separate pipeline collects the facial images of OKS members.

The sociogram data in Community of Interest is created from FOAF data with a SPARQL query that selects all the member-member pairs sharing at least one interest. A parallel SPARQL query picks all the member attributes needed in the visualisation. On basis of the result sets of the two queries, a general description of the Community of Interest is created. Finally, as Vizster uses a dialect of Graph Markup Language (GML), the sociogram data is transformed into a GML document.

For the Actor Network, the FOAF data are aggregated with the metadata collected from OKS wiki, including the page-specific editing information. The aggregated RDF data are then queried to find relations between the wiki pages and the members of OKS. The details of the pages and the members

are collected with two additional queries. The three result sets are used to, first, create a general description of the actor network and, finally, a GML representation of the sociogram.

The data processing procedures in the two pipelines show that although the raw data encoded in RDF and the resulting sociograms are both logically graphs, a lot of processing is needed to create a detailed end result. However, there are tools to visualise also the raw data that can be used to complement the visualisation in order to gain insight and make the process more transparent.

V. Discussion

For now, the data analysis in the two cases presented is being performed with query and transformation languages that enable simple low-level operations on basis of the data. The mathematical methods of social network analysis, graph theory, and data mining introduce expressive means to create new information into the member profiles. As a general matrix algebra tool is already integrated into the visualisation pipeline framework, we are now planning to introduce some selected social network analysis algorithms into the pipelines, thus making the creation of the sociograms more sophisticated. Also component enabling inference based on graphs, in particular ontologies, will be integrated into the environment. Nevertheless, we see that the visualisation end-user should be given means to alter both the data and the sociogram-creation algorithm. In addition, we see that the data analysis process should be as transparent to the end-user as possible (cf. [27]).

By design, FOAF is intended to enable people publishing information for machines to process: "If people publish information in the FOAF document format, machines will be able to make use of that information." [16]. This implies that the member profile information is explicitly defined by the people, not inferred by a machine. Despite the fact that FOAF, as RDF in general, is intended to include precise information that can be easily processed with e.g. query languages, in practice the FOAF information can be quite arbitrary.

To generalise, there are two approaches to apply FOAF in implementing social networking. The first option is to encode explicit profiles with FOAF directly and utilise high-end social network analysis and other data analysis methods in creating the sociograms. Alternatively, FOAF can be used to encode the results of data analysis based on explicit profile data and recorded user action, thus enabling a more direct creation of sociograms on basis of the FOAF description. At this point, we are leaning towards the latter approach but more work is needed before conclusions.

Currently, wiki data covers the majority of OKS data available for the visualisations. As new data sources emerge, integration work is needed to connect the data processing pipelines to the raw data. The two Semantic Web killer applications, RSS and FOAF, can be used jointly to make the interface between visualisations and OKS more generic. As both vocabularies are RDF based, they can be easily extended if new kind of data needs to be imported into the visualisations.

After the problems of technical nature with added transparency and parametrisation of data analysis are solved, we anticipate the emergence of issues related to e.g. the publicity of the profile information and the data recording the actions of a member. The improved visualisations we and other are aiming for, capable of crafting more expressive views to the contents of a digital ecosystem, together with the expanding nature of such an ecosystem come together with a problem named as "dilemma of collapsed contexts and unknown audiences" by boyd and Heer [13]. This means, essentially, that the members of the community evaluate the context of their actions of basis of the prevailing situation and act accordingly. As the ecosystem evolves, the context potentially enlarge and converge. The in-built version control system makes the issue of context collapsing even more meaningful.

VI. Conclusions and future directions

After the initial implementation and integration work, the first community visualisation are now in use in Open Knowledge Space. Vizster, the visualisation client currently used, provides the visualisation user with powerful means to navigate the pre-designed sociograms. However, since the user is not able to alter the data set, the options to manipulate the visualisations are limited. As the development of visualisation pipeline environment VizPipe proceeds, appropriate means to parametrise the data and the analysis can be offered to the user. We see that this both adds to the usefulness of the visualisation while at the same time the visualisation more transparent.

The possibility of community visualisations acting as means of pointing OKS members into rabbit holes leading to new, unexpected informations landscapes, thus supporting sharing information over disciplines, problem-solving, and innovation is a very high-level and abstract goal but, at the same time, something that the visualisations can enable at best. More work is needed before this objective can be understood and defined in a concrete level.

The idea of enabling community members to write themselves into the social network with the help of visualisations is also appealing. In order to achieve this, however, the integration of the visualisations and the content management applications has to be organic and, more importantly, the processes must be clear to the actors. In addition, the causalities between the actions of a member and the resulting visualisations have to be understandable, that is, the data processing and visualisation methods and algorithms have to be as transparent and intuitive as possible.

More work is needed before the issues related to evolving contexts and the publicity of social network data can be described in more detail.

Acknowledgment

The author would like to thank Jaakko Salonen for his invaluable comments and remarks on the article and Jaakko, Markus Mannio, Matti Haapaniemi, Ossi Nykänen and Antti Kortemaa and other members of OPAALS for their feedback during the visualisation development. This work has been

supported by the European Comission (IST network of excellence project OPAALS of the sixth framework program, contract number: FP6-034824).

References

1] C. Ware, Information Visualization, Second Edition: Perception for Design. San Francisco, CA, USA: Elsevier, 2004.

2] B. Shneiderman, "The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations," IEEE Visual Languages, pp. 336-343, 1996.

3] J. Heer, "Socializing Visualization," Social Visualization Workshop, CHI 2006, Montreal, Canada, April 2006.

4] wikipedia.org, “Rabbit hole,” 2007, http://en.wikipedia.org/wiki/Rabbit_hole.

5] J. McGonigal, "Down the rabbit hole," presentation at 050505: Zoning and Grinding, New Media Conference, Berkeley, CA, USA, May 2005.

6] T. O'Reilly, "What Is Web 2.0: Design Patterns and Business Models for the Next Generation of Software," oreillynet.com, September 2005.

7] P. Mika, "Social Networks and the Semantic Web," Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence, pp. 285-291, 2004.

8] L. Ding, L. Zhou, T. Finin and A. Joshi, "How the Semantic Web is Being Used: An Analysis of FOAF Documents," Proceedings of the 38th Annual Hawaii International Conference on System Sciences, 2005, pp. 113c.

9] J. Heer and d. boyd, "Vizster: Visualizing Online Social Networks," Proceedings of the 2005 IEEE Symposium on Information Visualization (INFOVIS'05), Minneapolis, MN, USA, October 23-25, 2005, pp. 33-40.

10] J. Scott, Social Network Analysis: A Handbook. London: SAGE Publications Ltd, 1991.

11] L.C. Freeman, "Visualizing Social Networks," Journal of Social Structure, vol. 1, pp. [np], 2000.

12] S. Staab et al., "Social networks applied," Intelligent Systems, vol. 20, pp. 80-93, 2005.

13] d. boyd and J. Heer, "Profiles as Conversation: Networked Identity Performance on Friendster ," Proceedings of the Hawaii International Conference on System Sciences (HICSS-39), Persistent Conversation Track, Kauai, HI, USA, January 4 - 7, 2006.

14] audioscrobbler.net, "Audioscrobbler - The Music Technology Playground," 2007, http://audioscrobbler.net/.

15] P. Brusilovsky, "Methods and Techniques of Adaptive Hypermedia," User Modeling and User-Adapted Interaction, vol. 6, pp. 87-129, 1996.

16] D. Brickley and L. Miller, "FOAF Vocabulary Specification 0.9," May 2007, http://xmlns.com/foaf/spec/

17] I. Herman, "W3C Semantic Web Activity," 2007, http://www.w3.org/2001/sw/.

18] S. Wasserman and K. Faust, Social Network Analysis: Methods and Applications. Cambridge University Press, 1994.

19] P.J. Carrington, J. Scott, and S. Wasserman, Models and Methods in Social Network Analysis. Cambridge University Press, 2005.

20] T. Vander Wal, "Folksonomy Definition and Wikipedia," November 2005, http://www.vanderwal.net/random/entrysel.php?blog=1750.

21] C. Marlow, M. Naaman, d. boyd, M. Davis, "HT06, tagging paper, taxonomy, Flickr, academic article, to read," Proceedings of the Seventeenth Conference on Hypertext and Hypermedia, Odense, Denmark, August 23-25, 2006, pp. 31-40.

22] O. Nykänen, "Interpretation Logics," 1st OPAALS Conference, Rome, November 26-27, 2007, unpublished.

23] O. Nykänen, M. Mannio, J. Huhtamäki, J. Salonen, "A socio-technical framework for visualising an open knowledge space," Proceedings of WWW/Internet 2007, October 2007, in press.

24] M. Kreuseler and H. Schumann, "A Flexible Approach for Visual Data Mining," IEEE Transactions on Visualization and Computer Graphics, vol. 8, pp. 39-51, 2002.

25] E.H. Chi, "A Taxonomy of Visualization Techniques Using the Data State Reference Model," Proceedings of the IEEE Symposium on Information Vizualization 2000, pp. 69, 2000.

26] E. Prud'hommeaux and A. Seaborne, “SPARQL Query Language for RDF,” 2007, http://www.w3.org/TR/rdf-sparql-query/.

27] K. Swearingen and R. Sinha, "Beyond algorithms: An HCI perspective on recommender systems," Procedings of the SIGIR Workshop on Recommender Systems, New Orleans, LA - USA, September 13th, 2001, September 2001.

An Evolutionary Multidimensional Trust Model for Digital Ecosystems

OPAALS Conference 2007

Mihaela Ion, Luigi Telesca
CREATE-NET
Trento, Italy
{mihaela.ion, luigi.telesca}@create-net.org

Jimmy McGibney, Dmitri Botvich
Waterford Institute of Technology
Waterford, Ireland
{jmcgibney, dbotvich}@tssg.org

***Abstract*—In a Digital Ecosystem (DE), diverse actors, like SMEs, public organizations, and user, and their products and services interact in an open decentralized infrastructure. They evolve like organisms in a biological ecosystem and provide complex solutions and a composition of solutions in a dynamic way. In order for different actors to cooperate and create relations, and stable or unstable coalitions, trust or different levels of trust need to exist between the actors, and without it, no collaboration would be possible. Hence, providing a reliable trust framework that allows establishing trust relations between different DE partners is essential for the growth and sustainability of the DE as a whole. This paper takes inspiration from different fields related to the DE philosophy like sociology, biology, economics and especially computer science, to propose a novel multidisciplinary trust model for DEs.**

Keywords-digital ecosystem, distributed trust, identity management, multidisciplinary approach.

I. Introduction

Digital Ecosystems (DE) have emerged as the new multidisciplinary paradigm for the evolution of Networked Organizations. Different definitions of Digital Ecosystems are available. This is mostly due to the complexity of the concept and to the disciplinary domain of reference. Anyway, although we are simplifying the concept, we can define the Digital Ecosystems as an open decentralized information infrastructure where different networked agents, such as enterprises (especially SMEs), intermediate actors, public bodies and end users, cooperate and compete enabling the creation of new complex structures. In Digital Ecosystems, the actors, their products and services can be seen like different organisms and species that are able to evolve dynamically. These organisms or species, through incessant interactions, alliances, adaptation and composition could experience new evolutionary paths in order to better adapt and survive to changing market conditions. In this context organisms are able to migrate from one ecosystem to another, can experience mutations or even die because of adverse conditions and digital selection.

Establishment of mutual trust is an essential requirement for any human interaction and collaboration. The same holds inside a DE: entities need to trust each other in order to build stable business relations and to make the DE sustainable and ensure its growth. Providing means for securely identifying entities in the DE and assigning to each a level of trust is a necessary requirement for the sustainability of the DE.

The dynamic nature of Digital Ecosystems poses many challenges for identity management and reputation-based trust systems. Existing technologies and models do not fully integrate the needs of SMEs and are not suitable for DEs. In this paper, we propose a new multidisciplinary model for DEs based on current security technologies and reputation mechanisms. Though we take a computer science approach, in order to solve concrete issues related to the open distributed platform of the DE, we also rely on concepts and methods from ecology, economy and sociology. As a result, we propose a novel multidisciplinary evolutionary trust model for the DE.

The model is based on a Distributed Identity Management model which enables creating a reputation framework by providing ways for securely identifying entities. The reputation framework has two main components: a Peer-to-peer Reputation model and a decentralized Trusted Rating Agencies model. DEs evolve in time in order to respond to changing conditions. To better accommodate the nature of DEs, the model uses evolutionary trust, a novel research concept that reflects the constantly evolving social relations.

II. Trust

In the real world, as in the digital one, trust is one of the key factors that permit the growth of social and business networks. As identified by Jenny Preece D, only when there is trust between people the relations can flourish, without trust there is no cooperation and therefore no society. Trust in online communities is even more important, than in real world, due to the difficulty of a physical interaction and to the lack of unintentional emotional states that could be expressed through body language and implicit behavior. Trust is therefore central and is also a basic requirement for a functioning DE. The concept of trust has been researched in different disciplines and due to its multidisciplinary dimensions it has been open to multiple interpretations. In this section we introduce different aspects of the notion and models of trust and we try to understand how trust can be measured and used in a computational manner to make DEs work.

A. The Meanings of Trust

The social concept of trust is very complex and sophisticated, perhaps deceptively so. Trust is closely related to many other social concepts such as belief, confidence, dependence, risk, motivation, intention, competence and reliability D. It is also interwoven with the areas of accountability and identity.

Trust management has become a fairly well-studied area in recent years, and several recent surveys summarise the state of the art DD. Much of the work on trust in the computer science domain attempts to provide computational measures of trust so that it can be used in making decisions related to electronic transactions, in a sense mimicking people's well-evolved forms of social interaction.

One of the difficulties of modelling trust computationally is that social trust is based on quite an intuitive and personalised subjective assessment. As trust is quite an overloaded term, most attempts to model trust start by defining what is meant by trust (at least for the purposes of that particular model). Thus there are currently several different definitions and interpretations of what trust means and how it can be used. Here we describe some contributions and definitions about trust coming from Social Sciences, Economics and Biology. This will help us in order to better understand the complexity of the topic and to define our multidisciplinary approach toward trust in DE research.

Trust in Social Sciences: Trust refers to a unidirectional relationship between a trustor and a trustee. A trustor is an autonomous entity that is capable of making an assessment. A trustee can be anything. In social sciences trust has several definitions, including trust as a measure of something that can be used in decision-making. Gambetta defines trust as a probabilistic measure indicating confidence in a certain type of behaviour, used as a basis for deciding whether to rely on another entity D. *"To trust" (or not) may be taken to mean the decision itself.*

B. Trust in Digital Ecosystems

Different layers and types of trust have been studied and identified in the DE research. Nachira (2002) D suggested the following layers of trust in DEs:

- *Trust in services and in technological solutions* in terms of confidence about platform security and reliability.
- *Trust in business activities*: agreement towards practices and procedures for specific sectors and local contexts.
- *Trust in knowledge* in terms of symmetric vs. asymmetric access to information.

In the DBE project (Deliverable 32.2) D, three types of trust in DE have been identified:

- *Trust type X* refers to *trust in the system*. The trust companies and users have in the technical architecture and services of the system determines them either to join or not the system.
- *Trust type Y* refers to the *expectations* established members have about joining users.
- *Trust type Z* refers to the *trust relationships between participants* of the system. A higher mutual trust results in a higher number of transactions and collaborations between participants.

Taking into consideration all the definitions and properties of trust that have been identified in different domains related with the DE philosophy, we will define in the following trust research challenges that we want to tackle in *OPAALS*.

C. Trust in OPAALS

Because of the constantly evolving and dynamic nature of relations in a DE, our goal is to model an adaptive multidimensional reputation-based trust. Trust will be primarily established based on recommendations expressed by users not only about other users, but also about data/knowledge, services and infrastructure. Hence, we tackle *multiple levels* of trust. Moreover, in a DE, users organize themselves in *social networks* and trust values are, therefore, local and relative to each user and his neighbourhood or social network. Because users interact in *different contexts* (e.g. automotive, construction, IT), the model allows expressing trust recommendations relative to a user-defined keyword (folksonomy). Contexts could be complex an expressed using a taxonomy. By using different contexts to express the trust level in a certain entity, we model multidimensional values for trust. For example, a service could be rated differently or availability, response time, memory usage, result accuracy etc.

III. Identity, reputation and their impact on trust

A Digital Ecosystem is a distributed environment in which there is no big-brother or central authority. A DE functions as a peer-to-peer network in which all peers are equal and free to join or leave the system, interact with each other mainly for business transactions, and form stable or unstable coalitions. In such a system, peers behave like autonomous agents. Each agent is free to reason and decide about the trustworthiness of other agents and decide with whom to transact.

Using a reputation-based trust model, agents can cooperate with each other in order to assess the trustworthiness of unknown peers based on the experiences of others. Thus, the reputation a peer has in a Digital Ecosystem is very important. From an economical point of view, reputation is an asset. Reputable peers are trusted for transactions which helps them make more profit. Sellers with a high reputation can increase prices while sellers with a bad reputation will not be able to attract any buyers and will eventually be eliminated form the system. From a sociological perspective, reputation helps building mutual trust between peers which is the basis for every interaction and cooperation such as a single transaction or a large coalition. From a computer science point of view, assigning reputation values to peers increases the security of the system. In this way, peers can prevent defected transactions, malicious attacks on peers or on the system as a whole.

In order to assign reputation values to entities, each entity that takes part in a DE needs to have a digital identity which is used when interacting with other entities. Peers do not interact with each other directly, but online through the system which increases mistrust. Dellarocas D shows that the more separated in time and space the entities are, the greater the risks. In Digital Ecosystems, identity management becomes a bottleneck especially when entities from different administrative domains interact with each other. Identity management is an important part of any reputation system, either centralized or decentralized.

The last aspect we have to consider is the evolutionary nature of the ecosystem. Relationships evolve during the lifetime of an ecosystem and peers change the way they interact with each other. Peers can only evolve in time and change their behaviour if a learning mechanism is used. In computing the reputation of a peer, information from outside the system can be additionally used like for example existing trust relationships and institutional or inter-ecosystem trust.

IV. Model overview

We provide a reputation framework that allows assessing the trustworthiness of agents based on their past behaviour and decide whether to interact or not.

The framework should provide support for cross-domain reputation assessment. The framework relies on a Distributed Identity Management Model DD for authenticating users and relating them with their past actions. There are two main components: a peer-to-peer reputation system and a Rating Agencies.

A. Identity Management

A prerequisite to any trust system is consistent identities which can only be provided by a reliable identity management system. Proper user authentication will be provided by a robust Distributed Identity Management Model based on widely used standards

A DE is a dynamic and unstable environment composed of diverse institutions which sometimes compete and other times collaborate. As a result, stable and unstable federations are created which poses many challenges to managing identities. First of all, organizations use divers types of certificates and identity technologies (e.g. X.509[1], SPKI, Kerberos and OpenID[2]) which are not always compatible with each other. Secondly, users often need to access applications, services or a composition of services located on different administrative domains. Finally, because of the dynamic nature of the environment, federating and sharing of identities becomes a complex task.

WS-Policy[3], WS-Trust[4] and WS-Federation[5] offer identity federation solutions, but these are pure federating approaches are hence viable only when there is a stable federation. In DEs, federation does not scale up because of the unstable and ad-hoc coalitions. Moreover, these models are heavy to implement by SMEs, being more suitable for large enterprises. What DEs need is an identity management model able to provide:

- exchanging identity information between companies independent of the standards they use, and
- sharing user identity between different domains, either federated or with no direct trust.

Additionally, the model needs to be targeted, easy to understand and straightforward to implement by SMEs.

We prose a model that satisfies these requirements and automates the process of identification between DE partners.

The key entities in the model are:

- *User*: any entity in the network (peer or web browser user, institution or person)
- *Service Provider (SP):* any entity that has one or more services or resources available to other entities.
- *Credential Provider (CP):* any entity that is able to provide digitally signed credentials to other entities.

The model considers all users to be equal and assumes there is no hierarchy of DEs. Each peer can be a CP (thus issue identification and authorization certificates to other peers) and each peer can be a SP (thus provide services and resources to other peers). A peer can be both a CP and a SP at the same time. Each user can freely choose what kind of certificates to accept. For that matter, each user and SP has a list of trusted CP and only certificates coming from a trusted CP are accepted. CPs can establish trust relations between them and accept certificates coming from a trusted CP. Each CP has a list of accepted security tokens (i.e. issued by a trusted CP) and issues certificates based on:

- secure tokens issued by the provider itself or
- trusted secure tokens (from CPs with whom it has trust relationships) or
- user registration information.

Thus users obtain a number of security tokens from different CPs that can belong to different DEs. The model also allows users to import certificates obtained outside the DE. This approach has several advantages. First of all, for legal and economic reasons, sometimes companies are required to use a certain kind of certificates (e.g. issued by a certain certification authority which is outside the system). Secondly, it allows building trust and networks of trust in the system by using trusted certificates and relations from outside the system. Moreover, it allows the system to take off fast and newcomers are accepted in transactions more easily. The lake of trust relations between peers can be a bottle neck to the growth of a new peer-to-peer system. By using this approach, our model overcomes this problem.

---

[1] Public-key and attribute certificate frameworks, 2005. ITU-T Recommendation X.509:2005 | ISO/IEC 9594-8:2005.

[2] http://openid.net/

[3] http://www-106.ibm.com/developerworks/library/specification/wspolfram

[4] http://www-106.ibm.com/developerworks/library/specification/ws-trust

[5] http://www-106.ibm.com/developerworks/webservices/library/wsfed

The model is based on the new SAML[6] (v2.0) standard for bridging different standards that companies might use (e.g. X.509, SPKI, Kerberos tickets, username/password etc) and for exchanging identification and authorization information independent of the technologies used by different SMEs in the DE. To cope with a wide range of identity mechanisms we require for every CP to support by default SAML v2.0. This means that each SP adopts the identity standard best suiting its needs but its related CPs supports SAML. This means that any SME could preserve its existing identity management infrastructure but should enhance its trusted CP with the ability to understand SAML. Furthermore, each CP must be able to issue SAML assertions derived (transformed) from any of the standards the CP supports.

Every user has an identity profile distributed in the peer-to-peer network which can be accessed by the user from any peer in the system using unique username & password chosen at registration. This approach accommodates users which do not own a computer and provides ease of use of the system. A user profile contains a list of all available certificates and security tokens the user has. The profile is encrypted with a long master password known only by the user and never stored in the system. To make the profile available at all times and to provide a secure storage, we replicate the profile on trusted nodes. Each DE has a number of trusted nodes determined by the peer-to-peer reputation system.

The model is based on transformation of security tokens from one CP to another and is further described in Figure 1.

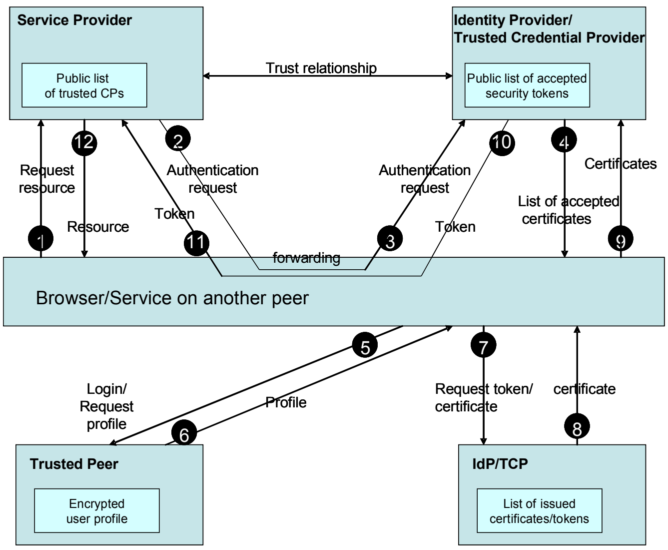


Figure 1: Basic Identity Management Model

When a user tries to access a service (1), the SP redirects it to the trusted CP (2-3). If the user already has a token from this CP, then the user can simply authenticate and be redirect to the SP. If the user is not known by the CP, the CP will provide a list of accepted security tokens (4). The user has his profile replicated on trusted peers and can login with username and password (5) and download his profile and decrypted on a secure memory with the master password (6). The user can check if there is a match between the requested tokens and the tokens in the profile. If a match is found, the user requests a token from the CP who issued it (7-8). The user provides this token to the CP (9) who will issue a new certificate that is understood and accepted by the SP. The CP might need to make a translation, i.e. issue another type of certificate that is understood by the SP.

Users wish to protect their privacy. For that matter, each CP has the responsibility to provide proper pseudonimity to end users by using a pseudonym, either chosen by the CP or by the user. The pseudonym is then certified in a trusted secure token to a SP. In case of misbehaviour, the SP explicitly asks a CP to reveal user identity. So, each CP maintains a database mapping user's pseudonimity with user's real identity.

Once we have defined a proper identity system, reputation systems can be built on top of it.

B. *Peer-to-peer Reputation System*

For the purposes of this work, we adopt a two layer model for communications between peers. Peers can either interact for a transaction or to exchange trust information. For modelling purposes, each service usage interaction is a discrete event. A logically separate trust management layer handles threat notifications and other pertinent information.

We mimic social trust by setting a fuzzy trust level. Each different service can then make an appropriate decision based on this trust level – e.g. certain actions may be allowed and others not. In our system, we model trust as a vector. In the simplest case, at least if there is just one service, this can be viewed as a simple number in the range (0,1). Each peer may then maintain a local trust score relating to each other peer of which it is aware. If peer A's trust in peer B is 1, then peer A completely trusts peer B. A score of 0 indicates no trust. If trust is to be a probability measure, then the (0, 1) range is natural.

Trust representations

We will represent trust as a probabilistic measure in the range (0, 1). Zero means no trust or no information about trust and 1 means complete trust.

If we want to represent trust as a complex value such that it is more meaningful and conveys more information about the trustee, each dimension of trust will be represented as a probabilistic value from 0 to 1. For example, if agent A wants to express the trust it has about service S, it can use the following structure:

(A, S, availability, 0.9)
(A, S, response_time, 0.3)
(A, S, result_accuracy, 0.8)
(A, S, memory_usage, 0.2)
(A, S, security, 0.6)

Users or companies could be trusted in a certain context, but not in another one. For example, A can trust B about car fixing, but not about baby sitting. In order to express trust in a particular context, an agent can use the following structure: (*TrusterID, TrusteeID, Context, trust_value*). Trust_value can

[6] Security Assertion Markup Language (SAML), 2005. www.oasis-open.org/committees/security.

either be simple or multidimensional as showed above. For defining contexts, we will allow users to build a folksonomy by defining own keywords for expressing contexts.

Architectural overview

We propose the overlay of a distributed trust management infrastructure on top of the service delivery infrastructure (which may itself contain multiple layers).

Figure 2 illustrates the relationship between underlying services and this new infrastructure. With our proposed trust overlay architecture, a trust management layer operates separately from the mechanics of the transactions themselves. Two message passing interfaces are defined between the transaction layer and the trust management layer and another between the trust managers of individual peers. The interfaces are as follows (Figure 2):

(1) Experience reports: Transaction engine → Trust manager
(2) Trust recommendations: Trust manager ↔ Trust manager
(3) Policy updates: Trust manager → Transaction engine

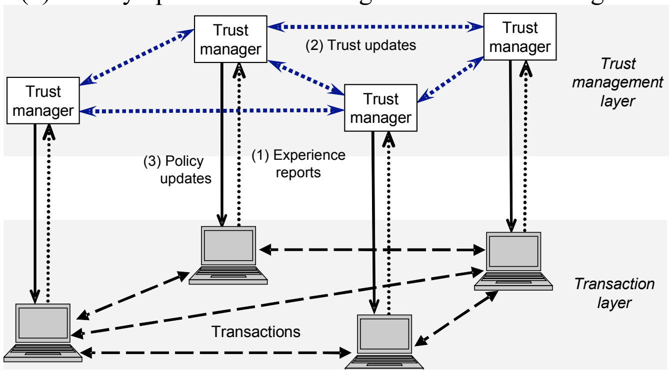


Figure 2: Trust overlay helps to secure transactions in a digital ecosystem.

Transactions are as normal, with the trust information just influencing protection mechanisms. The trust manager gathers transaction experience and uses this together with the experience of collaborators to inform its trust in other peers.

In the initial case, all trust values are set to a low value. Having initial values set to zero would prevent the so called *Sybil Attack* D, whereby attackers can take advantage of a default trust level by maintaining multiple identities. This is impractical though as we need to have some low-risk services enabled in order to get experience of other peers.

We then need to have a system to build up trust as peers gain experience of each other or learn of each other's reputation. In our system, trust can be updated in two ways:

1) *Direct experience:* On completion of a transaction between two peers, each peer updates its trust in the other based on a measure of satisfaction with the transaction.

2) *Reputation:* Peer A notifies other peers in its *neighbourhood* of the trust score that it has for peer B. This will change significantly following a security-related event.

How this neighbourhood is defined is significant. The neighbourhood of a peer is the set of peers with which it can communicate or with which it is willing to interact. The choice of neighbourhood peers is up to each individual peer to decide, and can be viewed as the peer's social network. In reality, choice of neighbours may depend on physical geography, network topology, frequency of contract or even trust level.

The main benefit of this system is in using these trust scores to tune security settings. Trusted peers can be dynamically provided with more privileges than untrusted peers. In our system, as mentioned, we model all interaction between peers in terms of services. Each peer then sets a threshold trust level for access to each service in which it participates. If the trust score of a peer decreases, for example due to detected suspicious activity by that peer, services available to that peer are reduced.

How this neighbourhood is defined is significant. The neighbourhood of a peer is the set of peers with which it can communicate or with which it is willing to interact. The choice of neighbourhood peers is up to each individual peer to decide, and can be viewed as the peer's social network. In reality, choice of neighbours may depend on physical geography, network topology, frequency of contract or even trust level.

The main benefit of this system is in using these trust scores to tune security settings. Trusted peers can be dynamically provided with more privileges than untrusted peers. In our system, as mentioned, we model all interaction between peers in terms of services. Each peer then sets a threshold trust level for access to each service in which it participates. If the trust score of a peer decreases, for example due to detected suspicious activity by that peer, services available to that peer are reduced.

Trust overlay protocol

This section describes a generalisation of a protocol already proposed in D to support trust-based email filtering, named TOPAS (Trust Overlay Protocol for Anti Spam). The TOPAS protocol allows for the collection of spam statistics to calculate trust, sharing this trust information between peers and feeding it back to mail hosts to allow them to more effectively filter spam.

(1) Experience report: Service host → Trust manager

The service host has an associated access controller. Each transaction attempt is processed by the access controller, with the result that it is either accepted or blocked. Accepted transactions are then monitored (e.g. by an IDS, or a spam filter if the service is email) and determined whether the transaction was benign or malicious. The result of this assessment of the transaction needs to be made available to the trust manager.

(2) Trust recommendation: Trust manager ↔ Trust manager

Peers collaborate to share trust information with one another. This is done by the trust managers, using the

primitives outlined here. Trust managers may issue recommendations spontaneously (for example on a usage alarm) or in response to a request from another peer. Note that a request may be issued by one trust manager to another, but a reply is not guaranteed. As mentioned earlier, message passing is asynchronous and the protocol requires no state information to be maintained by the corresponding entities.

(3) Policy update: Trust manager → Service host

The third part of this collaboration architecture is responsible for closing the loop. Direct experience is recorded by peers and shared among them. The result of this experience and collaboration is then used to inform the service host to allow it to operate more effectively. Specifically, the service host, on a transaction attempt peer *i* needs to be able to access the current trust information that its trust manager has on peer *i*. Although the service host may be able to use local storage to, for example, cache trust values, we do not place any such requirements on it. Thus we need the ability for the service host to request trust information from the trust manager and receive a timely reply.

Note on using this protocol. The functions specified at interface (2) above allow for either a "pull" or a "push" model for sharing trust information. Messaging is asynchronous for experience reports and recommendations.

"pull" model: The trust manager receives a *getTrust*, *getTrustResponseRequested*, or *getTrustAll* request for trust information about another peer, or all peers, and subsequently replies with a *trustReport* or *bulkTrustTeport* message. The sender of the request will need to use a timeout mechanism. In the case of a *getTrust* message, there is no onus on the recipient to reply. With *getTrustResponseRequested* and *getTrustAll*, the recipient should issue a reply even if this contains no trust information.

"push" model: The trust manager may be configured to spontaneously issue trust updates, either to other peers or to its local service host. This is typically for reasons of performance and efficiency. It is wasteful for peers to repeatedly poll each other unless there is useful new information. In the "push" case, each peer decides when and to whom to issue trust advertisements.

Model description

In this section, we present a model for using trust information to enhance security through collaboration in a rich multi-service decentralised environment. One of the problems of modelling decentralised systems like ad hoc networks is that it is unrealistic to take a top-down "bird's eye" view. Thus we model the set of peers as those peers of which a specific peer *is aware*.

We also make several assumptions. We assume that neighbour discovery and routing services are in place. We also assume that some kind of service registration and discovery is available to allow peers to reach an understanding of the set of services available and their associated trust thresholds. It should be noted that the assumption of a common threshold across all peers that provide the same service is something of a simplification. Even having all peers share an understanding of the relative meaning of trust threshold values is non-trivial.

We also assume authentication of identity. This could be done, for example, by having peers exchange public keys on their first interaction (in fact the public keys could be used as unique peer identifiers). Further messages between those peers could then be signed by the sending peer's corresponding private key so that the recipient could at least be confident that the sender is the same entity as that for which it has been building up a trust profile.

Topology:

Let $V_i = \{1,...,N_i\}$ be the set of peers of which peer *i* is aware. Some of these peers will be *adjacent* to *i*, normally by reason of network topology. This can be modelled by an adjacency vector, $A_i = (a_{i,j})_{j=1,...,N_i}$, where $a_{i,j} = 1$ if pair (i,j) are neighbours and $a_{i,j} = 0$ otherwise.

Services:

Let $S = \{S_1,...,S_M\}$ be a set of services that may be provided. Each peer *j* provides a set of services $S^j \subset S$. Some peers will just be service consumers, so S^j will in those cases be empty.

Trust thresholds:

Each service S_x has an associated *trust threshold* t_x, where $0 \le t_x \le 1$.

Representing Trust:

We denote the local trust that peer *i* has in peer *j* as $T_{i,j}$. Each other peer $k \in V_i$ will maintain its own local view of trust in *j*, which may, as we shall see, influence the local trust of peer *i* in *j*. Note that $T_{i,j}$ is a trust *vector*. In the simplest case, this can just be a number, but such a number may be associated with other attributes that relate to it, such as confidence in the trust score or recency, or the service(s) to which it relates.

Trust initialisation:

In the case where *i* has no prior knowledge of *j*, we will have $T_{i,j} = x$ where *x* is the *default trust*.

Trust decision: (using trust in service protection)

When peer *j* attempts to use service S_x provided by peer *i*:

- Service use is permitted if $f_x(T_{i,j}) > t_x$, where the function f_x maps trust vector $T_{i,j}$ onto a scalar number in the range (0,1)
- Otherwise peer *j* is blocked from using service S_x

Trust update following transaction:

After a transaction, by peer *j* on peer *i*:

- positive outcome: $T_{i,j}$ is increased according to some algorithm
- negative outcome: $T_{i,j}$ is reduced according to some algorithm

In general, following a transaction, we update $T_{i,j}$ according to:

$$T_{i,j} := f_e(T_{i,j}, E)$$

where E is a vector of attributes related to the transaction and f_e is a function defining how trust is updated based on transaction experience.

Trust update following referrals by a third party:

Peer *i* may receive a message from a third party peer, *k*, indicating a level of trust peer *j*. This can be modelled as peer *i* adopting some of peer *k*'s trust level in peer *j*, $T_{k,j}$. In general, following such a third party recommendation, we update $T_{i,j}$ according to:

$$T_{i,j} := f_e(T_{i,j}, T_{i,k}, T_{k,j})$$

where f_r is a function defining how trust is updated. This trust transitivity depends on $T_{i,k}$, as peer *i* can be expected attach more weight to a referral from a highly trusted peer.

Some algorithms for updating trust parameters

The way trust is updated based on both experience and recommendations has a profound impact on the usefulness of this kind of overlay system. Note that peers are autonomous in our model and each might implement a different algorithm for updating and using trust. It can also be expected that a peer's behaviour in terms of handling trust may change if it is hijacked. Potential trust update algorithms include:

- *Moving average*: Advanced moving averages are possible, where old data is "remembered" using data reduction and layered windowing techniques.
- *Exponential average:* Exponential averaging is a natural way to update trust as recent experience is given greater weight than old values, and no memory is required in the system, making it more attractive than using a moving average.
- *No forgiveness*: This in a draconian policy where a peer behaving badly has its trust set to zero forever. Even more extreme is where a peer that is reported by a third party as behaving badly has its trust set to zero forever. This could be used if a particularly sensitive service is misused.
- *Second chance (generally, n^{th} chance)*: Draconian policies are generally not a good idea. IDS and other security systems are prone to false alarms. A variation on the "no forgiveness" approach is to allow some bounded number of misdemeanours.
- *Hard to gain trust; easy to lose it*: To discourage collusion, there is a case for making trust hard to gain and easy to lose.
- *Use of corroboration*: To prevent an attack by up to *k* colluding bad peers, we could require positive recommendations from at least *k*+1 different peers.
- *Use of trust threshold for accepting recommendations*: It is possible to model the ability to issue a recommendation as a kind of service usage on that receiving peer. Thus the receiving peer can apply a trust threshold to decide whether to accept that recommendation in the same way as any transaction attempt is adjudicated.

For our initial implementation, we update trust using an *exponential average*. As mentioned above, this is a natural way to update trust, as least where trust is simply represented using a scalar number between 0 and 1. Using an exponential average, we can tune the responsiveness of a node's trust determinations to new information received. Setting a high value for the exponential average parameter (close to 1) means that trust is heavily influenced by new data (compared with long-term experience). Setting a low value for the exponential average parameter (closer to 0) means that trust is just slightly influenced by each new experience or referral.

More specifically, we *update trust based on direct experience* according to:

$$T_{i,j} := \alpha E + (1 - \alpha) T_{i,j}$$

where exponential average parameter α can be viewed as the *rate of adoption of trust*, $0 \le \alpha \le 1$. E is a binary attribute related to how the transaction is perceived (i.e. positively or negatively). Note that having α set to 0 means that the trust value is unaffected by the experience. Having α set to 1 means that local trust is always defined by the latest experience and no memory is retained. The higher the value of α, the greater the influence of recent experience of a node on the trust value recorded for that node. Lower values of α encourage stability of the system. If a succession of experiences of node *j* return the same binary trust evaluation, *E*, then the trust value $T_{i,j}$ converges towards *E* (towards 0 for a sequence of bad activity or towards 1 for repeated normal behaviour).

Peer *i* may receive a message from a third party, *k*, indicating a level of trust in peer *j*. This can be modelled as *i* adopting some of *k*'s trust level in *j*. As well as introducing a new parameter β indicating the level of influence of recommender trust on local trust, we also use $T_{i,k}$, how much peer *i* trusts peer *k*. Specifically, we *update trust based on referrals by a third party* according to:

$$T_{i,j} = \beta T_{i,k} T_{k,j} + (1 - \beta T_{i,k}) T_{i,j} \quad ,$$

where β is a parameter indicating the level of influence that recommender trust has on local trust, $0 \le \beta \le 1$. Note that, the larger the value of $T_{i,k}$, (i.e. the more *i* trusts *k*), the greater the influence of *k*'s trust in *j* on the newly updated value of $T_{i,j}$. Note that, if $T_{i,k} = 0$, (i.e. *i* has no trust in *k*), this causes $T_{i,j}$ to be unchanged.

Note that parameters α and β can be varied as required to implement strategies mentioned above, such as:

- *Hard to gain trust; easy to lose it:* here α and β are set to a lower value when trust is to be increased than for a decrease.
- *Use of trust threshold for accepting recommendations:* here β is set to zero where trust in the referring node is less than the threshold (say 0.5, for example).

C. Rating Agencies

The trust values computed by the P2P reputation system are *subjective* and in some situations may not be acceptable. For example, certain business transactions could have higher constrains. Because of that, we see the need of using Rating Agencies which issues certificates to registered users. Certificates issued by the agencies should be *objective* and hence more trustworthy. Unlike the P2P reputation system which uses recommendations expressed by users, the rating agencies make use of authorization certificates issued by different domains. A Rating Agency is a dedicated service that could be offered by each of the peers. Agencies specify predefined criteria on which users are registered (i.e. necessary credentials). Each entity decides on its own to register or not with an agency. Each SP or user decides on its own which agencies to trust. Similar to CPs in the Identity Management model, rating agencies are distributed across the system, establish trust relationships between each other and translate certificates issued by trusted agencies. Moreover, agencies across the system cooperate with each other to retrieve information about unknown users.

D. Learning

The trust model we propose for DE is based on a multidisciplinary framework first introduced in D and adapted in Figure 3. The right side column represents possible technology platforms suitable for DE service execution management. The left side column represents the trusted environment that SMEs use to perform their business goals.

Trust can be provided in a DE in two ways:

- Through experiences inside the system

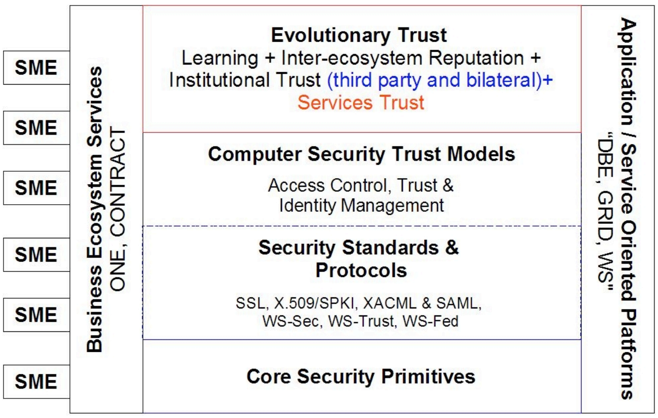


Figure 3: A Multidisciplinary Framework for Digital Ecosystems

- Through institutional trust: an institution CA initially decides to trust a user by examining existing institutional trust relations with the institutions who know the user.

This will create an independent and evolutionary platform capable to adapt and evolve on the basis of the evolution in social institutional trust.

V. Experiments

Initial simulation work has been done to assess the dyamics of our peer-to-peer reputation system.

Experiment 1: Convergence and stability of trust scores

In our first simulation, we have a small fully connected network of fifty peers and a single service. A default initial trust score of 0.2 is used for illustration purposes (to distinguish the bad from the simply unknown) and the trust updating algorithm is a simple exponential average for both direct experience and recommendations. Of the fifty peers, a small number produce a varying level of bad activity. Figure 4 shows that the trust recorded by a good peer ("node 1") for each of these less reliable peers converges to a different value, depending on the level of bad activity encountered. This is encouraging as we can envisage filters that work by subjecting activity to a degree of filtering scrutiny that is appropriate to the trust in the source.

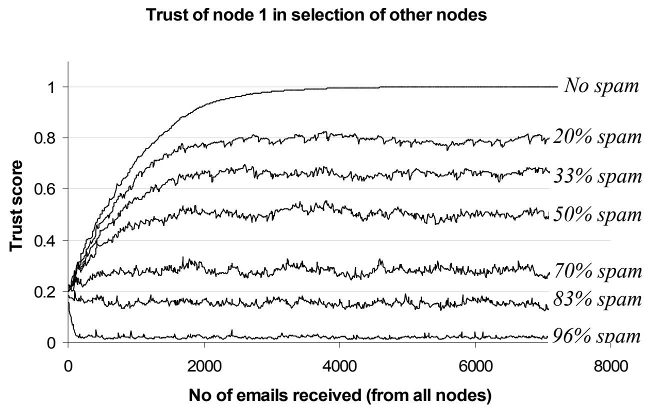


Figure 4: Trust level in peer converges to different value depending on tendency of peer to generate spam.

Experiment 2: Single-service scenario

The objective of this experiment is to see if we can get an improvement in the effectiveness of service access filtering by applying trust scores. The figures below show how this has been achieved in an illustrative case. In this experiment, a network of 50 peers is again simulated; also there is a single "bad guy" who is responsible for 50% of all activity in the system. Trust convergence for normal peers is a moderately fast exponential average. The neighbourhood consists on average of one-seventh of all peers, so we have a sparse, but still connected, topology.

For this experiment, we choose relatively flat (but distinct) probability density functions for bad activity indicators for both good and bad activity. Both have the Gaussian (normal) distributions shown in Table 1. Note the overlap implied by the relatively large standard deviation value for bad service usage.

TABLE I. PARAMETERS FOR GAUSSIAN (NORMAL) DISTRIBUTIONS USED IN EXPERIMENTS.

	Mean	Std Deviation
Benign service usage	8.0	4.0
Malicious service usage	1.0	2.0

Spam filters, for example, combine a variety of measures into a suspicion score and compare this score with a pre-defined threshold. For our experiments, a fixed threshold of 5.0 is chosen (*SpamAssassin* default) and used as a benchmark. As can be seen in Figures 5 and 6 below, a significant reduction in both false positives and false negatives can be achieved with auto-tuning of the threshold (based on trust values). Auto-tuning is of course most effective in a steady-state situation when trust values are quite stable. A range of other predefined threshold values were also tried, but with no better results than the value of 5.0 shown. Choosing a higher predefined threshold causes an increase in false negatives and choosing a lower predefined threshold causes an increase in false positives.

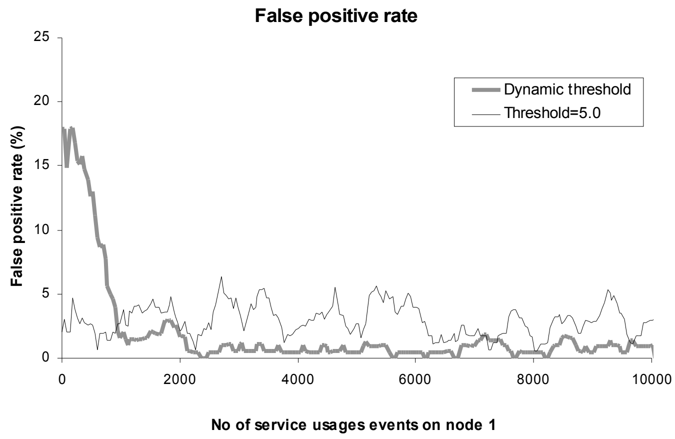


Figure 5. Comparison of dynamic vs. fixed threshold: impact on rate of false positives.

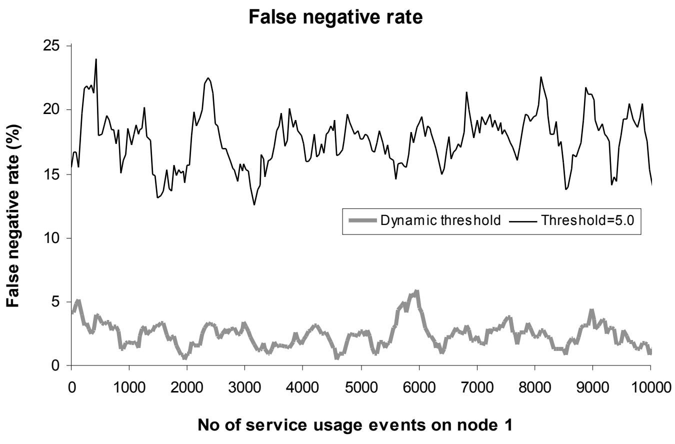


Figure 6. Comparison of dynamic vs. fixed threshold: impact on rate of false negatives.

ACKNOWLEDGMENT

This work is supported by the European Commission FP6 project OPAALS (Open Philosophies for Associative Autopoietic Digital Ecosystems).

CONCLUSIONS

This paper has taken inspiration from different fields related to the Digital Ecosystem philosophy, such as sociology, biology, economics and especially computer science, to propose a novel multidisciplinary trust model for Digital Ecosystems. Initial simulation results of our peer-to-peer reputation system are promising. It is planned to create an integrated implementation of the system during the next phase of the OPAALS project.

REFERENCES

[1] J. Preece, "Online Communities: Designing Usability, Supporting Sociability". Chichester, UK: John Wiley & Sons, 2000.

[2] R. Falcone and C. Castelfranchi , "Social trust: a cognitive approach", In *C. Castelfranchi, Y.-H. Tan (eds.), Trust and Deception in Virtual Societies*, pp. 55-90, Kluwer, 2001.

[3] A. Jøsang, R. Ismail, and C. Boyd, "A survey of trust and reputation systems for online service provision," *Decision Support Systems*, pp 618-644, March 2007.

[4] H. Li and M. Singhal, "Trust management in distributed systems," *IEEE Computer*, pp 45-53, February 2007.

[5] D. Gambetta, "Can we trust trust?" In D. Gambetta (ed.), Trust: making and breaking cooperative relations, pp 213-237, Blackwell, 1988.

[6] F. Nachira (2002), Towards a Network of Digital Business Ecosystems Fostering the Local Development. Discussion Paper. http://www.digital-ecosystems.org/ (last access 15.08.07).

[7] DBE Deliverables n. 32.1 on "DBE Legal Analysis (public and private regulatory framework) and Taxonomy" and n. 32.2 on "Benchmark and Analysis of 'Legal ICTs'", at http://www.digital-ecosystem.org/ (last access 15.08.07).

[8] C. Dellarocas, "Immunizing online reputation reporting systems against unfair ratings and discriminatory behaviour". In EC '00: Proceedings of the 2nd ACM conference on Electronic commerce, p. 150-157. ACM Press, 2000.

[9] H. Koshutanski, M. Ion, and L. Telesca (2007) "Distributed identity management model for digital ecosystems". In *Proceedings of International Conference on Emerging Security Information, Systems and Technologies (SECURWARE'07)*, (October), Valencia, Spain, IEEE press.

[10] M. Ion, L. Telesca, F. Botto and H. Koshutanski. "An Open Distributed Identity and Trust Management Approach for Digital Community Ecosystems". Third International Conference on Communities and Technology, (C&T 2007) Michigan, 2007.

[11] J. Douceur, "The Sybil attack," *Proc. Int'l Workshop on Peer-to-Peer Systems*, March 2002.

[12] J. McGibney and D. Botvich , "A trust overlay architecture and protocol for enhanced protection against spam", *Proc. 2nd International Conference on Availability, Reliability and Security*, April 2007.

[13] Telesca L. and Koshutanski H. (2007) "A Trusted Negotiation Environment for Digital Ecosystems". In *F. Nachira, M. Le Louarn, L. Rivera Lèon (eds) Building the foundations of Digital Ecosystems: FP6 Results and Perspectives*. Publisher: European Commission.

Peripheral Simulation Framework for Digital Ecosystems

T. Kurz[1], J. Nougera[2], R. Eder[1], T. Heistracher[1], J. Gabaldon[2]

[1] Salzburg University of Applied Sciences Urstein Süd 1, A-5412 Puch - Salzburg, Austria
{thomas.kurz, raimund.eder, thomas.heistacher}@fh-salzburg.ac.at

TechIdeas Llacuna 162-164, Suite 102, Barcelona 08018, Spain
{javier.noguera, jesus.gabaldon}@techideas.es

Abstract

The paper at hand describes the distributed architecture of an P2P simulation framework. Furthermore, simulation cases on critical mass and clustering mechanisms are explained. The peripheral approach bases on the biological simile of the peripheral nervous system in the human body. Consequently, the simulation framework is intended to work distributed and directly on the P2P nodes whose behavior will be simulated on larger scales. That leads to a better performance and additional testing capabilities for the network.

1. Introduction

The conceptualization and ongoing implementation of a network and service infrastructure for SMEs that bases on biologically inspired similes needs thorough simulation and emulation capabilities that accompany that development. This network and service infrastructure acts as a *Digital Ecosystem* as it is mediating optimum services amongst SMEs that are provided from a pool of services within that network. Finding optimum services relies on Genetic Algorithms that guarantee the best fitness available for the respective services.

In this paper we present an extended variant of the *EvESimulator* simulation framework that is both able to perform simulations of the evolutionary framework and to work as a visualization tool for the real-world P2P network.

2. The Evolutionary Environment Simulator

The Evolutionary Environment Simulator, short EvESimulator (see [5] and [7]), was designed and implemented in the DBE[1] project in order to run simulations on a biologically inspired P2P system - the Evolutionary Environment (EvE) (see [8]). Beside the simulation of the EvE the EvESimulator acted as a framework for understanding, visualizing and presenting the DBE concepts to many stakeholders within and outside the DBE project.

As the initial attempt was to do a simple simulation and visualization of the EvE, the simulator was based on the Recursive Porus Agent Simulation Toolkit (Repast) (see [2]). Additionally to a basic framework for agent based simulations Repast provides the basis for network visualization components. Utilizing Repast and adapting it to the needs of DBE allowed rapid development leveraging an established simulation framework. Nevertheless, drawbacks of using repast are outlined in Chapter 5.

The objectives of the EvESimulator were adapted and extended during the work with the framework and can be summarized at the end of the DBE as follows:

- Simulation of the intended behavior of the DBEs Evolutionary Environment
- Visualization of a network of operational SMEs and consequently the DBE
- Facilitation of import and export of data from partners in various formats
- Intensification of interdisciplinary collaboration

Even though the use of the EvESimulator and the integration of stakeholders from business, social- , natural-, and computer science was successful, the modifications on the initial framework lead to problems in performance and extendability which will be summarized in Chapter 5 of this paper.

[1] Digital Business Ecosystem, contract number 507953

2.1 Agent Model

The agent model as well as the service model for the EvESimulator are based on the assumption that a Digital Business Ecosystem (DBE) or a Digital Ecosystem (DE), respectively is a network of Small Medium Enterprises (SMEs), each offering and consuming services. We did not take into account explicitly the differences in B2B services versus C2B services but we assumed that the descriptions of the services in this DE will be described in a business language similar to BML [2] or SBVR [3].

Before describing the model of EvE agents a short definition of the term *service* in the context of the EvE and EvESimulator is given. EvE-Services are meta-descriptions of services which are persisted physically in the P2P network. These meta-description contains service name, basic SBVR description, reference to the actual service and history data of service migration through the network. For detailed information see [6].

Each of these EvE services (in the following just services) are modelled as a vector of attributes in which each attribute has a certain value. These attribute value pairs identify the service. A very simplified example could be a *hotel room service* which is characterized with the attribute *stars* with a value from 0 to 5, an attribute *price* with a certain value and an attribute *minibar available*, which can be zero or one. The more attributes in the overall system, the more difficult it is to find a perfect attribute match when searching and requesting specific services.

Contrary to the services, the activities of the SMEs are modelled as agents in the simulation. Each of the nodes in the network represent one SME. As regards the simulation we interchangeably use the terms SME, actor, agent and node. Each SME or actor, respectively, has a local service pool (LSP), a list of services on offer (Portfolio), and a list of services on demand (see Figure 1).

The local service pool is equivalent to a local repository for services. Besides the SME stating it's clear intentions and interests in the network, the system tracks the SME's service consumption as well. The local service pool data, besides configuration information, is used to set up the environmental conditions of the services which migrate [4] to the SME. Consequently, services which are probably useful for this SME will reside at the local service pool.

In the Portfolio, the SME manages a list of services which it offers to others. When a new service is on offer,

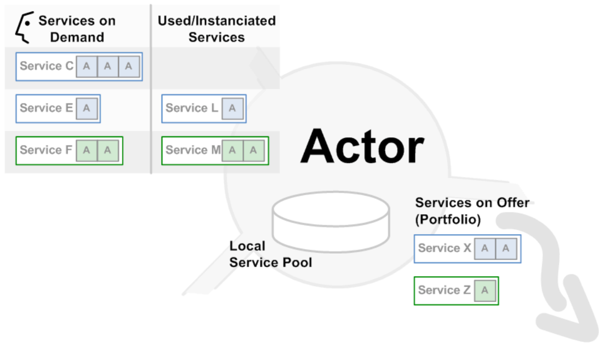


Figure 1. Representation of an Actor (SME or agent, respectively) within the EvESimulator. [6]

it is simply copied to the local service pool of the producer SME and is distributed to all known neighbors of this SME.

The list of services on demand represent also a kind of profile of an SME in the simulation because it indicates the interests of this SME. When new agents are created for the simulation, a list of demanded services is produced as well. Depending on the scheduler of the simulation the SME gets the allowance to request one of these services on demand and looks up the local service pool for adequate services or service compositions. This service search and combination is done by a genetic algorithm and a fitness function, respectively.

2.2 Simulation Examples

In order to see which kind of simulations we want to perform with the new framework introduced later in this paper we sketch here with (1) critical mass assessment and (2) clustering two of the main important simulation cases produced using the EvESimulator.

Critical mass assessment simulations were used to compare performance of global service repositories with distributed clouds of local service pools at SMEs. When an SME in the simulation gets the allowance to request a service one Genetic Algorithm (GA) is instantiated to search for service combinations in both, the global service repository and the pre-selected local service pool.

The findings on critical mass assessment were the following: After overcoming a bootstrapping phase in which the pre-selection of services in the local service pools is performed, the results of the searches were significantly better in the distributed version. Moreover, it turned out that the total number of attributes in the system for describing the services is an important factor for the performance difference and behavior of the search comparison. Contrary, the differing number of SMEs used in the simulations had no

[2] Business Modelling Language (BML) was developed in the DBE project for modelling small and medium enterprises based on ontologies. [4]

[3] Semantics of Business Vocabulary and Business Rules (SBVR) is a specification of the OMG. [9]

[4] Migration in biological terms is the movement of individuals from one biotope or habitat to another. This movement can be driven by better environmental conditions in the new biotope or habitat. [1]

major influence on fitness but on the time of the bootstrapping phase.

In the second simulation case on clustering it was assumed that the network connections will cluster SMEs depending on their service consume and offer behavior. That should lead to a natural clustering of SMEs and therefore a more effective and self organizing topology of the network. The most important concepts used were the simile of migration of services and the use of service history.

After several simulations it can be concluded that at least for a small number of SMEs the simulated network behaved as expected. Nevertheless, the clustering using the output from the GA based optimization performed much better than the user interaction based approach. For more details on these simulation cases see [6] and [7].

The number of nodes in the networks as well as the number of services in these simulations were quite small. Nevertheless, the time for running these simulations took hours to run on one single PC. The interest of the new simulation architecture has to take these shortcomings into account, which are discussed in Chapter 5 of this paper.

2.3 Collaboration Framework

As already mentioned the EvESimulator works not only as a simulation toolkit but also as a basis for collaboration. Project stakeholders include natural science, social science, computing and also business partners.

One of the core components of the Evolutionary Environment are Genetic Algorithms. It was tried to open the EvESimulator also for future implementations of Genetic Algorithms as well as other optimization methods. A clearly defined application programming interface for optimization algorithms was the basic building block for implementing four different algorithms till the end of the DBE project. Besides that, the model and configuration user interface is designed to allow reconfiguration or even extension of the current framework.

The focus of the efforts for cross domain collaboration of the EvESimulator was to customize the simulation and visualization capabilities of the framework also for non technical experienced stakeholders. Consequently, a CSV file import for social network topologies was implemented as well as XML based import and export files. Furthermore, a Java Webstart application provides a one-click installation for the main components of the simulation framework.

As the computing group had to implement the EvE, the EvESimulator acted as the simulation for finding also parameters for the settings of the real EvE. Unfortunately, because of the complexity of matching SBVR models and a lack of resources the EvE could not be fully finished. Additionally, the settings for the EvE strongly depend on the existing structure and size of the SME network and therefore, the parameters are expected to be unique for each application region and also for the evolutionary framework in the OPAALS project. At this point the user friendly user interface supports an easy simulation of networks and finding of parameters needed to stabilize the network at least in the bootstrapping phase.

From a business perspective the most attractive part of the EvESimulator is probably the visualization component. The simulator comes with a network visualization and a small webserver for showing the key facts about each SME. As the simulator can present different scenarios and the simulation can be slowed down, an SME can easily understand what the concepts of a DE are all about and how the system works based on real-world parameters.

3. The ServENT

The ServENT is the application that contains Fada (the P2P framework [10]), Swallow (the server container [3]) and a set of core services that add value to each peer and allow peers to collaborate and work together. Each peer of the Digital Ecosystem network is deployed with these services, that collaborate with the other services deployed in other nodes in order to create the ecosystem behaviour. Such services are:

- Knowledge Base (KB) stores meta-data information about the keep track of services deployed in the node and allow semantic queries.
- Distributed Storage System (DSS) along with other DSS services deployed in the P2P network, this services creates a virtual disk, using distributed hash table and replication policies, where user and other services can store information.
- Accounting generates a report about transactions between nodes of the network.
- Identity allow user identification using certificates. It can deny the execution of some or all services to unauthorized users
- Portal web application for easy management of the ServENT capabilities
- Evolutionary Environment (EvE; [8]) adds biological inspired service distributions using the data that can be found in the network, such as number of services, neighbour nodes, executed services, response time, etc. in an experimental stage

These core services provide only the basic functionality in order to create the Digital Ecosystem network. Other basic services have to be created and deployed by each user

or company. These services are plain Java classes that need to be deployed in the ServENT along with the necessary libraries and information about its capabilities.

Each service deployed in the ServENT publishes its interface in the same way a Web Service does, but both approaches are still not compatible. Users can search and find services, download the interface and incorporate it into their own service or application.

Services are also called 'adapters' because most of the times they will not codify the whole service, but a little proxy that will translate the requests and responses to legacy calls. This is the way how legacy applications can be shared into the P2P network. This allows service providers to deploy services with a well defined interface for communication and sharing of information, and codify the adapter to extract or insert information within a centralized application. This behaviour is required by the OPAALS project in order to distribute and share classical knowledge services such a wiki, a blog and forum spaces.

4 Core Services

The difference between core and basic services is that core services are part of the Digital Ecosystem infrastructure.Therefore, they are shared by all, or almost all, the peers in the network. Basic services, on the other hand, are specific applications owned by a few peers.

Another difference between core and basic services is that core ones have access to the ServENT internals. They can stop the ServENT itself. And they can stop, start, restart, deploy and undeploy other services, or even modify their internal ClassLoader. Core services were thought to be a part of the ServENT itself, and changing its behaviour which can be modified completely. The deployment of core services can be restricted or denied by the administrator of the computer where the ServENT is intalled.

As one can see, there are a few core services and lots of basic ones. Usually we create a new core service when we want to induce a new behaviour in the network. A core service A is designed to search for other services A in the network and communicate with them. All nodes without a service A deployed are not part of the new community and do not share its behaviour.

Examples of possible core services could be: file sharing, instant messaging communication, collaborative document edition, etc. All these examples involve two or more users that need to own the core service in their own ServENT. Examples of basic services are: hotel reservation, digital printing, bet gaming, buy items, etc.

5. Limitations of the Current Approach

Even though the current EvESimulator model was intended to support the creation of a Evolutionary Environment in a Digital Ecosystem from the beginning on, issues arose during the implementation and simulation. Taking advantage of the interdisciplinary group of stakeholders, the *EvESimulator* now reflects the feature requests of many of them and the model is quite complex.

In this paper we evaluate critically the *EvESimulator* for the needs of OPAALS and consequently present a new architecture for overcoming the main issues. Summarized the main shortcomings of the *EvESimulator* are the following:

1. Complex model
2. Based on Repast
3. Lack of performance
4. Difficulties in estimating the behavior of a real P2P network

In the following we describe these points in more detail:

The initial model described in [7] was modified several times in order to fit to the needs of not only the natural science domain but also to other stakeholders in the project. Although, the *EvESimulator* comes with a transformation method to transform the previous described model into a binary representation for running generic Genetic Algorithms, considerable computing power is needed to do the transformations in large scale networks. In order to be ready for e.g. simulations using real SBVR models it will be necessary to even extend the existing model. It is important to note here, that the existing model is based on the underlying Repast framework model which increases the complexity of the model considerably.

Utilizing the Repast framework for the *EvESimulator* seemed a reasonable approach in the beginning. The main point is that one can model without considering the details of an agent based simulation toolkit and Repast provides a basic agent model, scheduling and visualization capabilities. These reasons are still valid but nevertheless it was figured out that decentralization, customization to the needs of a Digital Ecosystem and encapsulation of our main building blocks (Genetic Algorithm, visualization and configuration) of our model is not easy to achive. Repast is built in order to support several different types of simulations and as regards the *EvESimulator*, one model would be sufficient for the moment and for the sake of performance it should be lean and open to distribution.

Continuing with the issue of performance, the *EvESimulator* is a stand alone application which runs currently on one machine. Of course the simulations can take longer or

run over night (and also grid computing systems were considered as a solution already), but for a big number of nodes all these approaches seem not very satisfying. As indicated in [7] with the current model and genetic algorithms running in parallel the time span of one simulation with 50 nodes takes about half a day on high-end PC hardware. If we consider large scale free networks with this setting it is almost impossible to run the simulations in the current architecture and we have to think of a more distributed approach (see Chapter 6)).

Besides the issues in estimating the behavior of Small and Medium sized Enterprises where we get support from the social science group of the DBE and OPAALS projects (see [7]) it is hard to predict the behavior of a P2P system at work. The availability, network traffic, birth and growth behavior of such a network can be simulated only in a difficult manner under real-world conditions at the moment. Therefore, the settings for the network have to be obtained and related to the P2P network of choice.

6. Peripheral Simulation Framework Architecture

In the following we are proposing a peripheral simulation framework architecture for a Digital Ecosystem. 'Peripheral' in this case means that many activities in the simulation are distributed in the network like in a peripheral nervous system. The peripheral nervous system in the human body is, opposed to the central nervous system, exposed to external influences like the nodes in a P2P system are exposed to the environment they are executed.

The exposure to external influences and the distribution to a P2P network has some drawbacks for the simulation but has also one important benefit which is: reality. The last issue mentioned in Chapter 5 of this paper was the difficulty to estimate the behavior of the actual system and to apply conclusions from simulations directly to the real network. By doing the simulation already in a distributed way the simulation already utilizes the real network and estimations for certain parameters are not longer needed.

Although the planned peripheral simulation framework could be implemented on any P2P system with few modifications, it was decided to use the previously described servENT infrastructure. The reason for this is that the servENT P2P network was already developed in the DBE project for the application in a scale free P2P network for digital ecosystems and so it seems to be the best choice at the moment.

6.1 Model and Architecture

As can be seen in Figure 2 it was planed to distribute the simulation components over the network and coordinate

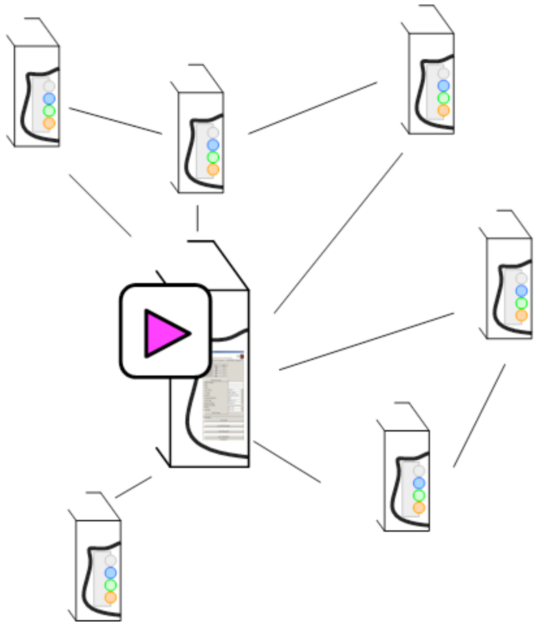


Figure 2. Distributed *EvESimulator*.

and appraise the simulation from one node in the network. For the moment this base-node does not need to be implemented as a servENT service but it can be considered to move it as one as well. All the remote components are modelled as servENT core-services and therefore have access to the functionalities of the local services and servent itself.

The base-node of the simulation acts as the central point of coordination and creation. It is planned to take over most of the user interface elements including the import and configuration of the network from the current *EvESimulator*. Consequently, input about the topology and different types of actors within an SME network can be set by a regional social science officer for example.

Taking the initial settings into account the node has two main tasks. First, it acts as the creator of the agents themselves. This is achieved by having agent-factories in place as well as service factories. Parts of these factories will be implemented in the agent to enable the agent to create services on it's own. According to user settings the agents can be created and pushed to the agent pools, distributed over the network (see also Section 2). After creating and defining the behavior of the agents, they can act completely independent.

Restricting the total independence, the second task of the servENT is coordination. All the agents are reporting to the base-node on a regular basis. For the sake of coordination basic adjustment events like tick-speed [5] can be

[5] The *EvESimulator* on top of the Repast toolkit is working based on ticks. One tick presents one time step without an assigned time span.

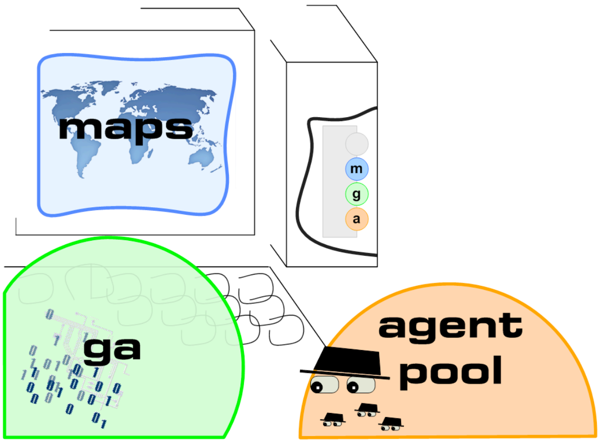


Figure 3. Core Services within one ServENT node.

passed to the agents during the simulation as well. Furthermore, the base-node has to observe the reliability of the nodes and check if some of them are offline. Although the servENTs core services have to implement an on-destroy method where final logs can be sent to the basis-node it cannot be taken as granted that these methods will be called in any circumstance.

Important as a coordinator is also the initial search for servents in the environment. Sepending on the uptime of the nodes the base-node can use the servents lookup service to detect the nodes available during the simulation time.

6.2 Core Services of the Simulation

The idea of the decentralized components of the *EvESimulator* made it necessary to split certain components from the core simulation which need the major part of the processing power. Therefore, three main components are chosen to be modelled as servENT core services (see Figure 3). They are namely the agent pool, the Genetic Algorithm (GA) and the visualisation component (Maps).

Firstly, the agent pool acts as registryof agents which act as virtual SMEs. Depending on the configuration an agent can have an servENT to operate from or several agents can share the resources of an servENT node. This is necessary in order to avoid setting up as many servENTs as nodes in the network which would be too time/consuming for larger network simulations. As the agent can access all services on its host servENT, it can also call methods of these services or of the servent itself and therefore the servent or certain services can be tested for reliability or performance along a simulation run as well.

The second service is the Genetic Algorithm (GA) service. Based on a common interface there can be several different implementations of GAs which run on the network. Beside the reusability of such a GA service, the distribution of the optimization will bring most probably the highest performance gain of the new architecture. The GA service will be called directly by an agent when the agent gets the allowance to proceed a request. If there is no GA service installed on the same servENT node, the P2P system automatically searches for other GA services in the network and therefore distributes the workload as well.

Finally a visualization of the whole network is planned for the second phase of the OPAALS project. Initially intended to be a visualization UI for the real network we plan to implement the visualization so generic that both, the visualization of the real P2P network as well as the visualization of the simulated evolutionary framework will be feasible. For a better visualization the planned utilization of Google-Maps will give the network an even more realistic appearance.

6.3 Additional Changes, Benefits and Drawbacks

Besides the changes in the architecture we learned from the previous experiences in the DBE project that we need to implement a lean simulation framework. That means that the suggested architecture should work without the use of Repast. Even though the implementation of the model, scheduler and agent-structure means additional effort, the benefits of a more flexible framework which can be customized to the needs of a Digital Ecosystem is crucial.

Summarizing the benefits of the new architecture and having the limitations of the old version in mind, the new *Peripheral EvESimulator* will succeed through performance and due to the more realistic network behavior. The decision of encapsulating the three main components and the reuse of the upcoming visualization considerably enhances the reusability and the future implementation of an evolutionary framework. Through the utilization of the actual network we learn more about possible customers needs and issues when programming services and the network is already tested during the simulation.

Nevertheless, there are still issues which have to be considered in future implementations. The model is still complex and if at a later stage the natural language based descriptions, e.g. SBVR, will be tested as well the model becomes even more complex. In order to overcome parts of that complexity, agents and services need to be modeled generic so that new service descriptions can be used without taking into account or inheriting the legacy attributes of previous models. Another big issue is the 'peripheral' one. The simulation can be effected by all the external influences as the real network. Consequently, nodes can go offline and the loss of this data or at least a recognition of this lost has to

be considered when designing the log and communication behavior.

7. Conclusion and Outlook

This paper introcuses the model and design of the peripheral EvESimulator. Details of the functionality as well as the fundamental terms are described, enabling the reader to understand first, the limitations of the initial and centralised version of the EvESimulator in DBE, second, the suggested architecture redesign as well as third, the feature enhancements.

Furthermore, based on experiences in the DBE project and based on future needs, the term *Peripheral EvESimulator* was introduced to address the need for a highly distributed agent simulation framework dedicated to simulate the concepts studied in the OPAALS research project. Leveraging this kind of simulation tool the effects of evolutionary concepts in terms of Digital Ecosystems can be studied in future.

Finally, the planned extensions to the visualization components will help to communicate findings to the Digital Ecosystem community as well as to attract new contributors. These new contributors can then profit from a leaner and cleaner toolkit while extending the simulation functionality to their needs.

8. Acknowledgements

This work was funded in part by the EU project OPAALS, contract number IST-034824.

References

[1] M. Begon, J. Harper, and C. Townsend. *Ecology: Individuals, Populations and Communities. Third Edition*. Blackwell Publishing, 1996.

[2] N. Collier, T. Howe, R. Najlis, M. North, and J. R. Vos. Recursive porus agent simulation toolkit - repast. Webpage. http://repast.sourceforge.net. Last accessed on 06/22/2006.

[3] T. D. Consortium. Swallow. Webpage. http://swallow.sourceforge.net. Last accessed on 14/09/2007.

[4] M. de Tommasi. D15.1 - Business Modelling Language 1.0. Version 1.0, April 2005.

[5] T. Kurz, M. Duerr, C. Ehgartner, and R. Greil. Evolutionary Environment Simulator - evesim. Webpage. http://evesim.sourceforge.net. Last accessed on 12/22/2006.

[6] T. Kurz and T. Heistracher. Simulation of a Self-Optimising Digital Ecosystem. *IEEE First International Conference on Digital Ecosystems and Technologies*, 2007.

[7] T. Kurz, G. Marcon, and A. P. Hisanaga Okada, Thomas Heistracher. D9.2 - Report on Evolutionary and Distributed Fitness Environment. Digital Business Ecosystems Project, July 2006.

[8] C. Masuch. Evolutionary Environment Network. Webpage. http://evenet.sourceforge.net. Last accessed on 12/22/2006.

[9] OMG. *Semantics of Business Vocabulary and Business Rules Specification*, March 2006. First interim specification, http://www.omg.org/docs/dtc/06-03-02.pdf. Last accessed on 12/22/2006.

[10] Techideas. FADA. Webpage. http://fada.sourceforge.net/. Last accessed on 14/09/2007.

Ecosystem-Oriented Architectures

[OPAALS Conference 2007]

Sonia Lain, Alastair Munro
Surgery and Molecular Oncology
University of Dundee
Ninewells Hospital
Dundee, United Kingdom
e-mail: s.lain@dundee.ac.uk, a.j.munro@dundee.ac.uk

Gerard Briscoe, Paolo Dini
Digital Ecosystems Laboratory
Department of Media and Communications
London Sch. of Economics & Political Science
London, United Kingdom
e-mail: g.briscoe@lse.ac.uk, p.dini@lse.ac.uk

***Abstract*– A primary motivation for research in digital ecosystems is the desire to exploit the self-organising properties of natural ecosystems. Ecosystems are thought to be robust, scalable architectures that can automatically solve complex, dynamic problems. What can Digital Ecosystems learn from cellular biology? How might cellular control systems be inspirational? What analogies can be drawn: can cell biology be applied to software services, bucket biology to economic modelling? These are wide-ranging questions that are motivating a number of interdisciplinary research teams and that this paper begins to address.**

ecosystem, stigmergy, autopoiesis, SBML

I. Introduction

Biomimicry in engineeringis a long-established process, and probably started with Leonardo Da Vinci during the renaissance, with his flying machine designed around the anatomical structure of birds. Despite his genius it is still debated as to whether his flying machine would have flown, which provides a reminder that such research may not be a straightforward process.[1]

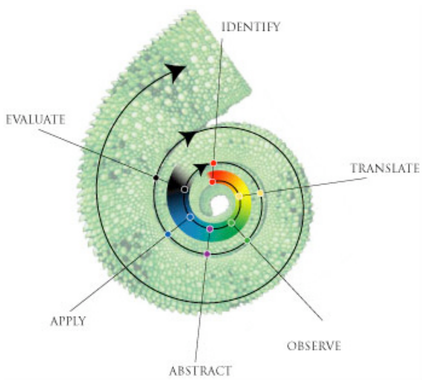


Fig. 1. Biomimicry Design Spiral

[1] http://freethinkr.wordpress.com/2007/06/

Nature has been in the research business for 3.8 billion years and in that time has accumulated close to 30 million "well adjusted" solutions to a plethora of design challenges that humankind struggles to address, with mixed results. Biomimicry is a re-emerging discipline that seeks sustainable solutions by emulating Nature's designs and processes. There are some great opportunities to learn how Nature has designed elegant solutions for some tough human-made problems.

Biologically inspired computing is also far from new. Well-known existing examples include, Neural Networks (NNs), Artificial Intelligence (AI) and evolutionary computing. We now wish to take this further, creating an Ecosystem Orientated Architecture (EOA) with the essential elements of biological ecosystems, building upon our previous work from the Digital Business Ecosystem (DBE) project [1]. In the DBE project we researched using biomimicry in computing engineering to create a Software Agent Ecosystem (SAE), called the Evolutionary Environment (EvE), mimicking the processes of evolution and ecosystems to create an EOA.Despite its success, the key distinction between our SAE and a biological ecosystem could be stated succinctly as a lack of autopoiesis, which is a construct for self-organisation of biological ecosystems. To create a digital ecosystem that demonstrates autopoiesis, we will need to determine the design patterns of the autopoietic constructs and algorithms common to all biological ecosystems, which we will explore in this paper.

II. Definition of Key Concepts in Biology of Relevance

Definition of key biological concepts of relevance to mathematics and computer science, together with some thoughts on the relationships between biology and computing.

A. *Defining of terms*

The links between biology and computing are important but bedevilled by a confusing nomenclature. A variety of terms[2] are in current usage, but few of them have been well-defined or characterised and communication is often confused. We will need to work with our sociological and linguistic partners in the OPAALS project to resolve some of these dilemmas. I am unsure as to the extent to which a uniform vocabulary is necessary, desirable, or even feasible.

There should be, at the core of all this, an appreciation that the interactions between biology and computer science take place along a two-way street. Computers can help us to make sense of biological systems and biological systems may help us to devise better computational techniques and technologies. As soon as we raise this issue we enter a thicket of controversy: we are in the territory here of the debates between reductionism and vitalism; between a determinist world view and one that accommodates the notion that the whole may be somewhat greater than the simple sum of its parts. Since this debate has raged, unresolved, for centuries we should respectfully acknowledge its existence but move on regardless, much as a river flows round a rock in the riverbed: the obstacle is there but need not impede progress, we can move on past and around it.

The BISTIC Definition Committee of the US National Institutes of Health issued a brief paper on July 17th 2000 (BISTIC 2000). It provided the following definitions for computational biology and bioinformatics:

"Bioinformatics: Research, development, or application of computational tools and approaches for expanding the use of biological, medical, behavioral or health data, including those to acquire, store, organize, archive, analyze, or visualize such data.

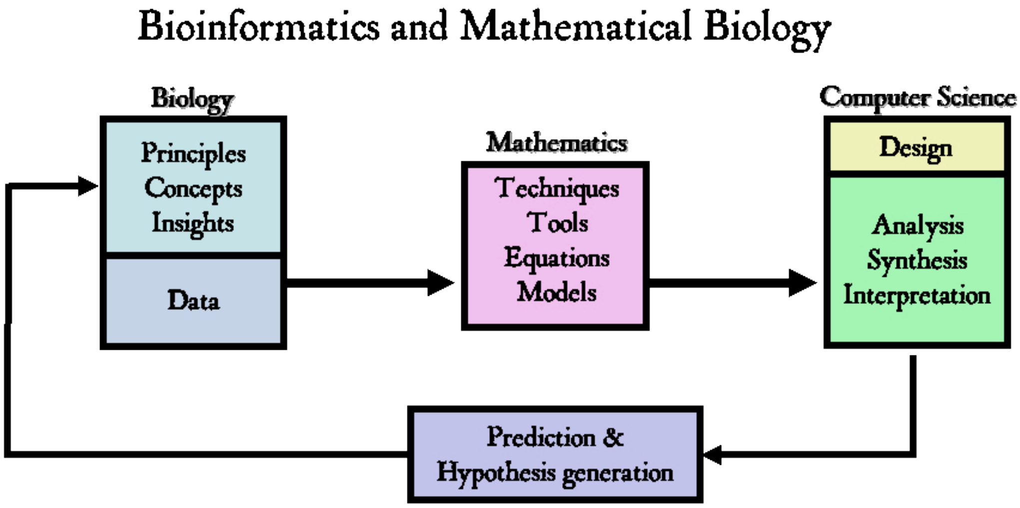
[2]Including: biocomputing; bioinformatics; biomathematics; computational biology; biological computation; evolutionary computation; amorphous computing and so on

Computational Biology: The development and application of data-analytical and theoretical methods, mathematical modeling and computational simulation techniques to the study of biological, behavioral, and social systems."

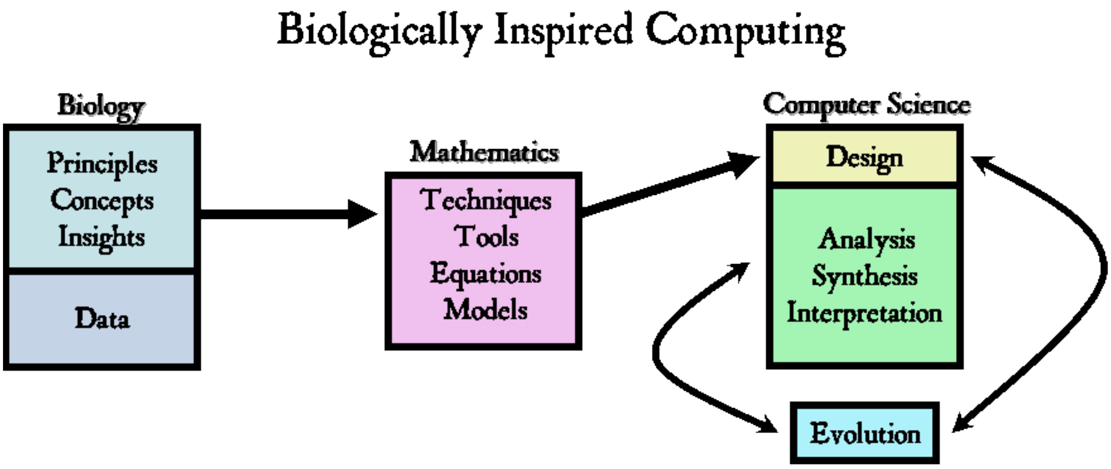
These ambiguous definitions achieve acceptability at the sacrifice of clarity. Perhaps it might be useful, for the purposes of the OPAALS project, simply to sketch out at this point the flow of information between biology, mathematics and computer science.

B. Mathematics and computer science in support of biology

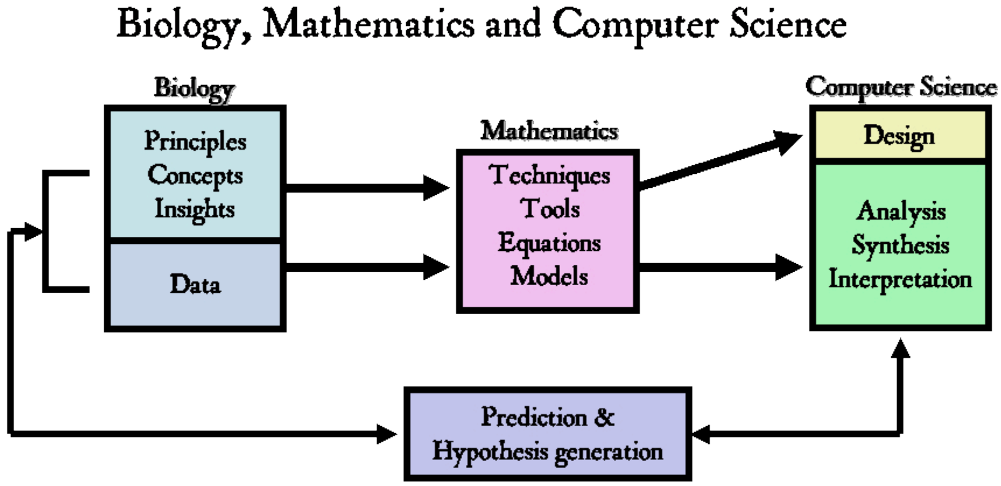
Firstly we can consider the ways in which mathematics and computer science can support biology:



Secondly we might consider how biology might support computing:



Finally, we might consider how the whole picture fits together



The importance of evolution by Natural Selection

The process of evolution by means of natural selection is an important component of optimisation both in biology and in computer science, a point made succinctly by John Maynard Smith "Where an engineer sees design, a biologist sees natural selection". Smith, after a professional lifetime studying evolution and its mechanisms, wrote a paper in which he mused upon the concept of information in biology (Smith 2000). He was, in effect, revisiting territory that had been worked over by Turing, non Neumann and many others. War is primarily concerned with the delivering of energy and the communication of information (Dixon 1976); likewise biology.

Smith, who had a background in engineering, was interested in DNA as form of code:

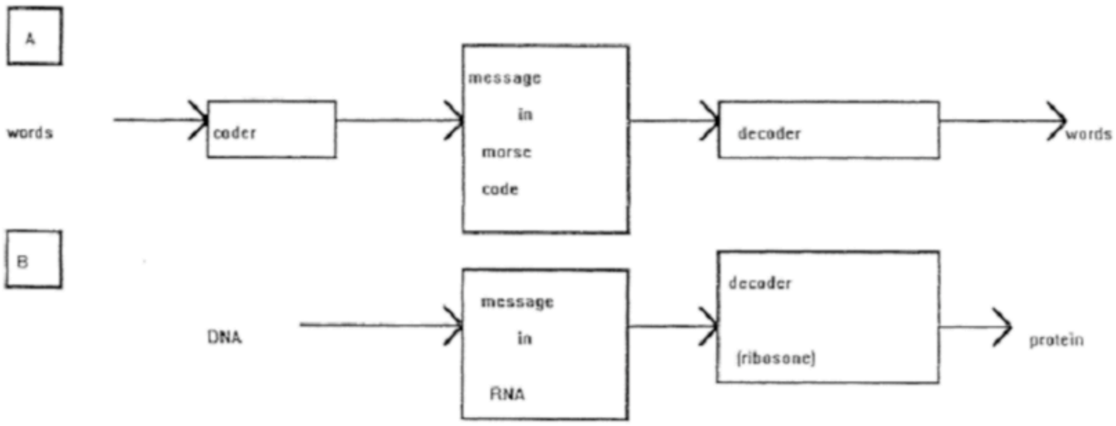
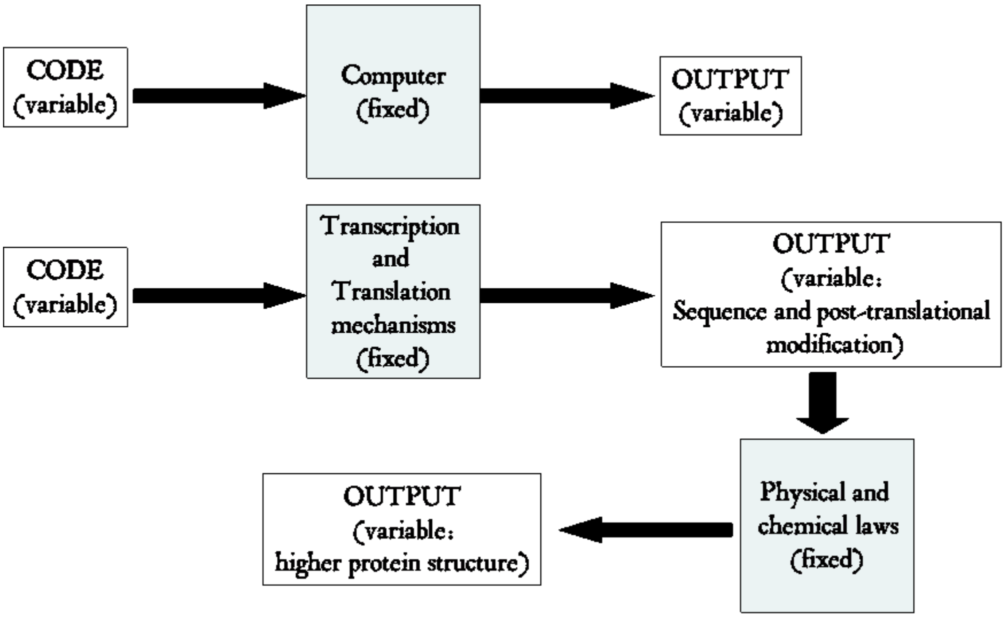


Fig. 2. Comparison of A, transmission of a human message by Morse Code, and B, translation of a message coded in DNA into the amino acid sequence of a protein.

He regarded natural selection as the coder, the agent determining the DNA sequence: just as in manual coding the coder is the person who takes the original text and, using a set of coding rules, converts into code. He points out that engineers have not yet solved the problem of converting information into a form that can be transmitted by chemical means: "..the difficulty of getting information into and out of a chemical medium. The living world has solved this problem". He also notes Monod's

development of the concept of 'gratuite', which we might loosely translate as fortuitous symbolism (Monod 1988). Monod's point was that, just because CAC codes for histidine does not imply that CAC was, for some physical or chemical reason, obliged to code for histidine. The fact that CAC codes for histidine is simply a matter of evolutionary chance. In Smith's words "molecular signals in biology are symbolic". His view is by no means universally accepted as the discussions following his paper illustrate. It should be clear, even from this brief account, that there is much common ground here between biology, computer science, linguistics and sociology. A region that the OPAALS collaboration is uniquely configured to explore and in which we may be able to make important contributions to a debate that is fundamental to questions concerning reproduction, replication, translation and, indeed, the basis of life itself.

The combination of variable components and fixed components that characterises a computing system also, with some additional processes, characterises living systems. There is an encouraging congruity here, encouraging in the sense that it suggests that advances in understanding in one arena might facilitate progress in the other arena.



C. Biomathematics

Biomathematics can be defined as the use of mathematical tools and techniques to model natural and biological processes. It has, by this definition, both practical (how can we predict this biological pattern or event) and theoretical (what mathematical formulation best describes this pattern or event) aspects. Kant proposed that the criterion of a true science lay in its relationship to mathematics: by this criterion biology is merely at the threshold of acknowledgement as a science. Attempts to "explain" nature in mathematical terms can be traced back to Plato and Pythagoras. Roger bacon regarded mathematics as both the door and the key to scientific knowledge ("Et harum scientiarum porta et clavis est Mathematica"). In Dundee and St Andrews the striking figure of D'Arcy Wentworth Thompson strove mightily to instantiate the vision and precepts of Kant and Bacon. His great work

“On Growth and Form”[3] has never had the influence its rigour and originality deserved. Working from meticulous observation of plants and animals, Thompson showed that growth and development obeyed certain mathematical laws. As he wrote in the preface to “On Growth and Form”: “Cell and issue, shell and bone, leaf and flower are so many portions of matter ... Their problems of form are, in the first instance, mathematical problems, their problems of growth are essential physical problems” . He showed that the principles that engineers used to design great bridges were the same principles that governed the structure of a dinosaur’s backbone. He pointed out that a nautilus shell is an example of an equiangular, or logarithmic, spiral.

Building on these foundations, 20th Century biology has increasingly incorporated mathematics into its repertoire of methods. Luria and Delbruck shared the Nobel prize for their mathematical description of the fluctuations in phage resistance in bacteria. Rossi and Lederer showed that radiation-induced cell killing showed a relationship between dose and effect that could be described by a linear quadratic equation. Without the mathematical interpretation of X-Ray crystallographic images Watson and Crick would have been unable to define the structure of DNA and, by extension, the mechanism by which the genetic code could be transmitted. The deciphering of the genetic code was, itself, a mathematical task and one that (somewhat surprisingly) has been overshadowed by the story of the double helix: an example perhaps of the medium triumphing over the message.

There are several levels at which biological problems can be approached: the global and beyond, as in, for example, the Gaia hypothesis; the biosphere; the population; the organism; the cell; the sub-cellular. Each level has different, and often contradictory, insights to offer. One of the key differences, for our purposes here, is between single-celled and multi-cellular organisms. Unicellular organisms (bacteria, yeasts and protozoa) are self-contained and can adapt to extremely inhospitable conditions[4]. Multi-cellular organisms are very different. They may have difficulty adapting to the extremes of physical environment but they can adapt extremely well to changing environments and, importantly, have the ability to adapt the environment itself.

D. The study of disorder is a useful approach to the study of order

One way to approach the question of what makes things work effectively is to look at what happens when things go wrong. In our jargon: we can use pathology to learn about physiology. The development of cancer epitomises the derangement of control mechanisms within cells and tissues and therefore serves as a useful pointer to the systems that are critical for normal cellular husbandry. Malignant transformation holds a distorting mirror to nature. By considering the changes involved in malignant transformation we learn a great deal about the systems that, quietly and effectively, impose the checks and balances that keep cells and tissues functioning normally.

The key components involved in malignant transformation, defined according to their effects on cell behaviour, are shown together with the implications for normal control in the table below[5]. This is a crude summary of extremely subtle systems but has the advantage of demonstrating that malignant transformation is unlikely to involve only one mechanism and that, crucially, the surrounding milieu plays a major role in permitting, or prohibiting, cellular events.

[3] On Growth and Form: D’Arcy Wentworth Thompson 1917

[4] for example the scalding water that emerges from volcanic vents in the sea-bed

[5] This is a development of ideas first put forward by Hanahan and Weinberg Cell 100:57 (2000)

Transforming quality	Biological mechanism	Implication for normal control
Immortality	Telomeres do not shorten	The lifespan of an entity should be finite, otherwise errors and problems will accumulate
Ability to develop and grow in the absence of external signals	Independence from growth factors	Entities should only develop when it is the appropriate time and place for them to do so
Ability to develop and grow in the presence of external signals inhibiting growth and development	Resistance to inhibitory factors	Entities should cease to propagate when they are commanded by the system as whole to do so
Ability to circumvent or avoid programmed self-destruction	Resistance to apoptosis	Self-elimination is an important control mechanism
Ability to breach normal boundaries	Local invasion; Distant metastasis	Architectural boundaries are an important control mechanism
Ability to co-opt normal (untransformed) cells into providing support	Angiogenic competence	The environment can influence, both negatively and positively, entities existing within it.
Avoid elimination by external agents	Avoid immunological attack	The milieu develops powerful mechanisms to deal with unwanted entities (trapping and elimination of errors)

We can extend this consideration of specifically important transforming events into a more general taxonomy of the biological principles of relevance to biologically inspired computing. In no particular order, these principles include:

Duplication and redundancy

Cells possess many different mechanisms for performing identical tasks. there is much unexpressed capability. If a pathway is knocked out then, usually, there are ways round the problem. There are remarkably few totally essential and irreplaceable pathways within a cell. Despite, or perhaps because of, evolutionary pressure, cells do not travel light. They appear to carry with them, despite the energy penalty, much material that is left unused. Parallel architectures are an emblematic example of this strategy: the design avoids critical vulnerabilities but at the price of supporting structures that may never be used. The emphasis, in non-biological systems, on the leanest meanest approach may not, in the longest of long runs, necessarily be the most effective strategy. Programmes that are prized for their apparent parsimony may not, ultimately, be the best solution.

Adaptability

A structure or pathway that is usually used for one purpose can relatively easily be used to accomplish a completely different task. Tanaka and colleagues have introduced the concept of “Compound Control” which they define as follows: spatial and temporal variations in signalling down a series of relatively simple routes (transduction pathways or neurons) can produce a huge variety of responses to a wide variety of changing environmental conditions (Tanaka, Okano et al. 2006).

Error trapping

Each time a cell divides, the genetic code is transcribed with a remarkable degree of fidelity, there is, approximately, only one unforced error per 107 to 108 bases copied (Kunkel 2004). There is an

elaborate proof-reading and correction system that is, in evolutionary terms, highly conserved. This emphasises another aspect of biological organisation: the combination of a high degree of plasticity with a high degree of conservatism.

Ruthless elimination of the unsuitable or unfit

This applies to far more than evolution itself and extends far beyond the concept of nature red in tooth and claw. Cells that are regarded as "other" are eliminated by immunological attack, and by phagocytosis. Cells that are in the wrong place at the wrong time, or whose time is simply up, or who have suffered critical degrees of damage can destroy themselves, a process known as by a specialised form of self inflicted death - apoptosis. Anoikis is a specific form of apoptosis that occurs when cells find themselves without a supporting milieu—classically when they detach from the supportive matrix upon which they have been growing.

The continual generation of errors

Although the genetic code is copied with great fidelity errors, nevertheless, occur. Most of these error are deleterious and, through the pressure of natural selection, eliminated. Some errors are fruitful, in that they improve the extent to which an organism is adapted, or is able to adapt, to its environment. Natural selection encourages the persistence through the generations of these fruitful errors. They key point is this: for evolution to occur by means of natural selection, there have to be errors in the first place. We call these errors mutations and so conclude that mutations provide the variation upon which evolution, through the mechanism of natural selection, can act.

Stigmergy

This concept embodies the notion that entities (creatures, molecules) can produce structures, architectural changes, that influence subsequent behaviour (Ramos and Merelo 2002). The classic example is the building of a termites nest. These nests are built by independently acting creatures with no ability to communicate directly with each other. And yet the building of a termites' nest looks, to the casual observer, as if it were a co-ordinated cooperative process. The co-ordination is dictated by the architecture: pheromones[6] are deposited with each ball of mud conveyed by each termite, the presence and concentration levels of the pheromones within the structure influences the behaviour of termites in the vicinity and through this mechanism chaos acquires purpose. The assembly, and subsequent influence of, microtubules within mammalian cells is another example of stigmergy at work (Tabony 2006).

Architecture both constrains and facilitates

Stigmergy is a specific example of how form can influence biological behaviour. The internal architecture of a mammalian cell provides abundant examples of the interaction between form and function. The mitochondrion[7] is the intracellular structure that provides cells with energy. The efficiency of biochemical processes within a mitochondrion are influenced by the architecture of the mitochondrion (McBride, Neuspiel et al. 2006). Architecture can also have a prohibitive influence. P53 acts, in part, as a transcription factor. Transcription factors act upon DNA within the cell's nucleus. If p53 is actively excluded from the nucleus then it cannot, by virtue of architectural exclusion, act as a transcription factor. This mechanism may account for the effective absence of p53 activity, despite normal p53 structure, during the development of some tumours (Lu, Pochampally et al. 2000). Nuclear exclusion may also influence the relationship between mdm2 and p53, with low levels of mdm2 causing p53 to

[6] A pheromone is a chemical secreted by one animal that influences the behaviour of another animal

[7] An interesting evolutionary quirk is that mitochondria may have originated from bacteria that had been engulfed by mammalian cells. The arrangement suits both parties and has persisted over millennia: the cell gains energy from the mitochondrion; the mitochondrion gains protection from the cell.

be excluded from the nucleus (Shmueli and Oren 2004). Similar architectural influences may apply to the Sir2 system of transcriptional regulation (Wilson, Le et al. 2006).

Biological effects are not just about means and motive, they are also about opportunity. Consider a mixed-sex boarding school for teenagers. There is plenty of reproductive potential but if the girls are kept separate from the boys then that potential, no matter how abundant, will not be realised.

Positive and Negative Feedback

Feedback loops, at all levels of organisation, are a feature of biological systems. The importance of maintaining the internal environment ("milieu interieur") was emphasised by Claude Bernard in 1865[8]. Out of this grew the concept of homeostasis, formulated by Walter Cannon in 1932. Homeostasis implies that a system maintains internal constancy, despite external perturbations, by means of processes that ensure a dynamic equilibrium. These processes may include, but do not exclusively involve, positive and negative feedback: as such homeostasis is similar to the concept of cybernetics—familiar to engineers and programmers. The endocrine system is a classic example of feedback and control: the body samples its own milieu, if a problem is identified (excess salt level, for example) then a hormone is released (in this case ADH from the hypothalamus) which acts to correct the problem in this case on the tubules of the kidney so that water is retained and the salt concentration is lowered. As the salt concentration falls, ADH production is switched off, until the next time.

More local loops also exist. A cell can produce messenger chemicals that act externally on the cell itself (autocrine control) or within the cell (intracrine effects) or on its immediate neighbours (paracrine control).

Contact inhibition

Cells growing in culture exhibit the property of contact inhibition. When each cell is in direct contact, on all sides, with other cells the culture ceases to divide. It is as if the cells "know" that they have reached the limiting capacity of the environment to support them and that further division would lead only to hardship and toil. Loss of contact inhibition is one of the behavioural hallmarks of malignant cells growing in culture.

The importance of the substrate

The physical and chemical properties of the medium upon which cells grow, or attempt to grow, can have important functional effects on cells at both an individual and at a population level. The environment can be either sustaining and permissive or hostile and unsupportive.

The cell as integrator

We can flummox ourselves with our attempts to assemble models of multiple pathways and the myriad interactions between them. Alternatively we can use cell-based systems in which the cells themselves perform integration and synthesis for us. We provide inputs, let the cells do the things they do best, and assess outputs. Who needs systems biology when there are cells to do the work for us?

III. Software Agent Ecosystems

We will now define the SAE [2], first recounting the process of biomimicry that led to its creation and design. We considered the fields of theoretical ecology and evolutionary theory to create a software ecosystem, with at least some of the properties of biological ecosystems. Arguably the most fundamental differences between biological and software ecosystems lie in the motivation and approach of researchers. Biological ecosystems are ubiquitous natural phenomena, whose maintenance is crucial to our survival. The performance of natural ecosystems is often measured in terms of their stability,

[8]"La fixit du milieu intrieur est la condition d'une vie libre et indpendante."

complexity and diversity. In contrast, software ecosystems as defined here are technology engineered to serve specific human purposes. In many cases, the purpose of a software ecosystem is to solve dynamic problems in parallel with high efficiency. This criterion may be related only indirectly to complexity, diversity and stability.

By comparing and contrasting theoretical ecology with the anticipated requirements of Software Ecosystems, we have examined how ecological features may emerge in some systems designed for adaptive problem solving. Specifically, we suggested that a Software Ecosystem, like a real ecosystem, will usually consist of self-replicating agents that interact both with one another and with an external environment. Agent population dynamics and evolution, spatial and network interactions, and complex dynamic fitness landscapes will all influence the behaviour of these systems. Many of these properties can be understood via well-known ecological models [3], [4].

A further body of theory treats ecosystems as complex adaptive systems [5]. These models provide a theoretical basis for the occurrence of self-organisation in both digital and real ecosystems. Self-organisation results when interactions among agents and their environment give rise to complex non-linear behaviour. It is this property that provides the underlying potential for scalable problem-solving in a digital environment.

We view a software ecosystem to be the digital counterpart of a natural ecosystem [2], [6], as it is a software infrastructure for supporting a large number of software services [7], [8], [9]. A two-level optimisation scheme inspired by natural ecosystems, in which a decentralised peer-to-peer network forms an underlying tier of distributed services. These services then feed a second optimisation level based on an evolutionary algorithm that operates locally on single habitats (peers), aiming to find solutions that satisfy locally relevant constraints. The local search is sped up through this two-fold process, providing better local optima as the distributed optimisation provides prior sampling of the search space by making use of computations already performed in other peers with similar constraints [7], [10], [11]. The services consist of an executable component and a descriptive semantic component. This is analogous to the way in which an agent is capable of both execution and having an ontological description. If the services are modelled as software agents, then the software ecosystem can be considered a Multi-Agent System (MAS) which uses distributed evolutionary computing to combine suitable agents to meet user requests for applications.

The users of our SAE create queries to the system by creating a request, specifying a semantic description of the agents or services they require. This description will define the metrics for evaluating the fitness for a composition of agents as a distance function between the request and the agents' descriptions. A successfully used agent, or composition of agents, can then migrate from one peer (habitat) to another, becoming hosted at habitats where it is useful in satisfying user requests.

The connectivity between habitats, in biological ecosystems, is defined by the geography. The SAE does not have a default geographical topology to define the connectivity of the nodes. Instead, a re-configurable network topology is used, which is dynamically adapted based on the observed migration paths of the individuals within the habitat network of the software ecosystem. Analogously to Hebbian learning, the habitats which do not exchange individuals will become less strongly connected, and the habitats which exchange individuals more often attain stronger connections. This leads to a network topology that is discovered over time, resulting in a network that resembles the connectivity of the businesses within the user base, typically a small-world network for Small & Medium Enterprises (SMEs) [12], [13], [14]. Such a network has many strongly connected clusters called *sub-networks* (quasi complete graphs) and a few connections between these clusters. Graphs with this topology have

a very high clustering coefficient and small characteristic path lengths [15], [16], as shown in Figure 3.

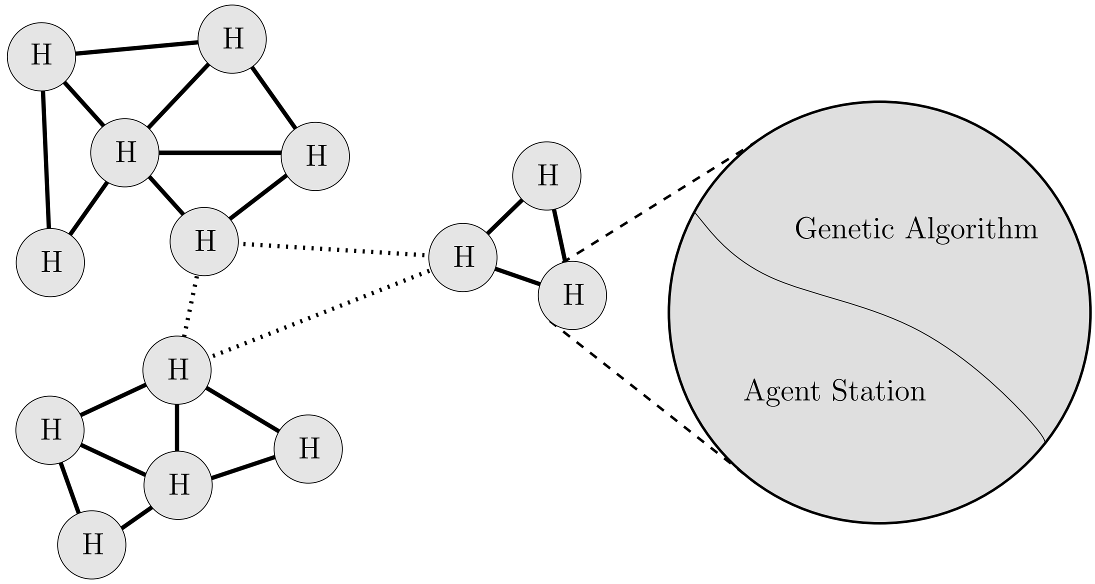


Fig. 3. Software Agent Ecosystem: optimisation architecture in which agents travel along the peer-to-peer connections that have the topology of a small-world network; in every node (habitat) local optimisation is performed through an evolutionary algorithm, where the search space is determined by the agents present at the node.

An evolving population is instantiated in response to a user request, using an evolutionary optimisation technique to generate the optimal combination of available agents that fulfil the user request. The population is seeded from the agents available (agent-pool) at the habitat in which the evolving population is instantiated. This allows the evolutionary optimisation to be bootstrapped by previous solutions stored. The evolving population of agents will then search the available agent combination space through evolution to find the optimal solution(s) to a user request. The fitness of individual agent-sets within a population is determined by a selection pressure applied as a *fitness function* instantiated from the user request, and works primarily by comparing the semantic descriptions of the agents with the semantic description in the user request. An agent's semantic description acts as a guarantee of its functionality, and is the inheritable component from one generation to the next in the evolutionary optimisation. Mutations can occur by switching agents in and out of the agent-set structure. Recombination (Crossover) can occur by combining elements of two agent-sets into a new agent-set. The optimal solution (agent-set) found can then be migrated through the interconnected habitats, recombining with other agents to meet more user requests in other evolving populations, helping to fulfil other user requests.

Distributed evolutionary computing utilises *parallel processing* to solve a particular request (problem) [17]. Examples include the near natural *Island Model* [18], in which a distance is set between the sub-populations on each island, including a probability of migration between one *island* and another. This approach has been used effectively in the determination of investment strategies [19]. The SAE uses a similar approach, but to solve several *similar* requests simultaneously. In our SAE different requests are evaluated on separate *islands* within a habitat, with an island being an evolving population of composite agents. So, adaptation is accelerated by sharing solutions with other islands evolving solutions to similar requests (problems).

The experimental results indicate that under simulation conditions the SAE behaves in some ways like a natural ecosystem, and at large scales outperforms the comparison system based on a ‘traditional’ Service-Oriented Architecture (SOA)[9]. These findings support the general idea that the self-organising properties of software ecosystems can assist in generating scalable solutions to complex dynamic problems. This suggests that incorporating ideas from theoretical ecology can contribute to useful self-organising properties in software ecosystems. [2]

A. Distributed Intelligence

A form of stigmergy, the *distributed intelligence*, was created with the intention of optimising the evolutionary dynamics within the SAE, after the consideration of several alternatives [21]. The *distributed intelligence* should assist our SAE to provide better solutions than it could alone, especially in the short term and to rapidly changing conditions, by positively affecting the evolutionary dynamics via the ecological dynamics. The targeted migration of the *distributed intelligence* will achieve this by targeting agent migration based on similar agents sharing their respective migration and usage histories. This process is visualised in Figure 4.

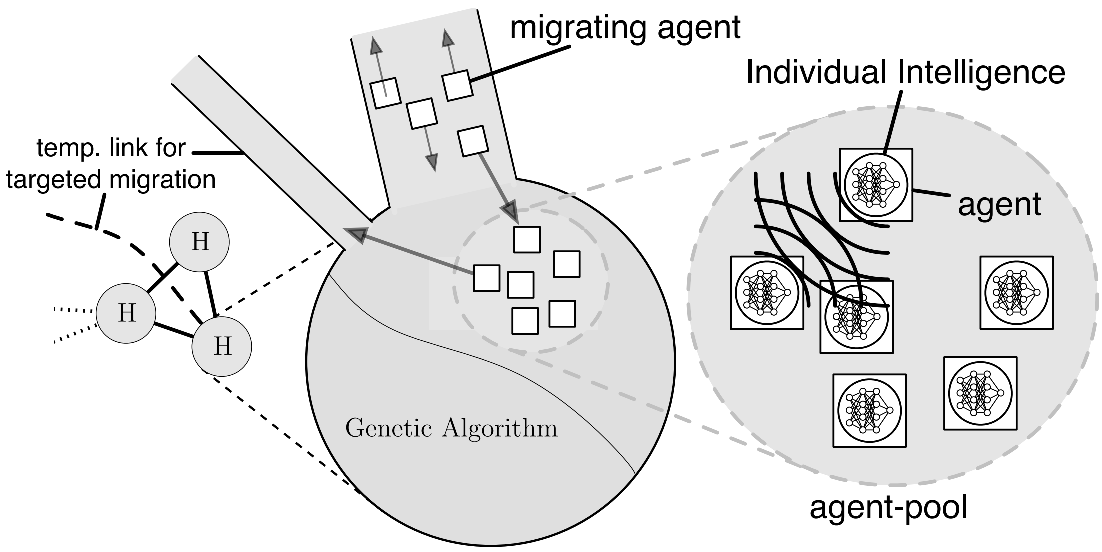


Fig. 4. Distributed Intelligence: augmentation in which an agent arriving at a habitat interacts with the other agents, sharing their respective migration and usage histories for targeted migration when they find one another to be similar.

The *distributed intelligence* will complement the evolutionary optimisation indirectly. An agent, when finding similarity with another agent based upon their descriptions, will be able to migrate. It will target its migration to a habitat where it will have the opportunity to be useful, based upon the migration and usage history of the similar agent. In biological terms, it is equivalent to providing each individual with a simple social interaction mechanism. It should allow niches to be fulfilled faster, so that the habitat clusters will reach their *climax communities* sooner, thereby significantly improving the speed

[9]Although a non-evolutionary Service-Oriented Architecture sounds more ‘traditional’, we should highlight that SOA was defined relatively recently [20], and the full realisation of its principles, for example loose coupling of services, has not been achieved yet. This work is being pursued in OPAALS by the UniS team in WP3, through an agent architecture for local coordinators that we hope can be integrated with the concepts discussed here

of the *succession* process. Also, communities will be able to adapt faster to changing conditions.

As the SAE increases in size (number of users), the number of agents (services) available globally will become increasingly large. For the evolutionary optimisation to continue to work efficiently as the SAE expands, optimal subsets of the power-set of agents available globally will be required at the habitats. The migration probabilities between the habitats will work to achieve this in a *passive* manner. Although the mechanism is effective, the migration of agents will be dilatory and therefore not strongly directed. It allows the agents, based primarily upon success at their current location, to spread in the correct general direction within the habitat network. The *distributed intelligence* will work in a more *active* manner allowing the agents immediate highly targeted migration to specific habitats, independent of success at the current location, instead of the generally directed migration only after success at the current location. This will help to optimise the subset of possible agent-sets found at the habitats, which are then used in the evolutionary optimisation processes to find applications (combinations of agents) to user requests.

The agents can interact with one another, using their embedded *individual intelligence* components, to determine functional similarity based on their semantic descriptions. This involves comparing their semantic descriptions to one another for similarity, and so is independent of the current habitat's context (user requests). If agents find similarity through the inter-agent comparison, then they can share their migration and usage histories to determine habitats where they could be valuable. This interaction would occur outside the evolutionary optimisation, and the consequence of this interaction is in the additional targeted migration of agents, which can occur between connected and unconnected habitats.

When targeted migration occurs between connected habitats it will speed up the natural migration of agents. When it occurs between unconnected habitats, it will help the habitat network to adapt to changing conditions faster than it otherwise would. In effect, during targeted migration the *distributed intelligence* short-circuits the small-world topology [2], to avoid being unnecessarily slowed by it. This may seem counter-productive after creating such a topology, but the small-world topology is what allows the habitat network to specialise solutions to specific users and specific user requests. This ability leads to a weakness, specifically that the habitat network can be slow to adapt to changing conditions, which the *distributed intelligence* can potentially address. The *distributed intelligence* will help catalyse the formation of clusters, and the assignment of nodes to the correct clusters, within the small-world topology. The desired effects of the *distributed intelligence* on the habitat network will be achieved by integrating the targeted migration with the existing migration feedback mechanisms.

The *distributed intelligence* will interact with the ecosystem dynamics of the SAE. Embedding the *individual intelligence* components within the agents, not only maintains the consistency of the Agent-Based Model (ABM) model of the SAE, but also strengthens the *agent* definition by giving the individuals (agents) some intelligence and control over their existence.

We first considered NNs for the learning-based pattern recognition of the *distributed intelligence*, because it is an established and widely applicable pattern recognition technique. The pattern recognition capabilities of NNs can be leveraged to allow agents to determine similarity based on their descriptions, using multilayer feed-forward NNs and the backpropagation training technique to specify the required pattern recognition behaviour.

We then considered Support Vector Machines (SVM), which is a newer technique and theoretically more suitable for the *distributed intelligence*. SVM are a learning system based on recent advances in statistical learning theory, and is becoming established as a tool for machine learning. SVM are a set

of related supervised learning methods used for classification and regression. They belong to a family of generalised linear classifiers. They can also be considered a special case of Tikhonov regularisation. A special property of SVM is that they simultaneously minimise the empirical classification error and maximise the geometric margin. Hence, it is also known as the maximum margin classifier [22].

The experimental results indicate that under simulation conditions the *distributed intelligence* constructively interacts with the ecological and evolutionary dynamics of our SAE. An effective pattern recognition technique is required for the *distributed intelligence* to operate effectively, and both NNs and SVM proved to be effective, with SVM proving to be marginally more effective than NN. The results also indicated that the *intelligence* created the improvement in the SAE, and not the additional migration created to enable the *distributed intelligence* to operate. Based on our theoretical understanding and the experimental results we would recommend SVM over NN for the learning based pattern recognition functionality of the *distributed intelligence* of the SAE.

IV. Completing the EOA of the SAE

Experimental results have indicated that under simulation conditions the SAE behaves only in some ways like a natural ecosystem. Further experimental results have indicated that under simulation conditions the application of our form stigmergy constructively interacts with the ecological and evolutionary dynamics of our SAE. However, there is still a lack of autopoiesis, shown by the inability of SAE to apply evolutionary optimisation to a *granularity* lower than that of the agent, i.e. the object and method level. A more autopoietic form of computing is required to enable this, and we are currently focusing on intracellular behaviour for our interactive model of computing, because cells are fundamental to the autopoietic behaviour inherent in life. Cells are the basic unit for the construction of all life and operate almost entirely through the process of gene expression. Cells are the biological construct that most obviously shows autopoietic behaviour, which is a process that works similarly at multiple levels of scale, and so an understanding of cellular operations is a critical first step in providing a tangible understanding of autopoiesis for interaction computing.

A. Framework for Computational Biomimicry

We will formalise the biomimcry design process in software engineering (as shown in Figure 1) through the use of a software engineering technique called *design patterns* to create a Framework for Computational Biomimicry (FCB).

"In software engineering, a design pattern is a general repeatable solution to a commonly occurring problem in software design. A design pattern is not a finished design that can be transformed directly into code. It is a description or template for how to solve a problem that can be used in many different situations. Object-oriented design patterns typically show relationships and interactions between classes or objects, without specifying the final application classes or objects that are involved. Algorithms are not thought of as design patterns, since they solve computational problems rather than design problems."[10]

Extending this concept, Biological Design Patterns (BDPs) will catalogue common interactions between biological structures using a Pattern-Oriented Modelling (POM) approach, which is meant to provide autopoiesis. These BDPs could eventually be applied to our software ecosystem to endow it with the same self-organising capabilities found in biological ecosystems, and one of the ways in which this will be evident will be the ability to apply the evolutionary process at a lower level of granularity than previously possible, i.e. the object and method level, instead of the service level as

[10]http://en.wikipedia.org/wiki/Design_pattern_(computer_science)

is currently done. To this end we will develop a modelling framework based on the Systems Biology Modelling Language (SBML), which utilises a domain modelling methodology based on UML to represent biochemical pathways.

"This UML-based definition in turn is used to define an XML Schema (Fallside, 2000; Thompson et al., 2000; Biron and Malhotra, 2000) for SBML. There are three main advantages to using UML as a basis for defining SBML data structures. First, compared to using other notations or a programming language, the UML visual representations are generally easier to grasp by readers who are not computer scientists. Second, the visual notation is implementation-neutral: the defined structures can be encoded in any concrete implementation language—not just XML, but C, Java and other languages as well. Third, UML is a de facto industry standard that is documented in many sources. Readers are therefore more likely to be familiar with it than other notations."[11]

This framework includes a translator from the specified chemical reactions to the corresponding differential equations for the time evolution of the concentrations of the reactants and products. Differential equation models can then be investigated in *Mathematica* and other similar packages. Whereas we hope that this approach will help UNIVDUN's research in the modelling of cell regulatory cycles, the benefit to software engineering arises mainly from the use of a domain modelling methodology that is based on UML; in other words, strengthening the formal and semi-formal language links between biology and software engineering.

B. Autopoiesis and Life

Life, cells and multicellular organisms have evolved over hundreds of millions of years. Through random variation, trial and error, and the pressure of natural selection, effective solutions have emerged to many generic problems. There is in effect a wealth of hard-won practical experience embedded within the command and control systems that govern and sustain cells and tissues. The problems that cells have solved include: how to survive in a competitive environment when resources are scarce; how to respond to damage; how to respond to specific forms of stress; how to respond to non-specific forms of stress; how to maintain integrity at the expense of sacrifice, and so on. Many of these solutions, honed by the millennia, can be expressed in mathematical or computational terms. Mediated by mathematics and logic, and their relationship to the construction of algorithms and computer programmes, these highly practical solutions can be made widely available for use outwith living systems.

Biology can also help us derive architectural principles that can help us address scale-dependent questions of resource distribution, adaptation time scales relative to signal propagation speed, logistics, and wider economics questions. In 1928, the biologist and geneticist JBS Haldane wrote a short article entitled "On Being the Right Size" (Haldane 1928). He pointed out the constraints that the laws of physics imposed upon biological development: heavy creatures need strong legs; insects may defy gravity, but are the victims of surface tension. He then drew, typically, a parallel between biology and politics: he pointed out that socialism might work well for smaller countries, but was probably untenable for what was then the rather large British Empire. This principle can also apply to social and computer networks: there is an optimum size that is determined by permissions and constraints residing outwith the system itself. Growth is part of life and life can teach us lessons on how to grow safely and effectively.

Growth is not, however, the only biological factor that we need to consider. We need also to think about reproduction (making copies) about communication (information processing and transfer) and

[11] http://sbml.org/specifications/sbml-level-2/version-1/html/sbml-level-2.html

about development (ontogeny). Life has important lessons to teach us here as well.

1) What is Life: a pre-autopoietic perspective: Biology is concerned with the study of living things. Which begs the question - how do we define life? The simplicity of this question is deceptive. We all know what life is but, when we are asked to put what we know into words, we flounder. Is fire alive[12]?

Property associated with life	**Possessed by Fire**
Metabolism	Yes, it converts fuel and oxygen into carbon dioxide and water
Ability to multiply	Yes, sparks from one fire can start other fires
Ability to move	Yes, a fire can spread through a forest
Adapt to circumstances	Yes, a fire can smoulder when fuel and oxygen are scarce only to flare up again when the supply is replenished
Arise spontaneously	Yes, a fire does not have to be lit: consider the ferocious chemical reaction between sodium and water.

So, in order satisfactorily to define life, we need to go beyond the simple attributes presented in the table—otherwise we would have to consider fire as a living thing. We are left with our original question—what is life?

Erwin Schrödinger, a physicist[13], famously asked the question in his 1944 book entitled, logically enough, "What is Life?": the most influential book in the history of molecular biology. He pointed out that, in contrast to the large number systems of physics, in which events were governed by statistical probabilities assessed in vast numbers of particles, biological problems involved the theory of small numbers. A single molecule could profoundly influence cellular events—in essence he anticipated DNA and the genetic code. He also believed that in contrast to the second law of thermodynamics (order leads to disorder, entropy increases in the purely physical world) life, and biological systems, are characterised by creating order from disorder—by decreasing the amount of entropy or randomness in localised systems that we call living organisms (although the overall entropy in the world does keep increasing.[14] By this definition biology might become the study of self-replicating entities that can produce order from chaos. Since fire produces disorder it would not, by this definition, be alive. A more recent formulation of Schrdinger's answer would include the concepts of multiplication, variation and heredity. The ability to multiply implies that an entity can make copies of itself; variation implies that the copies are not all perfect replicas of the original and heredity provides a mechanism for copying that is not, through mutation, always entirely true to the original. This line of argument leads inexorably to the dominant intellectual idea in modern biology—the concept of evolution and, more particularly, the concept of evolution through natural selection.

[12]I have taken and adapted this question from Maynard Smith and Szathmary "The Origins of Life " p 4-6 Oxford (1999)

[13]Erwin Schrödinger (1887-1961) Awarded the Nobel Prize for Physics in 1933 for his 1926 discovery of the wave equations in the description of atomic spectra. he was unhappy with the Bohr orbital model and sought to use eigenvalues to explain atomic spectra. He shared his prize will Paul Dirac. Bavarian by birth, he ended his days in Dublin.

[14]A fairly extensive discussion of entropy and the construction of order in the context of Digital Ecosystems can be found in [23]

There is little indication, from the published writings of Maturana and Varela that they were particularly interested in, or indeed, particularly aware of any of the foregoing. Their writings are characterised by an apparent ignorance of history: which may have led them, on occasion, into the problem that was best expressed by George Santayana in 1906: “Those who cannot remember the past are condemned to fulfil it”, sometimes rendered as “those who do not learn from the mistakes of history are condemned to repeat them”. It is quite clear that Maturana and Varela regarded their definition of autopoiesis as defining life itself: “Our proposition is that living beings are characterized in that, literally, they are continually self-producing. We indicate this process when we call the organization that defines them an autopoietic organization.” (The Tree of Knowledge by HR Maturana and FJ Varela, Shambhala, 1987). They point out that an organism or cell not only produces its own internal components but, crucially, also produces components, such as the cell membrane, that serve to demarcate it from its environment. Perhaps their key observation is this one: if we define entities in terms of their organisation, then there may be several structural means to the same end. The biological origins and further developments of the concept of autopoiesis will be investigated (currently in preparation).

C. Gene Expression: The p53 Pathway as An Exemplar

The p53[15] protein, discovered in 1979, is associated with a viral oncoprotein in transformed cells [24], [25]. Initially it was classed as an oncogene[16]. Ten years later *p53* was correctly reclassified as a tumour suppressor. This change in view arose as a result of several key findings: the observation that a diverse range of tumours contained mutations in *p53* [26], [27]; the demonstration that wild-type[17] p53 can suppress malignant transformation [28], [29]; p53 knockout[18] mice were shown to have a high propensity to develop spontaneous tumours [30]; the Li-Fraumeni syndrome[19] is associated with a germline mutation in the *p53* gene [31]. p53 function is altered in almost all cancers. Approximately 50% of solid tumours occurring in adults contain mutations in p53. In the majority of the other cancers, where wild-type p53 persists, the evidence suggests that its activity is mis-regulated by other factors, such as human papillomavirus E6 protein [32].

Quite apart from its importance as a tumour suppressor, recent literature demonstrates that p53 is implicated in the ageing process. Higher than normal levels of regulated p53 expression increase average lifespan in mice [33]. This finding, together with p53’s value as a target for cancer therapy demonstrates that p53 is a molecule of fundamental biological importance.

p53 is a stress inducible transcription factor that can positively or negatively regulate the expression of numerous genes. It is activated in response to a wide variety of cellular stresses including DNA damage, oncogene activation, hypoxia and factors that provoke aberrant chromosome distribution during mitosis. The outcome of p53’s transcriptional programme in response to stress prevents the survival of cells with damaged DNA or cells with an abnormal chromosome content. p53 exerts this protective effect through the induction of either cell cycle arrest or programmed cell death (apoptosis) in damaged/stressed cells [34]. The decision between the alternative fates i.e. the decision on whether p53 activates the expression of genes that lead to cell cycle arrest (e.g. p21 WAF1/Cip1) or whether it activates genes that induce the apoptotic machinery (e.g. PUMA) is thought to be exquisitely modulated by a panoply of co-factors and by post-translational[20] modifications on p53 (phosphorylation, acetylation etc) [34].

[15]P for protein, 53 because its molecular weight is 53kDa

[16]Oncogene - a gene sequence, the expression of which is associated with malignant change

[17]Wild-type - the normal (as opposed to mutated) gene product

[18]Knockout - a genetically modified lab animal in which a particular gene has been deleted

[19]Li-Fraumeni Syndrome - an inherited condition in which there is a high incidence of malignancy. Typically, breast cancer; soft tissue sarcomas; osteosarcomas; brain tumours; adrenocortical carcinomas; leukaemia.

[20]Chemical and physical changes in a gene product after it has been synthesised.

Over-expression of p53 is potentially catastrophic, it could prevent appropriate cell division of a cell or cause its immediate death. Its levels and activity in cells proliferating under normal conditions must be kept low. Although there is some evidence suggesting that there is some regulation of production of p53, the current view is that the amount of p53 in cells is primarily modulated at the level of inactivation and degradation. The active conformation of p53 is very unstable [35]: the molecule is effectively targeted for degradation and consequently has a short half-life, approximately 20 minutes [32]. The short half-life of p53 is due to its negative regulator mdm2 which catalyses the conjugation of ubiquitin[21] molecules to p53 [36], [37], [38]. As mdm2 is a p53 responsive gene, this creates a classic negative feedback loop by which activation of p53 causes the increase in the levels of its own negative regulator. Additionally, mdm2 can also inhibit p53 by binding to p53's DNA binding domain, and hence prevent p53 from functioning as a transcription factor [39]. The importance of mdm2 as a negative regulator of p53 is highlighted by the fact that mdm2 knockout mice die as embryos. This lethal effect can be reversed by knocking out p53 as well as the mdm2 [40].

Cellular stresses, such as DNA damage, lead to modifications in the p53 protein, typically acetylation or phosphorylation, that allows it to evade its interaction with mdm2 and/or mdm2-mediated degradation [41]. p53 also responds to oncogenic stress: the perturbations caused by the inappropriate and sustained production of factors promoting cellular proliferation. This results in a stress-induced increase in the steady state levels of p53 and, possibly, an increase in its affinity for DNA causes the transcription of genes promoting cell cycle arrest and/or apoptosis. Since normal cells have active cell cycle check-points, the response to stress, at least below certain thresholds, tends to be cytostatic[22] in normal cells. This halt allows cells to repair their damage (i.e. through activation of the DNA repair pathways) before entering the DNA synthesis phase (S phase) of the cell cycle or mitosis: the phenomenon of rest-and-repair. Once the stress is over, "repaired" cells reduce their p53 activity and are able to re-enter the cell cycle and proliferate normally. The ability of p53 to increase the expression of mdm2 is amongst the mechanisms that contribute to this recovery.

In tumours where the function of p53 is lost through mutation, cells are unlikely to be detained before entry into the S phase or mitosis. Sub-lethally damaged cells will be able to survive, thus propagating their defects, and adding to the problem of genomic instability in tumours. Only severely damaged cells will die immediately. Tumour cells where p53 is not mutated, but where its function is excessively diminished, tend to respond in a mixed way and some may die whilst a proportion will be able to rest and repair.

p53 is a tempting target for cancer therapy: by *fixing* mutant p53, replacing it in tumours where it has been deleted, or stimulating its apoptotic function in tumours where p53 is wild type but suppressed [42], it might be possible to curb the growth of a tumour or even eradicate it completely. Strategies include the discovery of agents that may stabilise active conformations in mutant p53 [35] or the delivery of genetically modified vectors (e.g. adenovirus) that express p53 [43]. The discovery of small molecules that specifically inhibit the interaction between p53 and mdm2 (e.g. the nutlins) [44] is an attractive method of enhancing the tumour suppressor function of p53 in those cancers where p53 is inactivated by mechanisms other than mutation.

When we study the p53/mdm2 control system we are looking at a system that is both biologically fundamental and mathematically tractable. Although the system has many collateral connections, any

[21]Ubiquitins - molecular chaperones that escort their target molecules to the proteasome, a garbage disposal system that destroys functional proteins.

[22]Cytostatic - causes cells to cease dividing but does not alter their long-term viability

one of which can influence its overall performance, it is, in itself, fairly simple and we can measure experimentally how levels of one component will affect not just the levels of the other components but also the behaviour of the system as a whole. It is likely that, when we study p53 and mdm2, we are studying an archetypical system. In other biological control systems the names of the players may change but their relative roles and influences may be very similar to those that obtain within the p53/mdm2 control system.

It is unsurprising, therefore, that the p53/mdm2 has been extensively studied and modelled. The p53/mdm2 system can, at its simplest, be regarded as a simple problem in homeostasis based on negative feedback [45]. Reality is more complicated than this and more sophisticated approaches are required. Time is an important factor: there is an inbuilt asymmetry in the p53/mdm2 interaction. mdm2-mediated degradation is a rapid process, ubiquitination is fast and simply involves protein-protein interaction. The p53-mediated production of mdm2 is, by contrast, a slower process involving transcription of DNA to RNA and subsequent synthesis of new protein.

The mathematical breadth of the approaches brought to bear on the analysis of p53/mdm2 interactions has been impressive: some modellers have already moved beyond differential equations and into areas such as the mathematics of bistability [46], [47] and stochastic resonance [48], [49], [50]. Models predicated on oscillatory behaviour have been particularly popular [51], [52], [53], [54], [55], [56], [57], [58], [59], [60]. One set of provocative, but uncorroborated, observations suggest that p53 may operate as a digital signal: the p53 response to irradiation, it is claimed, occurs as series of pulses of relatively fixed frequency but varying amplitude [54]. There is the suggestion, from analyses based on data from single cells, that radiation-induced DNA damage will bring about stable, undamped, oscillations in p53 and mdm2 levels in irradiated cells and that, with increasing radiation dose, the proportion of such cells, within the irradiated population, increases [54].

D. p53 Behaviour Summary

We all have the ability to produce p53, we have the code for p53 within every cell of our body. Potentially each of our 10^{13} cells could at any time produce active p53. Why would this matter? It would matter because p53 is, if you are a cell (and we are all made of cells), a very dangerous substance indeed. Here is what it can do.

It can activate a set of cellular programmes that instruct a cell to destroy itself immediately. The process is called apoptosis (from the Greek for leaf fall).

It can instruct a cell that was due to divide not to do so until it is fit to do so. This mechanism will apply to cells that are damaged (by drugs, or radiation or oxidative stress for example). It will allow time for rest and repair and, in evolutionary terms, makes perfect sense. Cells with mutations (faulty code) are forced to repair the mutation before they are permitted to divide, mutations are therefore much less likely to be propagated through the generations.

It can cause a cell to lose some of its reproductive capacity. A mammalian cell can undergo up to 50 or 60 divisions but thereafter enters the state of reproductive senescence. It can exist and metabolise quite satisfactorily but is unable to reproduce itself. Physiologically it is alive; reproductively it is dead. P53 has the ability to induce a cell to enter this state prematurely. A cell may be allowed to divide but only two or three more times, falling well short of its normal reproductive potential.

In essence p53 instructs cells to shape up (rest and repair) or ship out, either immediately (through apoptosis) or a little later (through premature reproductive senescence).

So if we were to take an analogy to computer code, what would the p53 code accomplish? It could cause destruction not just of all the code in the computer but also the computer itself. It could limit the expression of some of the code until the damage to the code had been repaired. The biological correlate, if you will, of self-healing software. It can allow faulty code to propagate for a while but then fizzle out leaving no obvious trace of its previous presence.

In summary the unrestrained expression of p53 would be a disaster for individual cells and for the organism as a whole. The existence of p53 is essential for cellular husbandry and for a functional society of cells. The production of p53 is very tightly controlled, for obvious reasons—uncontrolled p53 expression would produce cellular catastrophe. Complete lack of p53 expression is also extremely dangerous. Decreased p53 production is one of the cellular mechanisms that can promote the development of malignant change. So p53 is essential but only in tightly controlled amounts and only at appropriate times. One simple method whereby p53 levels are controlled is through its ability to bring about its own destruction. P53 has a partner protein, mdm2, that specifically targets p53 for destruction via the proteasomal system. Effectively mdm2 sends p53 down the garbage chute. Increased levels of p53 induce increased levels of mdm2 and so p53 brings about, through the action of mdm2, its own demise. This feedback system is ideally suited to the development of modelling techniques that might be of use in approaches to computing that are biologically inspired. The p53/mdm2 relationship is, in fact, extremely complex: the devil is in the detail. The reasons for this complexity may shed light on some of the difficulties that may be encountered when using over-simplified approaches to biologically inspired computing, including our SAE.

E. p53 Modelling

We have painstakingly acquired detailed qualitative and quantitative information on the p53/MDM2 interaction. There is little to be gained by presenting this detailed biological work in this document: the interpretation of hundreds of Western blots is a non-trivial task. Suffice to say that the work has been done and can now be used to inform the formulation of detailed algebraic and mathematical models which, in turn, can be incorporated into mathematically based and Systems Biology Modelling Language (SBML)-based Biologically inspired Design Patterns (BDPs), an example of which is shown in Figure 5

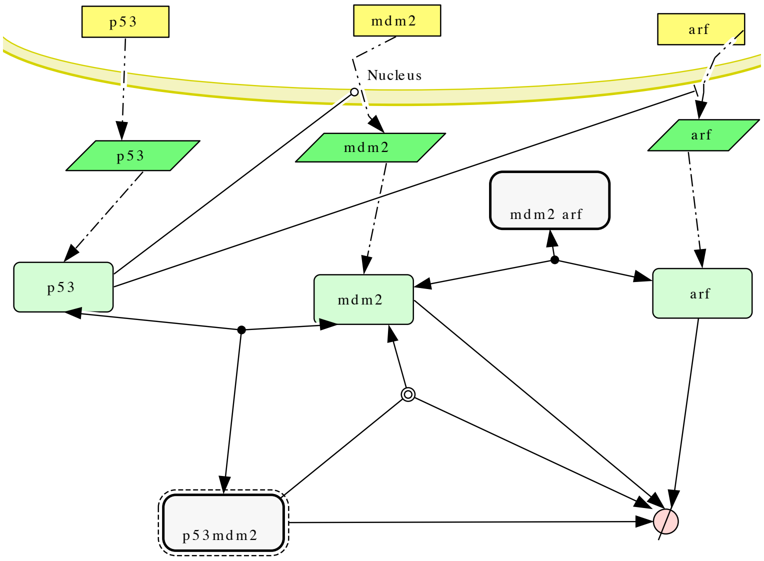


Fig. 5. Visualisation of the p53/mdm2/arf mathematical model, created using Cell Designer [61] from the SBML encoding of the model. The slanted oblong shapes represent mRNA molecules for the production of their respective proteins. The smooth corner oblong shapes represent proteins, with the large one embedding the smaller ones to represent a protein complex. The small circle on the edge of a protein shape, indicates a residue, here phosphorylation. The circles with the diagonal line represent degradation.

The p53/mdm2/arf model was encoded into the SBML, as shown in Figure 5, for distribution and interoperability with a range of simulation tools. The SBML is a machine-readable language, derived from the Extensible Markup Language (XML), for representing models of biochemical reaction networks. SBML can represent metabolic networks, cell-signalling pathways, regulatory networks, and other kinds of systems studied in systems biology. As many new and existing modelling tools have been enhanced to support SBML, it has become virtually ubiquitous for sharing models in the field of systems biology [62].

F. Interaction Computing

We are currently examining some ideas related to the connections between cell biology and software security. This work is being done in collaboration with the BIONETS project (www.bionets.org) where Daniel Schreckling, a researcher in software security from the University of Hamburg, is showing a similar interest to develop a biologically inspired mathematical framework for interaction computing. The bridge that we are in the process of building between these two very different fields relies on abstract algebra and logic and can be simplistically depicted as follows:

cell $->$ biology $->$ algebra $->$ logic $->$ security

In this context security represents one of the possible applications of a formalism that we expect to be of wider relevance. When referring to biologically inspired computing, reliance on some kind of evolutionary framework tends to be assumed by default. Whereas biological evolution does represent an essential model for biologically inspired computing, in this work we are focussing on the 'other' function of DNA. By this we mean all the processes relating to the life of the individual organism, thus a better name could be 'development', or 'morphogenesis', or 'gene expression'.

Our current activities are focussed on understanding the abstract algebra and its connections to non-standard logics. The objective is to reach a mathematical model that can formalise the stable interactive

behaviour of the cell components into an organisationally closed system that represents the archetype autopoietic system (i.e. a stem cell). Because cell biology is fundamentally digital, our hope is that by formalising cell-biological structure and behaviour in this manner we will arrive at the Interaction Machine model of computation as the kernel of digital autopoietic systems.

V. Conclusion

We will continue to learn the process of *gene expression* in autopoietic cellular behaviour for the creation of our interactive computing model, by investigating and modelling p53 using abstract mathematical modelling and SBML-based BDPs within our Framework for Computational Biomimicry.

Acknowledgements

This work is supported by the EU Framework VI Network of Excellence, Open Philosophies for Associative Autopoietic Digital Ecosystems, Contract No 034824 [63].

References

[1] "Digital Business Ecosystems project," *EU Framework 6 Integrated Project*, 2006. [Online]. Available: http://www.digital-ecosystem.org/

[2] G. Briscoe, S. Sadedin, and G. Paperin, "Biology of applied digital ecosystems," in *IEEE First International Conference on Digital Ecosystems and Technologies*, 2007.

[3] R. MacArthur and E. O. Wilson, *The Theory of Island Biogeography*. Princeton University Press, 1967.

[4] S. Hubbell, *The Unified Neutral Theory of Biodiversity and Biogeography*. Princeton: Princeton Univ. Press, 2001.

[5] S. Levin, "Ecosystems and the biosphere as complex adaptivesystems," *Ecosystems*, vol. 1, pp. 431–436, 1998.

[6] G. Briscoe and S. Sadedin, *Digital Business Ecosystems*. European Commission, 2007, pp. 48–55.

[7] G. Briscoe, "Digital ecosystems: Evolving service-orientated architectures," in *IEEE First International Conference on Bio Inspired mOdels of NETwork, Information and Computing Systems (BIONETICS)*, 2006.

[8] G. Briscoe, M. Chli, and M. Vidal, "BOF-0759 Creating a Digital Ecosystem: Service Orientated Architectures with Distributed Evolutionary Computing," in *JavaOne Conference*, 2006. [Online]. Available: https://www28.cplan.com/javaone06_cv_124_1/session_details.jsp?isid=277759&ilocation_id=124-1&ilanguage=english

[9] C. Masuch, "Evolutionary environment network." [Online]. Available: http://evenet.sourceforge.net/

[10] G. Briscoe, "Research blog," *http://www.iis.ee.ic.ac.uk/~g.briscoe/ICL/Blog/Blog.html*, 2007.

[11] ——, "Evolutionary environment architecture requirements," *DBE Project Internal Document*, 2005. [Online]. Available: http://www.iis.ee.ic.ac.uk/~g.briscoe/Eve/EveArchRequirements.pdf

[12] X. Yang, "Chaos in small-world networks," *PHYSICAL REVIEW E*, vol. 63, no. 046206, 2001.

[13] Censis, "Deliverable 27.1: Territorial social capital and driver SMEs," *Digital Business Ecosystems Project*, 2005.

[14] European Commission, "SMEs and co-operation," *Observatory of Europeans SMEs*, 2003.

[15] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.

[16] M. E. J. Newman and D. J. Watts, "Scaling and percolation in the small-world network model," *Physical Review E 60*, vol. 6, pp. 7332–7342, 1999.

[17] E. Alba and J. M. Troya, "A survey of parallel distributed genetic algorithms," *Complexity*, vol. 4, no. 4, pp. 31–52, 1999.

[18] D. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, 1989.

[19] M. Ward, "Life offers lessons for business," *galapagos*, 2004. [Online]. Available: http://news.bbc.co.uk/2/hi/technology/3752725.stm

[20] M. Papazoglou, "Service-oriented computing: Concepts, characteristics and directions," in *Proc. WISE03*, 2003, pp. 3–12.

[21] G. Briscoe and P. De Wilde, "D6.4 intelligence, learning and neural networks in distributed agent systems," *Digital Business Ecosystems Project*, vol. Contract no 507953, 2005. [Online]. Available: whttp://www.iis.ee.ic.ac.uk/~g.briscoe/ICL/Reports_files/D6.4.Intelligence.Learning.and.Neural.Networks.in.Distributed.Agent.Systems.pdf

[22] Y. Tang, B. Jin, Y. Sun, and Y. Zhang;, "Granular support vector machines for medical binary classification problems," in *Proceedings of the 2004 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*, 2004.

[23] P. Dini and E. Berdou, *D18.1-Report on DBE-Specific Use Cases*. www.digital-ecosystem.org, Deliverables: SP5: DBE Project, 2004.

[24] D. Lane and L. Crawford, "T antigen is bound to a host protein in sv40-transformed cells." *Nature*, vol. 278, no. 5701, pp. 261–3, 1979.

[25] D. Linzer and A. Levine, "Characterization of a 54k dalton cellular sv40 tumor antigen present in sv40-transformed cells and uninfected embryonal carcinoma cells." *Cell*, vol. 17, no. 1, pp. 43–52, 1979.

[26] J. Nigro, S. Baker, A. Preisinger, J. Jessup, R. Hostetter, K. Cleary, S. Bigner, N. Davidson, S. Baylin, and P. Devilee, “Mutations in the p53 gene occur in diverse human tumour types.” *Nature*, vol. 342, no. 6250, pp. 705–8, 1989.
[27] S. Baker, E. Fearon, J. Nigro, S. Hamilton, A. Preisinger, J. Jessup, P. vanTuinen, D. Ledbetter, D. Barker, and Y. Nakamura, “Chromosome 17 deletions and p53 gene mutations in colorectal carcinomas.” *Science*, vol. 244, no. 4901, pp. 217–21, 1989.
[28] C. Finlay, P. Hinds, and A. Levine, “The p53 proto-oncogene can act as a suppressor of transformation.” *Cell*, vol. 57, no. 7, pp. 1083–93, 1989.
[29] D. Eliyahu, D. Michalovitz, S. Eliyahu, O. Pinhasi-Kimhi, and M. Oren, “Wild-type p53 can inhibit oncogene-mediated focus formation.” *Proc Natl Acad Sci U S A*, vol. 86, no. 22, pp. 8763–7, 1989.
[30] L. Donehower, M. Harvey, B. Slagle, M. McArthur, C. Montgomery Jr, J. Butel, and A. Bradley, “Mice deficient for p53 are developmentally normal but susceptible to spontaneous tumours.” *Nature*, vol. 356, no. 6366, pp. 215–21, 1992.
[31] D. Malkin, F. Li, L. Strong, J. Fraumeni Jr, C. Nelson, D. Kim, J. Kassel, M. Gryka, F. Bischoff, and M. Tainsky, “Germ line p53 mutations in a familial syndrome of breast cancer, sarcomas, and other neoplasms.” *Science*, vol. 250, no. 4985, pp. 1233–8, 1990.
[32] S. Lain and D. Lane, “Improving cancer therapy by non-genotoxic activation of p53.” *Eur J Cancer*, vol. 39, no. 8, pp. 1053–60, 2003.
[33] A. Matheu, A. Maraver, P. Klatt, I. Flores, I. Garcia-Cao, C. Borras, J. Flores, J. Viña, M. Blasco, and M. Serrano, “Delayed ageing through damage protection by the arf/p53 pathway,” *Nature*, vol. 448, no. 7151, pp. 375–379, 2007.
[34] K. Vousden and X. Lu, “Live or let die: the cell’s response to p53.” *Nat Rev Cancer*, vol. 2, no. 8, pp. 594–604, 2002.
[35] N. Issaeva, A. Friedler, P. Bozko, K. Wiman, A. Fersht, and G. Selivanova, “Rescue of mutants of the tumor suppressor p53 in cancer cells by a designed peptide.” *Proc Natl Acad Sci U S A*, vol. 100, no. 23, pp. 13 303–7, 2003.
[36] Y. Haupt, R. Maya, A. Kazaz, and M. Oren, “Mdm2 promotes the rapid degradation of p53.” *Nature*, vol. 387, no. 6630, pp. 296–9, 1997.
[37] R. Honda, H. Tanaka, and H. Yasuda, “Oncoprotein mdm2 is a ubiquitin ligase e3 for tumor suppressor p53.” *FEBS Lett*, vol. 420, no. 1, pp. 25–7, 1997.
[38] M. Kubbutat, S. Jones, and K. Vousden, “Regulation of p53 stability by mdm2.” *Nature*, vol. 387, no. 6630, pp. 299–303, 1997.
[39] J. Chen, V. Marechal, and A. Levine, “Mapping of the p53 and mdm-2 interaction domains.” *Mol Cell Biol*, vol. 13, no. 7, pp. 4107–14, 1993.
[40] R. Montes de Oca Luna, D. Wagner, and G. Lozano, “Rescue of early embryonic lethality in mdm2-deficient mice by deletion of p53.” *Nature*, vol. 378, no. 6553, pp. 203–6, 1995.
[41] S. Shieh, M. Ikeda, Y. Taya, and C. Prives, “DNA damage-induced phosphorylation of p53 alleviates inhibition by mdm2.” *Cell*, vol. 91, no. 3, pp. 325–34, 1997.
[42] D. Lane and S. Lain, “Therapeutic exploitation of the p53 pathway.” *Trends Mol Med*, vol. 8, no. 4 Suppl, pp. S38–42, 2002.
[43] K. Wiman, “Strategies for therapeutic targeting of the p53 pathway in cancer.” *Cell Death Differ*, 2006.
[44] L. Vassilev, B. Vu, B. Graves, D. Carvajal, F. Podlaski, Z. Filipovic, N. Kong, U. Kammlott, C. Lukacs, C. Klein, N. Fotouhi, and E. Liu, “In vivo activation of the p53 pathway by small-molecule antagonists of mdm2.” *Science*, vol. 303, no. 5659, pp. 844–8, 2004.
[45] S. Harris and A. Levine, “The p53 pathway: positive and negative feedback loops.” *Oncogene*, vol. 24, no. 17, pp. 2899–908, 2005.
[46] V. Chickarmane, B. Kholodenko, and H. Sauro, “Oscillatory dynamics arising from competitive inhibition and multisite phosphorylation.” *J Theor Biol*, 2006.
[47] K. Wee and B. Aguda, “Akt versus p53 in a network of oncogenes and tumor suppressor genes regulating cell survival and death.” *Biophys J*, 2006.
[48] C. Muratov, E. Vanden-Eijnden, and E. Weinan, “Self-induced stochastic resonance in excitable systems,” *Physica D: Nonlinear Phenomena*, vol. 210, no. 3-4, pp. 227–240, 2005.
[49] C. Muratov, E. Vanden-Eijnden, and W. E, “Noise can play an organizing role for the recurrent dynamics in excitable media.” *Proc Natl Acad Sci U S A*, 2007.
[50] J. Tyson, “Another turn for p53.” *Mol Syst Biol*, vol. 2, 2006.
[51] A. Ciliberto, B. Novak, and J. Tyson, “Steady states and oscillations in the p53/mdm2 network.” *Cell Cycle*, vol. 4, no. 3, pp. 488–93, 2005.
[52] B. Ogunnaike, “Elucidating the digital control mechanism for dna damage repair with the p53-mdm2 system: single cell data analysis and ensemble modelling.” *J R Soc Interface*, vol. 3, no. 6, pp. 175–84, 2006.
[53] S. Pigolotti, S. Krishna, and M. Jensen, “Oscillation patterns in negative feedback loops.” *Proc Natl Acad Sci U S A*, 2007.
[54] N. Geva-Zatorsky, N. Rosenfeld, S. Itzkovitz, R. Milo, A. Sigal, E. Dekel, T. Yarnitzky, Y. Liron, P. Polak, G. Lahav, and U. Alon, “Oscillations and variability in the p53 system.” *Mol Syst Biol*, vol. 2, 2006.
[55] D. Hamstra, M. Bhojani, L. Griffin, B. Laxman, B. Ross, and A. Rehemtulla, “Real-time evaluation of p53 oscillatory behavior in vivo using bioluminescent imaging.” *Cancer Res*, vol. 66, no. 15, pp. 7482–9, 2006.
[56] G. Lahav, N. Rosenfeld, A. Sigal, N. Geva-Zatorsky, A. Levine, M. Elowitz, and U. Alon, “Dynamics of the p53-mdm2 feedback loop in individual cells.” *Nat Genet*, vol. 36, no. 2, pp. 147–50, 2004.
[57] R. Lev Bar-Or, R. Maya, L. Segel, U. Alon, A. Levine, and M. Oren, “Generation of oscillations by the p53-mdm2 feedback loop: a theoretical and experimental study.” *Proc Natl Acad Sci U S A*, vol. 97, no. 21, pp. 11 250–5, 2000.
[58] L. Ma, J. Wagner, J. Rice, W. Hu, A. Levine, and G. Stolovitzky, “A plausible model for the digital response of p53 to DNA damage.” *Proc Natl Acad Sci U S A*, vol. 102, no. 40, pp. 14 266–71, 2005.

[59] N. Monk, "Oscillatory expression of hes1, p53, and nf-kappab driven by transcriptional time delays." *Curr Biol*, vol. 13, no. 16, pp. 1409–13, 2003.
[60] J. Wagner, L. Ma, J. Rice, W. Hu, A. Levine, and G. Stolovitzky, "p53-mdm2 loop controlled by a balance of its feedback strength and effective dampening using atm and delayed feedback," *Syst Biol (Stevenage)*, vol. 152, no. 3, pp. 109–18, 2005.
[61] "Cell designer." [Online]. Available: www.celldesigner.org
[62] "SBML." [Online]. Available: en.wikipedia.org/wiki/SBML
[63] "Open philosophies for associative autopoietic digital ecosystems project," *EU Framework 6 - Network of Excellence*, 2006. [Online]. Available: http://www.opaals.org/

Linguistic Dimension of Terms:

towards community's common vocabulary and concepts

[OPAALS Conference 2007]

Oxana Lapteva, Hagen Peukert
Computational Linguistics
University of Kassel
Kassel, Germany
oxana.lapteva@uni-kassel.de
hagen.peukert@uni-kassel.de

Abstract—**The constantly growing amount of textual information in network environments opens a new dimension of the extraction of meaningful data. In the context of an interdisciplinary community, the construction of "common language" is an important and challenging task closely related to community building and development. One of the issues in the common language construction is the extraction of shared terms. We claim that the extraction of relevant terms provides the fundamentals of lexicon development, conceptual and ontological representation of knowledge and Semantic Web. This paper investigates important issues of the terminological work through the lens of an interdisciplinary community. We present first results of the term extraction and collection by means of the OPAALS (Open Philosophies for Associative Autopoietic Digital Ecosystems) project organisation. This task involves the building of a representative and balanced text corpus, its analysis by applying the TermExtractor tool provided by [1] and introducing the linguistic view of the community's relevant terms.**

Keywords: term extraction, terminology building, natural language processing

I. Introduction

Successful term identification is key to getting access to the specific knowledge and information existing in the communities. Terms seem to be the excellent descriptors of the informational content of textual documents [2]. However, it is not to be sneezed at their numerous linguistic variations [3]. In the context of interdisciplinary communities, the terms and their relationships convey knowledge and its structure across different subject fields. Due to the complexity of term integration and assembling from different disciplines, we refer to the term identification as an important research topic in areas of natural language processing, ontology building, organisational communication and community building. Since terms and their conceptual representation reflect "the features with the most condensed and expressive levels of semantic value" [4], their extraction and analysis are advantageous regarding the processes of knowledge representation, sharing and creation. Domain specific knowledge contains domain relevant terms and concepts as well as semantic relations between them.

Our work was inspired by the European Network of Excellence Project OPAALS (Open Philosophies for Associative Autopoietic Digital Ecosystems) dealing with building a sustainable interdisciplinary research community and developing the theoretical foundation for Digital Ecosystems by integrating three different research domains: social science, computer science and natural science. By means of the terminological and conceptual representation of the OPAALS knowledge, we recognise that the challenging task of building "common language" is regulated by the composite character of these research domains.

There is a growing research interest related to the automatic term extraction methods in natural language processing [5, 6, 7, 8]. The automatic term identification is a complex task that requires precise and efficient Natural Language Processing (NLP) techniques. There are three main approaches of the terminology extraction from text corpora [9]:

- ➔ linguistic approach that considers the terms as specific morpho-syntactic patterns [10]. It is based on techniques that "detect and extract the strings whose structure match some given pattern"[9].
- ➔ statistical approach that identifies different statistical features of words. According to [9], a variety of the statistical approaches deals with the automatic extraction of multi-word terms through calculations involving some association measures.
- ➔ hybrid approach that combines linguistic and statistical techniques. By applying this approach, one can first work with the statistical processing of the given information and afterwards apply different types of linguistic filters. Contrariwise, the statistical measures can be done based on the results from the linguistic analysis. Reference [7] applied syntactic filters after statistical processing, in order to extract the statistically significant word combinations that match some given morpho-syntactic pattern [6]. References [11] and [12] applied statistical measures to a list of term candidates previously selected through linguistic techniques.

In our work, we use the TermExtractor tool [1] to automatically extract relevant multi-word items from text corpora. Furthermore, we provide the results of the linguistic analysis.

II. Research Background

A. Natural Language

The study of terms and their use in specific contexts has strong links to different areas of research. The complex structure of human language and word sense disambiguation issues require the interaction of different subject fields such as computational and corpus linguistics, domain-specific language (DSL) and language for special purposes (LSP), terminology and lexicography, information retrieval (IR), text mining, and others. The complexity of natural language stems from the intricacy of language itself [14]. In the context of the community development, it is important to understand how linguistic items are selected and organised to serve the specific cognitive aims of a particular knowledge domain.

Our choice of words and their usage may be conditioned by what we are talking about and to whom we are talking [14]. Signs and their meaning are formed by social groups primarily as part of the social division of labour in society. These issues influence the terminological work within a specific domain. A large number of groups may develop symbol systems (terms and concepts) and share knowledge, which they do not share with the rest of society. There may be a considerable degree of common knowledge and shared meaning [15]. Hence, the structured terminology can be seen as a multidimensional, interacting system that helps to build and organise communities and their relations and interactions with others.

One of the important goals of community building, especially in the context of the OPAALS project, is knowledge production, representation and sharing. The community plays a critical role in determining what its members acknowledge to be true, and how they develop knowledge. Knowledge is always situated in place, time, conditions, practices and understandings [16]. Regarding collaborative knowledge production and management in the interdisciplinary communities, there is a need of an environment that can be used and managed intuitively from members of different domains and backgrounds. In order to achieve this goal, a “common language” on the basis of a mutually shared vocabulary, its meanings and relations, is necessary. Having an amount of relevant terms existing in a certain community or organisation and representing their meanings in a given domain, will provide an important base for subsequently establishing conceptual and ontological systems.

B. Terminology

We refer to the discipline Terminology as “the study and the field of activity concerned with the collection, description, processing and presentation of terms...” [17]. The primary function of terminological processing is the collection of the relevant terms containing terminological information which is undertaken in order to improve communication. Additionally, it deals with recordings of the special lexicon of a language, an archive of lexical usage and meanings. The creation and sharing of terminological resources play a significant role in the terminology and are strongly related to the processes of communication and dissemination of specialist knowledge and information.

In our research we are concerned about the collection, description, processing and presentation of the specialized vocabulary (terms) of the interdisciplinary community. In the context of the OPAALS research community, our term extraction process deals with the language of the special field. The collected terms with their description and relationships represent the knowledge structure. Therefore, approaching the study of terms in the OPAALS interdisciplinary community helps to understand the structure of its knowledge system and provides the complete and coherent picture about the nature, behaviour and interaction of concepts and their associated terms.

C. Term: definition and representation

Term is a linguistic representation of a concept. In a certain field, the term identification in a specific language can be done through the particular function of an object [17]. Terms are represented by single words, expressions, symbols, acronyms or multi-word phrases selected directly from the text corpus by means of term-extraction methodologies [4]. According to [17], terms have a “variable pragmatic status, which is usually associated with their [...] acceptability, exclusiveness of existence, and spread of use”. A common process of the term formation is the creation of new concepts by means of new discoveries and restructuring of existing knowledge. Generally, terms are represented by the limited number of morphological and lexical structures. The majority of them are simple or multi-word nouns. However, depending on the knowledge domain, such terms can also be represented by a verb, adjective, noun phrase, verb phrase, or adjective phrase.

One of the critical points in term identification is the ability of humans or computer programs to differentiate between words and terms. It is important to recognise that the terminological work is not equal to the simple collection of words frequently occurring in the text corpus. Our aim is to establish specialized vocabularies of the OPAALS community and to tie them to their respective concepts. There are different types of terms that are used to label specialized concepts:

Simple terms that are represented by just one word.

Complex terms that can be represented by two or more words. They can form different types of a terminological phrase.

Term abbreviation that is represented as a shortened form of a complex term or terminological phrase, made up of letters or syllables of its components.

Some examples of aforementioned term types related to the research of the OPAALS project are presented in the table below:

TABLE I. Term types

Term types	**Terms**

Simple terms	Community, Knowledge Language
Complex terms	Digital ecosystem Evolutionary framework Community network Semantic web
Abbreviations	OPAALS (Open Philosophies for Associative Autopoietic Digital Ecosystems) OKS (Open Knowledge Space) DE (Digital ecosystem)

.

III. Case Study: OPAALS Community

In the context of extracting relevant terms related to the knowledge system and structure of the OPAALS community, we need to take into account some of its characteristics such as geographical distribution, interdisciplinary and multilingual properties.

A. Geographically distributed community

As geographically distributed community, the OPAALS project links people across countries and organisational units such as work packages, institutions and concrete task forces. Several factors (distance, organisational affiliation, cultural differences etc.) influence the process of developing common language. Due to the cultural differences that can easily lead to communication difficulties and to misinterpretation, the process of identifying and extracting common terms is inevitable. Language also introduces a very basic barrier of communication [18]. Therefore, it is important to communicate at the same conceptual level.

B. Interdisciplinary character of the OPAALS

The OPAALS community is a system of three interacting and collaborating domains: social science, computer science and natural science. One of the OPAALS's overarching aims is to build an interdisciplinary research community in the field of digital ecosystems and therefore to focus on knowledge production and management. Besides the dispersed organisation of the project participants and institutions, the members' affiliation with different research fields within and/or across scientific domains provides a challenging background for the analysis of language, knowledge representation, and terminological work towards the development of the "common language".

C. Multilingual character of the OPAALS

Finally, the multilingual character of the community members reflects a challenging background for language research. The official language of communication is English. However, the fact that most of the members use it as a foreign language, complicates the process of identifying a common conceptual surface.

IV. Methodology

A. Corpus-based Approach

In our study, we use a corpus-based approach for extracting statistical and linguistic information of relevant terms existing in the OPAALS community.

A corpus is "a collection of text or speech material that has been brought together according to a certain set of predetermined criteria" [19]. Reference [20] introduces several characteristics of the corpus-based analysis:

➔ empirical character (analysis of the actual patterns of use in natural texts)

➔ a corpus, the large collection of natural texts, builds the basis for analysis

➔ computer-based approach of analysis by means of automatic and interactive techniques

➔ the analysis of the corpus depends on quantitative and qualitative techniques

There are several advantages of the corpus-based approach in the terminological studies [21]:

a) Using electronic corpora enables the automatic identification and extraction of the key terms relevant to the subject field.

b) The use of electronic text corpora provides the possibility of determining "linguistic patterns, such as different kinds of syntactic, semantic and pragmatic details" [21] .

c) The context of a particular term or term phrase, for example through collocation, concordance and other linguistic techniques can be exploited.

d) The corpus-based approach allows the retrieval of the conceptual characteristics of terms.

e) Corpus analyses allow to share information of different subject fields and enable them to reuse data.

Reference [19] identifies two important criteria of a corpus: representativeness and balance. Representativeness refers to the construction of the corpus that is "as representative as possible of the domain under study, by including samples from a broad range of material" [19]. Such corpus provides a comprehensive picture of the language population. The corpus we use in our study contains the web sites from the OPAALS wiki pages. We have chosen this resource because it directly provides us with the material of investigation, that is, the language used in the OPAALS research community. Thus, it enables us to study the specific vocabulary as it is today, but also, by comparison, how it will change while the community grows closer together. To make this corpus as representative as possible, we applied a certain set of filtering criteria. Since we are interested in the

terminological and conceptual knowledge representation of the OPAALS community, the pages containing reports, discussions of research topics, descriptions and progress of research tasks, etc. have been included in the corpus. Contrariwise, web sites containing system-relevant information (e.g., help pages), meeting information (e.g., place, time, directions, accommodation), and so on have been filtered out and not included in the corpus.

A balanced corpus refers to a collection that "attempts to cover as many textual styles as possible by trying to include samples from various genres" [19]. Our text corpus contains different styles of information and knowledge representation, from discussions, agenda and notes to the structured reports and articles related to all three research domains existing within the OPAALS community: social science, computer science and natural science. Working with the electronic corpus is in our case an excellent way to acquire knowledge about the OPAALS subject field and its specialized language.

Having created the corpus and completed cleaning and filtering processes, we applied the TermExtractor tool developed by [1] to identify and extract the specific terms existing in the OPAALS community.

B. Term Extraction

Term extraction is a process of "scanning" the text corpus (text or group of texts) to identify terms, concepts and other relevant information (e.g. the context where a particular term occurs). It is an important step in our research towards the systematic examination of the common vocabulary, its development and changes occurring through the dynamic communication and community development processes. The term extraction process starts with the identification that involves the recognition and selection of specific words or word combinations. After candidate terms have been identified in a text, several calculations and filtering processes need to be applied in order to collect terms and concepts that are indeed part of the subject field under study.

As other terminology extraction systems, the TermExtractor tool identifies the relevant terms from a given corpus based on two steps [13]:

a) A linguistic processor that parses the text corpora and extracts typical terminological structures, e.g. compounds (community building), adjective-noun (digital ecosystem) and noun-preposition-noun sequences (description of work, grammar of metaphors, exchange of goods, character of money).

b) Filtering of terminological candidates

The procedure of the term extraction through the previously mentioned steps is based on five values: Domain Relevance, Domain Consensus, Term Cohesion (or Lexical Cohesion), Artificial Frequency and Term Weight [13].

Domain Relevance

This value is calculated based on two measures: the frequency of terms and the measurement of a term with respect to the target domain via comparative analysis across different domains. The mathematical representation of the Domain Relevance is given as follows [13]:

$$DR_{D_i}(t)=\frac{(\hat{P}(t/D_i))}{\left(\max_j \hat{P}(t/D_i)\right)} \quad (1)$$

The conditional probabilities $\hat{P}(t/D_i)$ are estimated based on the frequency of term t in the domain.

Domain Consensus

The second value measures the distributed character of the term in a domain. More precisely, the domain consensus is expressed as [13]:

$$DC_{D_i}(t)=-\sum_{d_k\in D_i} norm_{freq}(t,d_k)\log(norm_{freq}(t,d_k)) \quad (2)$$

where d_k is a document in D_i and $norm_{freq}$ is a normalized term frequency. This value of the Domain Consensus is then "normalized for each term in the [0,1] interval" [13]. In case of an "even probability distribution across the documents of the domain" [13], the consensus becomes high.

Term Cohesion (or Lexical Cohesion)

The computation of the Lexical Cohesion aims to measure the association of multi-word terms based on the co-occurrence frequencies. The mathematical representation of Term Cohesion is expressed as follows [13]:

$$LC_{D_i}(t)=\frac{(n*freq(t,D_i)*\log(freq(t,D_i)))}{(\sum_{w_j} freq(w_j,D_j))} \quad (3)$$

Artificial Frequency

TermExtractor is able to analyse in any type of document different text layouts: bold, italic, title, underlined, capitalized, coloured, small caps, etc. Depending on the layout, it assigns different values to the term. For example, if a term occurs once in bold, the TermExtractor assigns to it a higher value (e.g., an artificial frequency of 5) rather than the low raw frequency (e.g., 1) [13].

Term Weight

Term Weight is a linear combination of Domain Relevance, Domain Consensus, Lexical Cohesion and Artificial Frequency [13].

$$w(t,D_i)=\alpha*DR+\beta*DC+\gamma*LC \quad (4)$$

The TermExtractor tool has demonstrated the successful and qualitative work of terms identification and filtering. The results of the linguistic analysis of terms extracted by the tool are presented in the following sections.

V. Term Collection: First Results

A. Overview of extracted terms

From the text corpus containing files of the OPAALS wiki, we have got more than 250 terms. The top ten are represented in the table below:

TABLE II. Top 10 terms from the text corpus.

Term	TW	DR	DC	LC	AF
digital ecosystem	0.924	1.000	0.920	1.000	0.563
social network	0.770	1.000	0.856	0.111	0.161
social science	0.769	1.000	0.823	0.207	0.197
research interest	0.762	1.000	0.859	0.088	0.098
sustainable community	0.737	1.000	0.708	0.140	0.401
business model	0.735	1.000	0.717	0.223	0.260
knowledge model	0.731	1.000	0.789	0.051	0.105
community building	0.730	1.000	0.722	0.105	0.309
community currency	0.726	1.000	0.490	0.301	1.000
computer science	0.722	1.000	0.697	0.280	0.154

(TW – Term Weight, DR – Domain Relevance, DC – Domain Consensus, LC – Lexical Cohesion, AF – Artificial Frequency)

The entire list of terms mirrors the interdisciplinary character of the OPAALS domain knowledge. It contains multi-words related to the social science, natural science, and computer science domains.

TABLE III. Interdisciplinary Terms

Social Science	Computer Science	Natural Science
social network	automatic code generation	cell metabolism
epistemic culture	client-server architecture	gene expression
socio-economic constructivism	software engineering	

B. Analysis of the results: Linguistic Dimension

In this section we provide the linguistic analysis of the extracted terms, their specifics, formation and development. Many special terms we obtained are created by identifying a particular function of an object (TableIV).

TABLE IV. Terms and functions

Object	Function	Term
Code	Generate	Code generation
Software	Develop	Software development
OKS	Visualise	OKS visualisation
Gene	Express	Gene expression

Subsequent specification of some terms can produce more complex entities, for example “automatic code generation”.

According to [17], term determination can be expressed by compound and derived nouns and any word class. Both types of determination has been observed in the field of our study. In both cases (compounding and derivation) the new term is a certain specification of the original one. Examples of such terms, identified through the linguistic analysis of the extracted terminological list, are shown in Table V.

TABLE V. Derivation and Compounding

Derivation		Compounding	
System	Ecosystem	Language	Natural language
			Formal language
Structure	Infrastructure	Community	Community building
			Online community
Term	Long-term	Network	Social network
			Community network
Formal	Semi-formal	Knowledge	Knowledge management
			Knowledge base

Another technique, though less frequent, has been found while analysing and extracting terms from text corpora. It is conjunction. Conjunction is defined as “the process by which two concepts are combined as equals in a new concept and this fact is reflected in the term” [17]. In the collection of terms extracted from our corpus, there are some candidates presented in Fig. 1.

Conjunction

Socio-economic constructivism
Client-server architecture

Figure1. *Conjunction*

When analysing terms in the context of term designation, we refer to the [17] who distinguishes three major approaches:

1. The use of existing resources

The term designation through the use of existing resources can be transferred by a metaphor. Two techniques serve this purpose: simile and the exploration of the polysemic nature of designations. Simile uses expression such as "-style", "-like", or "-type". The exploration of the polysemic nature refers to the resemblance between terms. This is based on the similarities in form, function, and position [17]. The list of exemplary terms is shown in Fig. 2.

knowledge model
community building
knowledge space
virtual space
digital culture

Figure2. Term Designation through the use of existing resources

A special case of designation is the production of homonyms, different words that are pronounced the same and possibly spelled the same. The existence of three research domains in the text corpus leads us to an interesting observation:

There are terms relevant for the OPAALS project community. However, the meaning and context of their use vary depending on the domain (computer science, natural science, or social science).

Trust – trust in social science and trust in computer science

Network – network in social science and network in computer science

In order to determine different meanings of the same term, further analysis (e.g., collocation concordance, etc.) is necessary.

2. Modification of existing resources

The designation of new concepts through the modification of existing resources can be done in different ways. In the collection of terms extracted from our corpus we found four types of designation:

<u>Changing word categories (suffixation)</u>

The changing word categories can be processed in different ways. The analysis of the terminological collection discovered the tendency of changing word categories from verb-to-noun and from noun-to-adjective.

VERB => NOUNS

In the collection of OPAALS-terms, two types of Verb => Noun transfer occur. Fig. 3 depicts the *Verb + ing* suffixation.

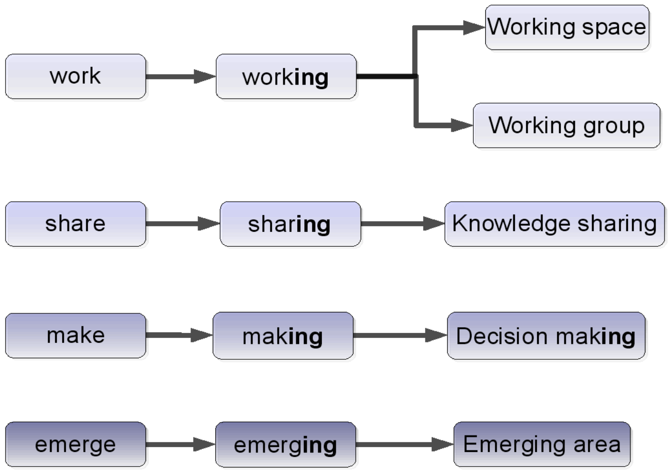


Figure3. Verb + ing suffixation

The second type of changing word categories through Verb => Noun rules is the *Verb + ion* suffixation shown in Fig. 4.

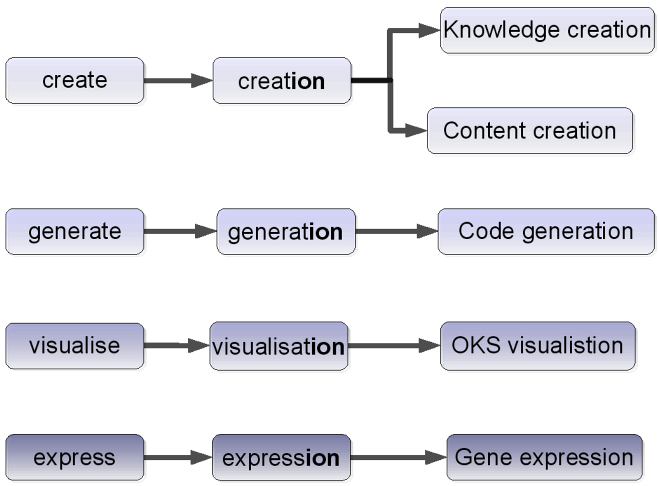


Figure4. Verb + ion suffixation

NOUN => ADJECTIVE

Finally, Fig. 5 shows examples of NOUN => ADJECTIVE suffixation.

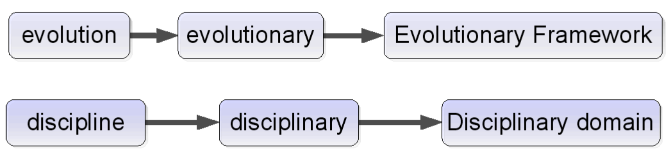


Figure5. Noun-Adjective suffixation

Compounding

The process of compounding is characterised by the combination of existing words into new ones. Examples of terms that have been formed by compounding are shown in Fig. 6.

Digital ecosystem
Autopoietic systems

Figure6. Compounding

Conversion

Conversion is defined as a "syntagmatically varied use of the same term" [17], for example, if a noun is used as a verb or an adjective is used as a noun.

community building
knowledge creation
software development
code generation

Figure7. Conversion

Compression

The compression can be represented in any form of shortening an expression by means of abbreviation, clipping, acronyms, etc. In our case study, we found a variety of terms that has been generated through the processes of compression. For example, OPAALS (Open Philosophies for Associative Autopoietic Digital Ecosystems) and OKS (Open Knowledge Space) became autonomous terms that are often used as part of more complex terms such as "OPAALS community", "OKS system", "OKS visualisation", and others.

VI. Conclusion and Future Work

This study shows that the automatic term extraction based on the electronic text corpus is a powerful technique for enriching the network of term candidates. We analysed the collection of terms through the lens of a linguistic dimension. Based on the accomplished work, several steps need to be done in the future to achieve our goal of a "common vocabulary" development:

a) Corpus modifications

The dynamic character of the OPAALS wiki-platform requires constant and continual adjustment, expansion, and filtering.

b) Context analysis

An important task in terminological work is the analysis of contexts the extracted term appears in. This process would help to identify the relationships between terms as well as their meanings.

The construction of the terminological collection and its linguistic analysis are the first steps in our research. Further steps will bring us closer to the overarching goal of building the common conceptual surface in the OPAALS interdisciplinary community.

References

1] F. Sclano, and P. Velardi, TermExtractor, http://lcl2.uniroma1.it/termextractor/.

2] P. Srinivasan, "Optimal document indexing vocabulary for MEDLINE," Information Processing and Management, vol. 32(5), 1996, pp. 503-514.

3] C. Jacquemin, J. L. Klavans, and E. Tzoukermann, "Expansion of multi-word terms for indexing and retrieval using morphology and syntax," in Proc. Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics, 1997.

4] R. Feldman, and J.Sanger, The text mining handbook: advanced approaches in analysing unstructured data. New York, NY: Cambridge University Press, 2007.

5] K. W. Church, and P. Hanks, "Word association norms, mutual information and lexicography, " in Proceedings of the 27th Annual Conference of the Association of Computational Linguistics, 1989, pp. 76-82.

6] G. Smadja, "Retrieving collocations from text Xtract," Computational Linguistics, vol. 19(1), 1994, pp. 143-177.

7] B. Daille, "Study and implementation of combined techniques of automatic extraction of terminology," in J.L. Klavans and P. Resnik, Eds., The balancing act: combining symbolic and statistical approaches to language, pp. 49-66, Cambridge, MA: MIT Press, 1994.

8] S. Evert, and B Krenn, "Methods for the qualitative evaluation of lexical association measures," in Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics Toulouse, France, 2001.

9] I. Alegria, A. Gurrutxaga, P. Lizaso, X. Saralegi, S. Ugartetxea, and R. Urizar, "Linguistic and statistical approaches to Basque term extraction," in Proc. The Production Of Specialized Texts, 2004.

10] D. Bourigault, and C. Jacquemin, "Term extraction + term clustering: an integrated platform for computer-aided terminology," in Proc. European Chapter of the Association for Computational Linguistics, Bergen, 1999, pp. 15—22.

11] B. Daille, "Combined approach for terminology extraction: lexical statistics and linguistic filtering," in UCREL Technical Papers, vol. 5, University of Lancaster, 1995.

12] J. Justeson, "Technical terminology: some linguistic properties and an algorithm for identification in text," in IBM Research Report, RC 18906 (82591), 1993.

13] F. Sclano, and P. Velardi, "TermExtractor: a web application to learn the shared terminology of emergent web communities" in Proc. of the 3rd International Conference on Interoperability for Enterprise Software and Applications (I-ESA 2007), 2007, in press.

14] A. Farghaly, and B. Hedin, "Domain analysis and representation," in A. Farghaly, Ed., Handbook for Language Engineers Stanford, CA: SCLI Publications, 2003.

15] B. Hjorland, "Domain analysis in information science: eleven approaches - traditional as well as innovative," Journal of Documentation, vol. 58(4), 2002, pp. 422–462.

16] N. A. Van House, "Epistemic machineries of environmental online communication," in A. Sharl, Ed., Environmental online communication, London: Springer, 2004, pp. 199-208.

17] J. C. Sager, A practical course in terminology processing. Amsterdam/Philadelphia: John Benjamins Publishing Company, 1990.

18] E. Wenger, R. McDermott, and W.M. Snyder, Cultivating communities of practice: a guide to managing knowledge. Boston: Harvard Business School Press, 2002.

19] K. Megerdoomian, "Text mining, corpus building, and testing," in A. Farghaly, Ed., Handbook for language engineers, Stanford, CA: CSLI Publications, 2003, pp. 213—268.

20] D. Biber, S. Conrad, and R.Reppen, Corpus linguistics: investigating language structure and use. Cambridge: Cambridge University Press, 2004.

21] C. Popp, "Nomenclatures and computer-aided terminography," in Language for special purposes: perspectives for the new millennium, F. Mayer, Ed. Tübingen: Gunter Narr Verlag, 2001, pp. 195—201.

Enabling Trust with a Distributed Accountability Model for Digital Ecosystems

Paul Malone[1] and Brendan Jennings[2]

[1,2]Paul Malone, Brendan Jennings, Telecommunications Systems and Software Group , Waterford Institute of Technology, Waterford, Ireland, e-mail: (pmalone, bjennings@tssg.org

***Abstract*—Successful uptake of digital ecosystems requires trust in the underlying platform and in the users of that platform. Accountability mechanisms provide a basis for creating, monitoring and evolving trust. This work analyses the requirements for accountability in digital ecosystems and proposes an distributed accountability model for deployment of digital ecosystems in peer-to-peer networks.**

***Index Terms*— Accountability, Decentralised, Distributed, Trust, Digital Ecosystem.**

I. Introduction

The digital ecosystem paradigm provides a complex system analagous to that of the classical ecological environemtn in nature. Interactions between entities in such a system require that there is trust established not only between the entities, but also in the platform upon which these interactions occur. In addition, where these environments are being used for economic transferral of value, it is essential that interactions can be accurately monitored and independently validated against Service Level Agreements (SLAs). Robust accountability mechanisms must be made available to accurately provide the means through which these interactions can be logged. Such accountability facilities enable trust levels to be established, monitored and evolve. In the case where interactions fail, accountability provides the means through which the cause of this failure can be clearly and unambiguously established.

In the Digital Business Ecosystem Project some work has been done in the study of trust for Digital Business Ecosystems. In particular, the project provides a taxonomy that is comprised of three major dimensions: trust types (X, Y and Z), building blocks of regulatory trust (privacy and consumer protection, e-signatures and security, jurisdiction and consumer protection) and operational perspectives (DBE relationships, actors and software lifecycles).

The trust types were identified on the base of regulatory trust in the DBE environment, where the regulatory aspect was considered *"central to building trust relationships between partners"* (Deliverable 32.2). Several levels of trust can be distinguished in a DE:

Trust type X refers to the *trust in the system*. The trust companies and users have in the technical architecture and services of the system determines them either to join or not to the join the system.

Trust type Y refers to the *expectations* established members have about *joining users*.

Trust type Z refers to the *trust relationships between participants* of the system. A higher mutual trust results in a higher number of transactions and collaborations between participants.

The purpose of the work described here is to examine accountability requirements in peer-to-peer digital ecosystems with a view to establishing a model of accountability which provides the required functionality for peer-to-peer digital ecosystem deployment.

Wang and Singh [WANG-07] view trust in federated systems as a reputation mechanism incorporating the knowledge of others in the system when deciding whether to trust in another party. They augment this view of trust in peer-to-peer systems with "belief" or expectation that the other peers output will provide the expected result. This belief approach is underpinned by available "evidence". Strong accountability provides trustworthy evidence and in turn provides accurate assessment of trustworthiness of peers and the system also.

Peer-to-peer networks are a desirable choice for deployment of digital ecosystems, due to the inherent decentralised, resource sharing and scalability nature of these networks. A suitable accountability framework must provide the following in order to successfuly provide traceability and establish trust.

A. Decentralised

One of the fundamental requirements for digital ecosystems is that there is no single point of failure. If any one node (or any sets of nodes) becomes unavailable, the system must be capable of recovering completely without any external intervention. A suitable accountability solution requires therefore that there is no dependence on a single centralised accountability manager or co-ordinator and that all functionality is distributed across the system.

B. Service Composition

A crucial aspect of digital ecosystems is collaboration between participants. In this sense it is important that services and resources can be combined to create new services. This service composition needs to be dynamic and cannot be known in advance. An accountability solution requires that such a dynamic composition of services can be seamlessly and efficiently accounted for.

C. Scalability

The number of peers in a digital ecosystem is arbitrary, ranging from a handful to several thousand. A suitable solution needs to work for large sets of peers as well as large service compositions.

D. Security

The implications for security when providing accountability in dynamic composing in digital ecosystems are wide ranging. Integrity of accounted data, availability of accounted data, confidentiality of accounted data, privacy of transactions al need to be considered when designing a suitable accountability model.

II. Accountability

Koppell [KOPP-05] cites The Public Administration Dictionary's definition of accountability as *"a condition in which individuals who exercise power are constrained by external means and by internal norms"* and makes the point that the external means (e.g. citizens, legislature, courts) may have conflicting interests. Furthermore he makes the point that "*Some analyses treat "accountability" as a synonym for law-abiding while others envision responsiveness to popular demands"*. Given the lack of conceptual clarity and the multiple meanings of the term accountability Koppel introduces the problem of Multiple Accountability Disorder (MAD) whereby an organisation suffering from MAD moves between behaviours influenced by conflicting concepts of accountability.

Koppell argues that providing a single definition for the term "accountability" renders the concept meaningless and provides five dimensions to accountability to enable clearer discussions:

1. *Transparency*: Did the organisation reveal the facts of its performance?
2. *Liability*: Did the organisation face consequences for its performance?
3. *Controllability*: Did the organisation do what the principal desired?
4. *Responsibility*: Did the organisation follow the rules?
5. *Responsiveness*: Did the organisation fulfil the substantive expectation?

The remainder of this section examines accountability in computer systems focussing on on SOA and peer-to-peer models of accountability .

A. Accountability in Computing Systems

Since the proliferation of computer systems in the 1970s in business and governmental support systems, an acknowledgement for the need of accountability in these systems has steadily grown. This requirement has been further accelerated by the availability and general ubiquity of the Personal Computer through the 1980s and 1990s through to today's widespread availability of broadband Internet access for small businesses and individual domestic users. Nissenbaum [NISS-94] recognised that a community which insists on accountability where participants are answerable for their actions provides a signal for high quality work and encourages responsible actions among the participants. With the absence of such accountability, no participant is answerable for risks or harm. She believed that accountability was being undermined in the emerging computerised society and that this undermining was due to three underlying factors rather than emerging computer system proliferation:

1. *A narrow understanding of the concept of accountability*
2. *A set of assumptions concerning the capabilities and shortcomings of computer systems*
3. *An acceptance that producers of computer systems were not fully answerable for the consequences of the usage of those systems*

Nissenbaum argued that life-critical systems required accountability not only to provide trust in those systems, but also to provide a means through which the end user had recourse in the event of malfunction.

Providing accountability, according to Nissenbaum, was not ultimately concerned with pinning blame on someone, and provides three recommendations to encourage accountability in the emerging computerised society:

1. *Keep accountability distinct from liability to compensate.*
2. *Clarify and promote a substantive standard-of-care.*
3. *Impose strict liability for defective consumer-oriented software and software whose impact on society and individuals is great.*

B. Accountability Protocols

The purpose of traditional network security protocols is to provide secure communications in insecure networks. These protocols prevent unauthorised agents or persons from obtaining private data or impersonating another entity. These protocols provide protection where there is trust established between the network resource and the authorised user. They do not provide any protection where there is no such trust. In a scenario where two participants in an electronic exchange of data or services are unknown to each other, more stringent protocols are required to ensure the identity of the participants and to account for their actions during the interaction. One method of strengthening trust and achieving this goal is a preliminary registration process such as SET[1] or Visa 3-D[2]. Accountability protocols go further than this and provide the ability to ensure two further elements are present in the interaction [BELLA-06]:

1. *Non-repudiation of actions*: neither party can deny their actions in the interaction.
2. *Fairness*: Both participants receive the expected outcome of the transaction, or neither do.

An accountability protocol provides lasting evidence about actions performed by the peers. In general such protocols involve three parties, the two participant peers

1 Secure Electronic Transaction (SET) is a standard protocol for securing credit card transactions over insecure networks, specifically, the Internet.

2 Visa 3-D Secure is an authentication technology that uses Secure Sockets Layer (SSL/TLS) encryption and a Merchant Server Plug-in

and a trusted third party (TTP). [KREM-02] provides a set of definitions relating to non-repudiation including definitions of different types of TTPs:

Inline TTP: A TTP involved in each message transmitted during the protocol.

Online TTP: A TTP involved in each session of the protocol but not each message.

Offline TTP: A TTP only involved in a protocol in the case of incorrect or dishonest behaviour, or in the case of network failure.

Neutral TTP: A TTP whose assistance is not pre-conditioned by knowledge of the the information to be exchanged.

Transparent TTP: An offline TTP which produces evidence indistinguishable from the message exchange in a faultless case.

In a message interchange between A and B, non-repudiation of origin (NRO) provides a validation to B that the message originated from A, while non-repudiation of receipt (NRR) provides validation to A that B received the message. Such non-repudiation is normally brought about through the use of signatures, encryption, notarisation and data integrity mechanisms. [KREM-02] provides a detailed description of several fair non-repudiation protocols with no TTP, with inline, online and offline TTPs. Two such protocols using online TTPs are *the fair non-repudiation protocol* [ZHOU-96] and *the certified email protocol* [ABADI-02]. Two further protocols without the need for a TTP are also of interst, namely the *Markowitch and Roggeman protocol* [MARK-99] and the *Mitsianis protocol* [MITS-01]. Further details of all these protocols are available in the corresponding papers and of other protocols in [KREM-02].

C. Accountability in Network Services

As previously noted, traditional security mechanisms are not sufficient in providing adequate accountability facilities in electronic communications. This is becoming more evident as services grow more complex and inter-dependent. In service provision platforms, services often consist of a set of interacting components spanning several trust domains. In grid computing and peer-to-peer systems, these components may run on infrastructures controlled by untrusted third parties. In this scenario fully accountable services provide guarantees of correctness and also aloow for increasing trust among participants towards each other and in the system as a whole.

Yumerefendi and Chase [YUM-04] provide 3 fundamental properties of accountability in terms of service provision:

Undeniable: The actions of an accountable actor (a service or its clients) are provable and non-repudiable.

Certifiable: A client, peer or external auditor can verify that an accountable service is behaving correctly and prove any deviations from this correct behaviour to a third party.

Tamper-evident: Any attempt to corrupt a service state should carry a high probability of detection.

Yumerefendi and Chase argue that *"system builders should view accountability as a first-class design goal of services and federated distributed systems..."*. In examining accountability as a general design goal, they conclude that *"a key limitation of accountable design is that it is not fully general and that it must be 'designed in' to application structure and protocols"*. They provide a framework for accountable services which is an extension of an approach know as KASTS [MAN-02]. The purpose of this framework is to provide accountability for a general service class which accesses and updates internal state in response to client or peer requests. Execution occurs in a sequence of numbered rounds. In each round the service updates one internal state variable. At the end of each round the service publishes a signed, timestamped non-repudiable digest of its internal state and provides this to an external observer.

When a client requires an execution from a service, it digitally signs the request. After verifying that the request is valid the service executes the request and on completion returns a result with cryptographic evidence certifying its correctness relative to the published round requests. It should be noted that this approach requires access to the internal state of the services in question and as such is unsuitable for the needs of the service driven peer-to-peer approach being pursued.

D. Accountability and Service Oriented Architectures

The rise of Web Services and an increased interest in Service Oriented Architectures (SOA) in general has provided business-to-business (B2B) type transactions with a set of open inter-operable standards through which Internet business transactions can be performed. As an approach to building IT systems, SOA connects applications across a network via a common communications protocol, allowing organisations to reuse old software. Web Services have in fact become the de-facto standard for such B2B interactions. This proliferation of Web Services in providing inter-operable, scalable and composed B2B services has a drawback in terms of providing accountability; there is no standardised approach to providing accountability within the current set of protocols and standards. While there exist specifications on Security [WSS] and on Trust [WST], these are only applicable within trusted domains. This lack of clearly specified accountability models has prompted some work in the area of accountability in Web Services/SOA (e.g. [ROB-05], [ZHANG-06], [ZHANG-07]).

RosettaNet define a set of externally observable

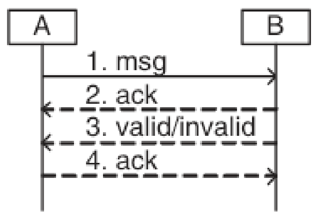


Fig. 1: Business Message Delivery Platform

elements of B2B exchange in terms through a set of Partner Interface Processes (PIPs). Typically each message is acknowledged with a confirmation of receipt. Fig. 1 shows the delivery of a business message and its associated acknowledgements. The initial message receipt is acknowledged by B, who in turn sends a message signifying whether the original message was valid or invalid. Upon receipt of this message, A again acknowledges receipt.

Fair Non-repudiable Web Services (WS-NRExchange)

Robinson et al, [ROB-05] describe a protocol (which they call WS-NRExchange) to enable non-repudiable and fair interactions with Web Services. Their protocol makes use of the non-repudiation protocol of Zhou and Gollman [ZHOU-96] and is based on applying such a protocol to the message exchange pattern shown in Fig. 2. This is achieved by the introduction of an inline TTP, the Delivery Agent (DA).

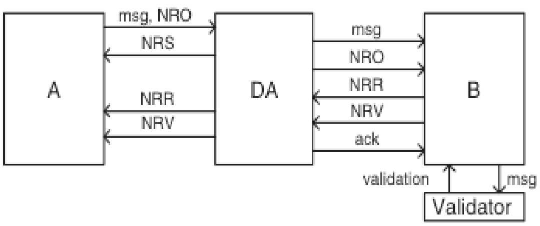


Fig. 2: Business Message Pattern with inline TTP for Non-repudiation

This approach provides four types of evidence for non-repudiation.

1. **NRO**: non-repudiation of origin; *msg* originated at A
2. **NRS**: non-repudiation of submission: *msg* was submitted by A
3. **NRR**: non-repudiation of receipt: *msg* was received by B
4. **NRV**: non-repudiation of validation: *msg* was validated (positively or negatively) by B

In terms of implementation Robinson et al suggest the use of interceptors to abstract the non-repudiable message exchange from the service endpoints. Most implementations of web services have such interceptors available (e.g. Axis Handlers). This approach is shown below in Fig. 3, which depicts the proposed WS-NRExchange architecture.

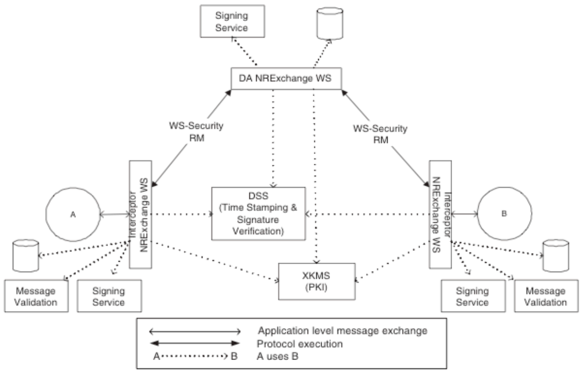


Fig. 3: WS-NRExchange Architecture

Accountability in SOAs

Zhang et al, [ZHANG-06], [ZHANG-07] present a model framework for accountability in SOAs which leverage Bayesian Networks in diagnosis of faults and a learning process linked with reputation. This model is more concerned with the self-healing aspect of autonomics through the monitoring of faults and reasoning, than providing a system of non-repudiation service provision.

They recognise that in the real world, business processes are often naturally organised hierarchically. This hierarchical management is efficient in that it aids the large-scale service failure as it facilitates a divide and conquer approach.

They present a 3-D approach (Detect, Diagnose and Defuse), which is implemented as follows:

1. **Detect**: Agents monitor behaviour of atomic services according to a predefined and report exceptions to an Accountability Authority (AA).
2. **Diagnose**: The AA makes use of Bayesian Network reasoning to discover the root of the problem when Service Level Agreement (SLA) violation occurs. This reasoning makes use of reported monitoring information gathered at the Detect phase.
3. **Defuse**: The AA updates the reputation of each atomic service based on the outcome of the Diagnosis phase. The AA then re-composes the service through selection of services least likely to violate SLAs, based on reputation.

In [ZHANG-07], they identify a set of new challenges for accountability techniques when deployed in SOAs:

- **Causality**: The outputs from upstream services have impacts on downstream services. Accountability approaches need to be capable of dealing with these causal relationships in these graph model based systems.
- **Scalability**: Fault diagnosis mechanisms need to scale to large-scale SOA systems.
- **Efficiency**: In terms of gathering runtime status information on the performance of services in large-scale SOA deployments, strategies must be designed to collect appropriate subsets of this status information while still allowing for accurate fault

diagnosis.

- **Uncertainty**: In SOAs, the behaviour of services is often uncertain at runtime. Accountability mechanisms need to built on a probability and statistical theory basis.

They designed a Hierarchical Diagnosis Algorithm (HDA) and a Accountability Authority (AA) to which faults are reported in order to address these issues in SOAs.

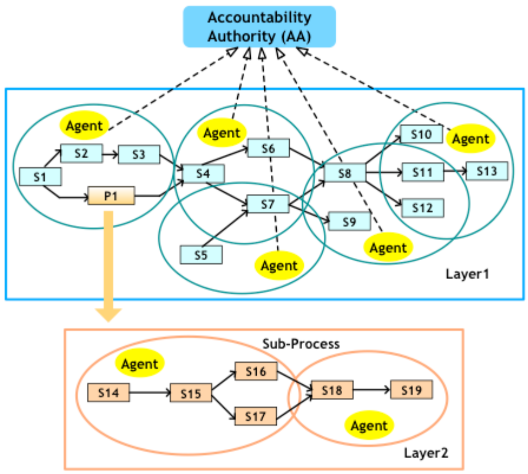


Fig. 4: SOA Accountability Architecture

E. Accountability and Peer-to-Peer Systems

Peer-to-peer networks operate through the leveraging of resources made available by its participants rather than relying on a set of servers providing a set of centralised finite resources. A pure peer-to-peer network operates on the basis that all peers are equal and each operates as both a client and a server in terms of providing and consuming services and/or content. These types of (pure peer-to-peer) networks have become the subject of much research and commercial interest in recent times due to their inherent lack of a single point of failure as resources are replicated across nodes, low cost of deployment in large-scale roll-out (due to the non-requirement of centralised management systems) and the fact that as more resources join the network, the greater the capacity of the network (due to the fact that it is the participants who provide resources). Another element of peer-to-peer technology that is of benefit on a social level is that, by its nature, it empowers its users who in effect collectively own the network and its resources. It is partly this reason that peer-to-peer networks have become the technologies of choice in the area of digital ecosystems.

In terms of accountability, peer-to-peer networks bring the same challenges that are in place for accountability in traditional client server models, i.e. Non-repudiation, fairness, etc. In fact, in peer-to-peer the requirements could be considered more important due to the fact that participants now operate in entirely untrusted environments without any centralised authentication mechanisms providing trust and strong user identity. In addition, the lack of centralised control brings its own challenges in terms of replication of (sometimes sensitive) data and resources across peers in untrusted environments.

The remainder of this section will present research published in the area of accounting and accountability in peer-to-peer networks and also projects which have been active in this field. While much of the work on accountability in peer-to-peer systems is motivated by a desire to reduce the 'freeloading' which can occur in resource sharing systems, there are also stronger real world economic drivers where these systems are used in sustaining digital communities with service provision.

MMAPPS

The IST MMAPPS (Market Management of Peer to Peer Services) project[3] looked at how services can be traded in peer-to-peer applications. The project viewed the provision and consumption of services in peer-to-peer environments as a marketplace and enhanced schemes, such as micro-payments, with more innovative, non payment-based accounting schemes such as ratings, where individual members of a particular community receive a rating score based upon their contribution. This rating then affects how other members provide services to that individual.

During the service preparation phase, the pricing module selects the tariff which will be used to account for usage of a particular service. As the tariff has to know the characteristics of the service to account for, it is developed in advance closely together with the service. The corresponding tariff parameters, however, are usually set dynamically and can be different on every peer. Optionally, fixed application-wide tariff parameters can be used which need to be specified by the Rules & Policies module [MMAPPS-10].

MMAPPS have also adopted a token-based approach to provide accounting and distributed pricing facilities to peer-to-peer service provision. With this approach, tokens are traded for service usage. A file-sharing scenario is discussed in [HAUS-03]. The project has defined an abstract architecture [STRU-03] satisfying this token based approach which can be seen in the following diagram.

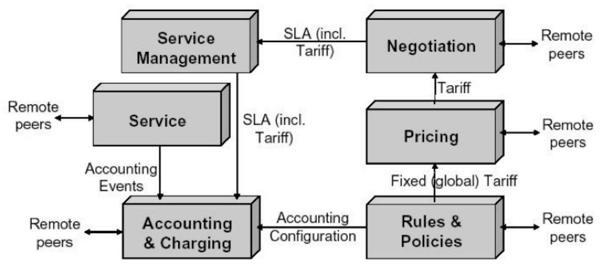


Fig. 5: Accounting & Charging Modules of MMAPPS

3 *MMAPPS (Market Management of Peer-to-Peer Services),* http://www.mmapps.org *(domain has lapsed)*

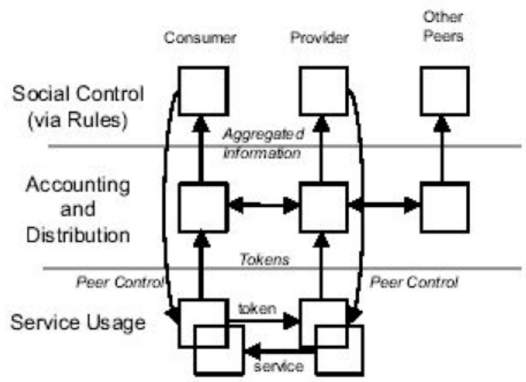


Fig. 6: MMAPPS Token-based Peer-to-Peer Economics Architecture

Karma

Karma [VISH-03] is a framework for sustaining resource sharing peer-to-peer networks using a secure economic model. The motivation for the work is the avoidance of freeloading is resource sharing networks. The system s economic in that it keeps track of the resource purchasing power of peers in the network.

Each peer is represented by a scalar value called a *karma* which represents the purchasing power of that peer. If a peer requests a resource, but does not have sufficient karma to purchase the resource, the transaction may not proceed. Participants are thereby incentivised to contribute as many resources (or more) as they consume. The framework provides properties of non-repudiation, certification and atomicity, protecting against both malicious providers and consumers. In addition, periodic correction to the outstanding karma in the system is performed in order to reduce the effects of inflation or deflation. Inflation can occur when peers use up their balance to consume resources and then leave the system. Deflation can occur when peers accrue large balances and then leave the system.

It is assumed that the minimum number of nodes in the network is *k*. Karma maintains its internal state via a peer-to-peer distributed hash table (DHT). The bank-set $Bank_A$ is a set of peers that independently maintain the karma balance of peer *A*. Each participant is assigned a unique identifier in a circular identifier space e.g. *NodeID(A)*. The bank set $Bank_A$ is the *k* closest nodes in the identifier space to *HASH(NodeID(A))*. This mapping is chosen as it allows the implementation to be layered on top of an existing DHT such as Pastry [ROWS-01]. Using this method the routing to each of the bank-set nodes can be performed efficiently. Each node in the bank-set $Bank_A$ independently stores the karma balance of *A*, signed by *A*'s private key (making the value tamper resistant). Each bank-set node also contains a transaction log of recent payments. Karma is not concerned with how the resources are shared or how a price is agreed upon for resource.

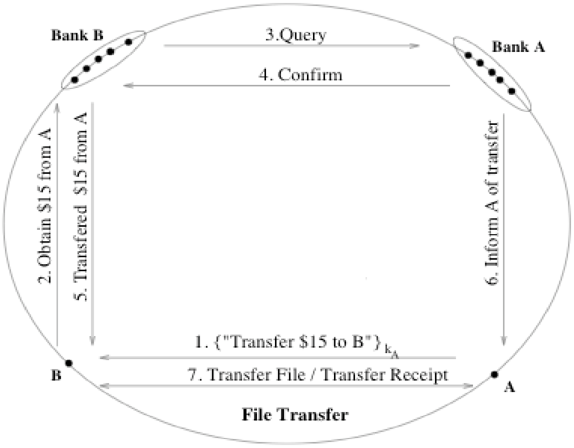


Fig. 7: KARMA resource/value exchange protocol

When a new node joins the system, a randomly selected public/private key pair is provided and using a value *x*, such that $md5(K_{public})$ *equals* $md5(x)$, in the lower *n* digits, where *n* is a system wide parameter, the new node's *nodeID* is set to $md5(K_{public},x)$. The node certifies that this calculation has been performed by signing challenges from its allocated bank-set with its private key.

Initially *A* sends *B* a signed authorisation for $Bank_A$ to transfer the agreed karma amount to $Bank_B$. *B* forwards this message to $Bank_B$, which in turn contacts $Bank_A$. If *A* has sufficient karma to satisfied the transfer amount, the amount is deducted from $Bank_A$'s balance and credited to $Bank_B$. $Bank_B$ then notifies *B* that the transfer took place and *B* releases the resource to *A*. The initial transfer from *A* includes the balance *A* will have at the end of the successful transaction (signed by *A*'s private key). Similarly when *B* passes this message to its bank-set, it includes a signed version of its own expected balance at the end of the transaction. This mechanism ensures that each bank node contains the latest balance corresponding to their client node.

PeerMint

In 2005 Hausheer and Stiller published a decentralised scheme for accountability in peer-to-peer networks which they called *PeerMint* [HAUS-05]. The work was performed as part of the MMAPPS[4] project and was concerned with accountability mechanisms for commercial peer-to-peer applications. The provided scheme can be used as a means of ensuring fair sharing of resources or as a means of applying charging and payment mechanisms to peer-to-peer applications. The work introduces the concept of session peers in addition to account peers in order to reduce significantly the possibility of collusion between peers. The session peers are responsible for maintaining balances and updates for the current session and for validating actions against a predetermined Service Level Agreement (SLA) between the peers.

While the accounting data is secure in that it ensures availability and integrity of data, it does not provide mechanisms for confidentiality or privacy. The interface to the accounting mechanisms is generic and can be easily applied to various environments.

4 *MMAPPS (Market Management of Peer-to-Peer Services),* http://www.mmapps.org*(domain has lapsed)*

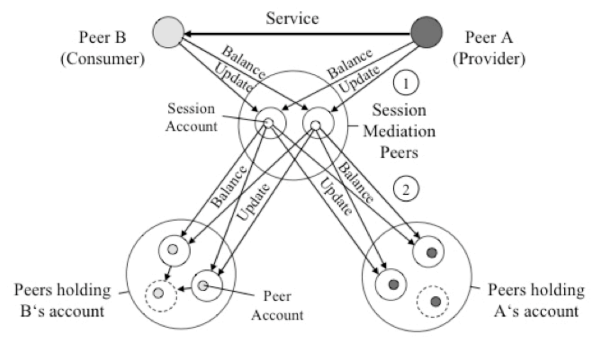


Fig. 8: Peermint Distributed Accounting

PeerMint takes a redundant approach to both session peers and account peers through the replication of data across a set of peers. The selection of account peers is made in a similar way to to the hashing method of Karma [VISH-03] in order to optimise routing. The selection of session peers also uses this approach but the hash is performed on a combination of the peer IDs and a timestamp denoting the point of session initiation. In both cases (account and session peers), when a peer goes offline a new peer is selected to take its place. The new peer obtains the balance from the remaining peers associated with the account or session. The model is shown below in Figure 14. New peers are shown as a dashed circle.

The PeerMint scheme was implemented using a local instance of FreePastry[5], an open source implementation of Pastry [ROWS-01]. The experimental results showed that the scheme's message overhead increased slowly in small networks, but that this overhead levels off as the network grows.

III. Accountability Model for Digital Ecosystems

A. Considerations

The relevant approaches outlined in Section 2 are addressed here with a view to suitability. The ones related to peer-to-peer networks and SOA are discussed here.

WS-NRExchange

The fair non-repudiable web services framework described by Robinson et al, [ROB-05] provides a useful solution and architecture to providing accountability in webs services. It has one major drawback for our needs in that it uses an inline TTP as an integral part of its architecture. Given the purely distributed nature of our desired infrastructure, this approach is not compatible with our needs.

SOA Accountability Architecture

This approach is a very useful contribution in that it addresses directly the challenges of Causality, Scalability, Efficiency, Uncertainty and Dynamicism as identified by [ZHANG-07]. One element of this framework that is of concern is the centralised nature of the Accounting Authority. It can be that the functions of this element can be deployed as a distributed service and thus over come this drawback.

MMAPPS

The *MMAPPS* project examined how services can be traded in peer-to-peer applications. While the project itself proved a useful architecture for accounting and charging for these services, it did not directly address accountability in terms of non-repudiation etc. A separate element of the project provided such a mechanism (*PeerMint*) that in fact does provide such non-repudiation elements, although service composition is not addressed.

KARMA

Karma provides a framework for non-repudiable resource sharing among peers and uses a secure economic model to achieve this end. The approach satisfies much of what is required, but again does not address service composition.

PeerMint

PeerMint is a decentralised scheme for non-repudiation accountability in peer-to-peer networks. The *PeerMint* approach is very similar to that of *Karma*, but goes further through the introduction of session peers in order to reduce the possibility of collusion among peers in creating reputation and value. *PeerMint* does not consider service composition in its approach.

Requirement	PeerMint	SOA Accountability Architecture
Decentralised	**Yes** The model is fully distributed as it takes a redundant approach to the distribution of accountability data.	**No** Not a distributed approach as the Accountability Authority is centralised
Service Composition	**No** The model does not consider Service Composition as it is primarily concerned with binary transactions between peers.	**Yes** The primary concern of this architecture is to account for composed services in an SOA environment. In addition this approach address the challenges for accountability in SOAs, namely Causality, Scalability, Efficiency, Uncertainty and Dynamicism
Scalability	**Yes** The model has been tested to show that the message overhead levels off for larger scale deployments	**Yes** The model uses Bayesian Networks in diagnosis of faults and demonstrates in this regard.
Security	**Partly** Data integrity and availability are satisfied by this approach. Confidentiality and privacy aspects are not considered.	**No** No security aspects were considered
Contracts	**Partly** SLAs are exchanged before the transaction takes place.	**No** Contracts are not considered as a part of this design

B. Model Description

We can see that most of our requirements are largely satisfied by one or both of these approaches. As *PeerMint* provides a fully decentralised approach that is suitable for our needs and the *SOA Accountability Architecture* provides a framework for accounting for composed services, it is likely that a combination of these approaches will provide most of what is required. The model under development is a combination of the*PeerMint* model published by Hausheer and and Stiller [HAUS-05] combined with the *SOA Accountability*

5 FreePastry, open-source implementation of Pastry, http://freepastry.rice.edu/FreePastry/

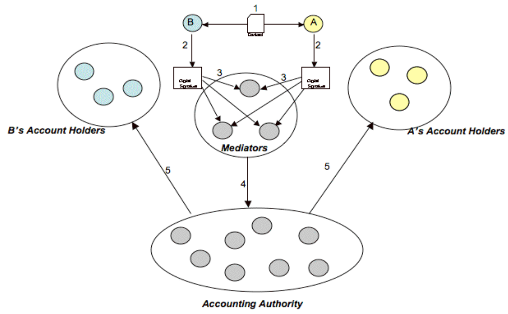


Fig. 9: Accountability Model

Architecture desigend by Zhang et al [ZHANG-07]. The idea here is to take the concept of the *Accounting Authourity* introduced by Zhang and to deploy this as a distributed service and combine this with the distributed *PeerMint* model. The model is shown in . Although the diagram shows a simple transaction between 2 services, the introduction of the *Accounting Authority* provides the required functionality necessary for composed services. The role of the accounting authority in a composed service scenario is provided by the super-set of mediation peers.

The model shows two peers (A and B) interacting in a peer-to-peer services trading environment.

Prior to interaction between the peers, each peer is assigned a set of trusted peers as account holders. This assignment is based on historical availability of peers and this set of peers is updated over time as the system evolves. Upon joining the network, each peer is assigned a unique 128-bit peer ID calculated fron that peer's public key using a secure hashing method. This is the approach taken by both The public key is provided by the peer's identity (modelled through Task 4.1, Distributed Identity Model in OPAALS). The original selection of account holders is based on a combination of trustworthiness and leaf sets closest to the peer to be accounted for. The trustworthiness of these peers is determined based on the distributed trust model being developed within workpackage 4 in OPAALS. In a similar way (as in *PeerMint*) the session peers are selected using a hash on the combination of the interacting peers IDs in combination with a timestamp. Both the account holders and the session peers are normal peers in the network. An interaction between A and B is shown in the diagram and is explained as follows:

1. Initially the peers exchange a contract which includes the charging schemes they agree for service consumption. At runtime when the peers agree to interact, they are assigned a set of session mediation peers. These mediation peers are provided with a copy of the contract. These peers are responsible for holding accounting data from both A and B for the current interaction session.
2. Each message passed between A and B each session is metered and the resulting usage data is digitally signed by the peer itself. In combination with the distributed identity model being developed and the availability of peers' public keys, these signatures can be verified.
3. The digitally signed usage data is then passed to the mediation agents who compare A's usage with B's to check that both are agreeing on how they are interacting and that these interactions conform with the previously agreed contract.
4. As the session evolves, the accounted data is passed to the super-set of mediation peers operating as the Accounting Authority and checked for integrity with respect to the overall service composition.
5. At the end of the session, the usage data is consolidated into a usage record by the Accounting Authority. These usage records are delivered to the corresponding set of account holders for both A and B.

IV. Conclusion and Furure Work

The model provided gives most of the functionality required to allow for distributed accountability in peer-to-peer networks where dynamic service composition is the enabler for multi-peer collaboration. However, some requirements are not satisfied fully by this model, namely security issues. Privacy is a security aspect which also need to be addressed in a future evolution of this model. In the model provided here, any peer which holds *A*'s public key can, in theory, access details of *A*'s account from its account holders. In the next evolution of this model an access control overlay needs to be added to overcome this issue. Furthermore, decisions on hashing methods, routing tables, leaf-sets and selection of account and session holders need to further considered as the OPAALS platform design emerges.

In addition, further work needs to be undertaken with regard to receiving inputs from other parts of the OPAALS project team to better understand the requirements in terms of business models for community networks and also socio-economic aspects (particularly that of community currencies) .

VI. References

[WANG-07] Wang, Y., Singh, M.,*Formal trust Model for Multiagent Systems*, Proceedings of International Joint Conference on Artificial Intelligence (IJCAI'07), 2007

[KOPP-05] Koppell, J., *Public Administration Review*, Public Administration Review,2005

[NISS-94] Nissenbaum, H., *Computing and Accountability*, Communications of the ACM,1994

[BELLA-06] Bella, G ., Paulson, L. C.,*Accountability protocols: Formalized and verified*, ACM Transactions on Information and System Securit,2006

[KREM-02] Kremer, S., Markowitch, O., Zhou, J.,*An intensive survey of fair non-repudiation protocols*, Computer Communications,2002

[ZHOU-96] Zhou J. and Gollman D.,*A fair non-repudiation protocol*, Proceedings of IEEE Symposioum on Security and Privacy, 1996

[ABADI-02] Abadi, M., Glew N., Horne B., Pikas, B.,*Certified email with a light on-line trusted third party: design and implementation*, Proceedings of the 11th international conference on World Wide Web, 2002

[MARK-99] Markowitch, O., Roggeman, Y.,*Probabalistic non-Repudiation without Trusted Third Party*, Proceedings of Second COnference on Security in Communication Networks, 1999

[MITS-01] MItsianis, J.,*A New Approach to Enforcing non-Repudiation of Receipt*, 2001

[YUM-04] Yumerefendi, A.R., Chase, J. S.,*Trust but verify:*

accountability for network services, Proceedings of the 11th workshop on ACM SIGOPS European workshop: beyond the PC, 2004, 2004

[MAN-02] Maniaitis, P. and Baker, M., *Enabling the Archival Storage of SIgned Documents*, Proceedings of the 1st USENIX Conference on File and Storage Technologies (FAST 2002), 2002

[WSS] Web Services Security (WSS) TC, *Web Services Security: SOAP Message Security 1.1*, 2006

[WST] OASIS Web Services Secure Exchange TC, *WS-Trust 1.3*, 2007

[ROB-05] Robinson, P., Cook, N., Shrivastrava, S., *Implementing fair non-repudiable interactions with Web services*, Proceedings of Ninth IEEE International EDOC Enterprise Computing Conference, 2005

[ZHANG-06] Zhang, Y., Lin, K., Yu, T., *Accountability in Service Oriented Architecture: Computing with Reasoning and Reputation*, Proceedings of IEEE Conference on e-Business Engineering, 2006

[ZHANG-07] Zhang, Y., Lin, K.,, *Hierarchical Management of Service Accountability in Service Oriented Architectures*, Proceedings of IEEE International Conference on Service-Oriented Computing and Applications, 2007

[MMAPPS-10] Gerke J.(Editor), *Specification and Implementation of the Peer-to-Peer Middleware; Deliverable 10 of the MMAPPS Project*, 2004

[HAUS-03] David Hausheer, Nicolas C. Liebau, Andreas Mauthe, Ralf Steinmetz, Burkhard Stiller, *Token-based Accounting and Distributed Pricing to Introduce Market Mechanisms in a Peer-to-Peer File Sharing Scenario*, , 2003

[STRU-03] Strulo, B., Smith, A., Farr, J., *An Architecture for Peer-to-Peer Economies*, , 2003

[VISH-03] Vishnumurthy, V., Chandrakumar, S., Sirer, E. G., *KARMA: Asecure Economic Framework for Peer-to-Peer Resources*, Workshop on Economics of Peer-to-Peer Systems, Berkeley, CA, USA., 2003

[ROWS-01] Rowstron, A., Druschel, P., *Pastry, Scalable, distributed object location and routing for large-scale peer-to-peer systems*, Proceedings of IFIP/ACM Middleware, 2001

HAUS-05: Hausheer, D., Stiller B., NETWORKING 2005, 2005

Declarative Service Composition with SBVR

[OPAALS Conference 2007]

Alexandros Marinos
Department of Computing
University of Surrey
Guildford, Surrey, UK
a.marinos@surrey.ac.uk

Paul Krause
Department of Computing
University of Surrey
Guildford, Surrey, UK
p.krause@surrey.ac.uk

Abstract—**This paper begins with an introduction of declarative programming, SBVR, service composition and the RESTful architectural style. On this basis, it is shown how these elements can be combined into a coherent service composition architecture that significantly reduces the barriers to entry for both producers and consumers of services within a digital ecosystem.**

Keywords: REST, SBVR, Declarative Programming, Service Composition

I. INTRODUCTION *(HEADING 1)*

In many ways, service composition has been the ultimate promise of service oriented computing. The ability to pick out services from the open web and connect them arbitrarily could revolutionize the way business is conducted. However, the hidden complexities and barriers to entry for both consumers and producers have prevented this promise from materializing. The current approaches to service composition, bound by the complexity of the WS-* stack, have prevented business people from utilizing it for exposing services but also for composing services exposed by others. However, ideas not previously examined are coming to the forefront, either new or not previously considered. This paper attempts to explore how these new approaches that touch on different aspects of service composition and can help make it available to a much broader audience. This paper is structured as follows: Section 2 examines the various aspects of service composition and the alternative approaches that can be leveraged. Section 3 sets the foundations for an architecture that utilizes these alternatives to achieve the pre-existing goals. Section 4 goes into details of how such a system would be implemented while section 5 offers a case study that illustrates the concepts that were presented in sections 3 and 4. Section 6 gathers the future work that still remains and offers some concluding remarks.

II. THE BUILDING BLOCKS

A. Service Composition

Service composition is the process of combining atomic services in order to achieve a composite goal. Service composition is separated into manual composition and automated composition. In manual composition, the services are positioned in a workflow by the user whereas in automated composition the user defines the goal of the service composition and depends on a software tool to define the most suitable way to achieve this goal. According to [2], automated composition can also be subdivided into three categories.

The first is 'Fulfilling Preconditions' in which pre-existing services are combined in UNIX pipeline fashion to fulfill the requirements of a service that does not exist in atomic form. For example a .doc to .pdf converter and a .pdf printer can be combined to emulate a .doc printer.

The second is 'Generating Multiple Effects' in which a number of services should be executed to produce separate but interrelated outcomes. A typical example of this type of composition is the travel scenario where flight and hotel can be booked independently but must be coordinated for their result to be useful.

Finally, a type of composition called 'Dealing with Missing Knowledge' is defined, where for instance a list of metro stations may be combined with a list of hotel addresses to identify hotels near metro stations. In this type of composition additional information sources are queried and the results are combined with the results from a service provider to satisfy more complicated queries.

While these types of service composition can be independent, there are problem domains which require a combination of approach. In this regard, the three types can be considered building blocks for a complete service composition system.

B. Declarative and Imperative Programming

Two major approaches to computer programming are the declarative and the imperative approach. In order to understand their differences, it is important to separate the 'what' and the 'how' of a solution to a computing problem. The 'what' refers to the properties that a solution must possess whereas the 'how' refers to the steps followed to achieve the required solution. Declarative programming focuses on specifying the 'what' and depending on a software tool to decide which is the most appropriate way of reaching the goal. An example of a declarative language is the well-known SQL which specifies properties of data but not the way to retrieve it, which is left to the database implementation.

Imperative programming focuses on the 'how', bypassing the need to define the properties of the required solution since the user can guarantee the desired properties by directly controlling the algorithm. Java is considered an imperative languages, although in later versions, declarative elements have

appeared either in the language itself or in libraries designed for it.

In the domain of service composition it is clear that there is a strong correlation between automated composition and declarative programming. Similarly, manual composition can be thought of as an application of imperative programming. It is important to be aware of the connection between programming paradigm and service composition strategy when selecting the appropriate method of interfacing with the user.

C. Expressing Requirements

In efforts related to service composition such as [6], various languages such as OWL-S have been used to describe the user's desired goal, usually serialized in XML. These languages require training to read and write but provide exact semantics to machines so that they can be directly executed.

A new and promising approach to human-computer interfacing is OMG standard Semantics of Business Vocabulary and Business Rules (SBVR [3]). As defined by [2], "SBVR provides a way to capture specifications in natural language and represent them in formal logic so they can be machine-processed". This allows users to be able to verify the requested service composition by directly reading the structured natural language used by SBVR which can then be parsed and executed by a machine.

Since SBVR is a way to capture specifications, there is no technical barrier to following a specific programming style. Specifically, one could conceivably use it to specify the structure of a process therefore conforming to an imperative programming style. However, the SBVR specification [3], quoting the Business rules manifesto [4] states:

> *"**Separate From Processes, Not Contained In Them**. Rules apply across processes and procedures. There should be one cohesive body of rules, enforced consistently across all relevant areas of business activity.*
>
> ***Declarative, Not Procedural**. Rules should be expressed declaratively in natural-language sentences for the business audience. A rule is distinct from any enforcement defined for it. A rule and its enforcement are separate concerns."*

While SBVR can be used in an imperative style, the full benefits of the Business Rules Approach that spawned it can only be attained when it is used declaratively. Work on expressing information systems declaratively with the use of business rules has been presented in [10]. This work predates SBVR but provides essential guidelines to its utilization in an information system.

D. Distributed Computing Architectural Style

Service composition approaches in literature assume a foundation of Web Services built on the WS-* stack including specifications such as SOAP, WSDL, XML Schema and others. It is to be expected that this paradigm is used in literature given the activity around it in industry and standards organizations in the past few years. Essentially the WS-* stack represents RPC-style interaction tunneled through the HTTP protocol as a transport.

However observation of uptake in the wild suggests that a different paradigm is taking hold and recently industry seems to be noting this. The APIs of major web applications from Google[13], Amazon[16] and many others are built using a different architectural style termed REpresentational State Transfer (REST). Even WS-* heavyweights such as Microsoft are working on projects that utilize REST [12], [14]. Also recently a number of standards have been approved that are compliant with and based on the REST style. [15], [17].

Contrary to what may be assumed, REST is not a new development. It was first identified in 1999 by Roy Fielding in his PhD dissertation [5] as a term to describe the architectural style used by the web. Consequently, HTTP 1.1 was released to better align the web with the principles of REST. While many have since championed REST as a competing web service paradigm to the WS-* stack, it has only recently begun to be more seriously considered with the publication of works such as [7] and the apparent lack of expected adoption for WS-* technologies outside the corporate firewall.

The main concept of REST is the resource as a document that is identified by a URI, a uniform resource identifier. Resources are to be accessed by a universal interface of well defined methods that should be resource-agnostic and therefore have the same, standard effect on all resources. In the case of HTTP 1.1, the methods include GET, PUT, POST and DELETE [18]. Other protocols such as WebDAV define methods such as LOCK and UNLOCK, suitable for transactions on resources. This Spartan interface is in contrast with the WS-* approach of defining a multitude of unique methods, one for each use case to be executed.

In the context of service composition, the separation between verbs (methods) and nouns (resources) encourages a more declarative style of expression where the user is concerned with constraining the outcomes of the service composition rather than the providers of the services or orchestrating procedure calls. REST also yields a more loose-coupled service composition since service providers can be alternated based on resource provided rather than interface compatibility.

III. Design and Architecture

A. Requirements

Service composition within a Digital Business Ecosystem requires focus on the items to be composed. Additionally, it is necessary to approach the business users with a perspective familiar to their pre-existing mindset. This requires minimizing the learning curve or if possible removing it altogether. Since the main focus is on Small and Medium Enterprises, the barrier to implementation of a service on the ecosystem should also be considered.

B. Design Approach

Since we are operating within the context of a business ecosystem, our initial focus is on 'Generating Multiple Effects' type of service composition that can handle use cases similar to the travel scenario. We believe that 'Fulfilling Preconditions' and 'Dealing with Missing Knowledge' types of composition can be then added incrementally.

A declarative programming paradigm is selected in order to move the burden of implementation to the providers of the platform rather than the business users. In this way the business user can express requirements by stating intent rather than workflow. SBVR complements the declarative approach and lowers the barrier to entry even further compared to XML-based languages.

A RESTful architecture brings improved loose-coupling, simpler implementation and leverages the pre-existing infrastructure of the web. A REST web service can also be naturally utilized as part of a businesses web presence by acting as the server to an Ajax in-browser client therefore encouraging reuse and reducing implementation overhead. Additionally, resource-orientation is natural for service composition within a Digital Business Ecosystem since the consumer is not interested on the provider but in the resource (product) provided and it would be advantageous for the underlying architecture to also reflect this.

It is important to make a distinction in the types of resources that can participate in a RESTful service composition. On the one hand there are bounded resources. These resources exist before being requested for by the user and therefore lists of available resources can be made to any query. For example a place on a specific flight exists before being requested by the user. These resources are called bounded resources. On the other hand, there are resources that are created specifically based on a user's request. These resources are called unbounded resources. It is important to note that these resources cannot be queried through a generic query but have to be requested on a one by one basis. An example of such a resource is a taxi reservation. If one were to ask what taxi reservations could be made during a specific day, the list would be infinitely long due to the granularity of such a service, and therefore unusable.

IV. Implementation Principles

A. Service Description

While RESTful web services have had uptake in practice, a suitable description language has not yet been standardized. A notable effort is Web Application Description Language [9] developed by SUN which is purpose built for describing restful web services but not yet mature. An alternative is WSDL 2.0 [19] which has been extended to express RESTful services but it's WS-* rooted complexity is a significant drawback. Additionally, expressivity provided by these languages is limited to variable types. Additional implementation information is expressed as text to be read by a developer. Leveraging SBVR as the basis of a service description language should provide sufficient capabilities for integration without human intervention while reducing the complexity level presented by these languages. Additionally SBVR can provide integration advantages when used as a language to express both services and requests on these services.

B. Repository

A service repository in a RESTful architecture has not yet been considered in literature. This seems reasonable considering the poor uptake of UDDI and the usage patterns associated with web services on the public internet. It is however useful to consider it within the context of a Digital Business Ecosystem. What is needed from the repository is to have semantics for all the resource types used within the ecosystem and the ability for service providers to register as providers of a specific resource. Any provider should be able to create a new resource type however it is expected that market forces will lead to demand-driven standardization of resource types. In fact it should be more feasible for a business owner to register as a provider of the competition's resource than it is to implement the same service interface as would be required in the WS-* approach.

C. Requirements Expression

Since our focus is on the 'Generating Multiple Effects' type of service composition, we should use SBVR as a means of expressing the desired effects and correlations between them. Within a RESTful architecture this translates to resources and constraints on their attributes. Therefore a two step process is needed. Firs the user selects the desired resources from the repository from which an SBVR vocabulary is generated. Then, based on this vocabulary, rules are written to express what combinations of these resources are acceptable with both absolute and relative constraints. Absolute constraints refer to properties of the combination itself whereas relative constraints refer to properties of a given combination of resources in comparison to all other available combinations.

D. Combination Generation

According to the vocabulary and rules that express the request, the combination generation algorithm should query the relevant providers and determine the combinations that satisfy the absolute constraints. Methods of RESTful querying are already implemented in Microsoft Astoria and Google's GData [11] which can serve as a reference. Determining the appropriate combinations requires identifying a suitable algorithm. Backtracking may fit the requirements; however there is probably a more efficient way to search the solution space for matches. It is important to note here that bounded and unbounded resources should be treated at different phases due to their different nature. At first, range queries are made to all bounded resources once and viable combinations of the results generated. Afterwards, the unbounded resources are queried repeatedly, once for each generated combination of bounded resources. At the end of this process, a list of combinations exists.

E. Combination Evaluation

After the suitable combinations are determined, they can be evaluated either manually or automatically. Automated

evaluation depends on relative constraints entered during requirements expression whereas manual evaluation returns the combinations to the user for arbitrary ordering. These two approaches can be combined by using automated evaluation as a step before manual evaluation therefore easing the user's work. However, if a composition is to be exposed as a service itself, manual evaluation is not feasible.

F. Transaction Generation

Once the desired combinations are determined, they should be combined into a transaction tree to be executed by a transaction model. The existence of more than one suitable combination is necessary due to the fact that the resources have not been locked since querying. Locking resources in this fashion would place great strain on providers and lead to system-wide delays.

V. CASE STUDY

In order to illustrate the concepts presented above, this section includes a case study. The case study is based on the well-known travel scenario where a user is trying to fulfill a request for travel arrangements.

A. Service Description

Each resource in the ecosystem is described by an associated set of vocabulary and rules that serves as a service description for each of the providers of the resource. For example the Flight resource could be described with the statements in Figure 1. Although not seen in the example, a resource could also be constrained by rules.

1. Flight *has* Departure Date/Time
2. Flight *has* Arrival Date/Time
3. Flight *is from* Departure Airport
4. Flight *is to* Arrival Airport
5. Airport *is in* location

Figure 1. Example of a resource description in SBVR.

B. Service Selection

As the user selects the resources that are added to the composition, the associated vocabularies and rules are also added to the service composition so that the rules about the service composition itself can be written using them.

C. User Requirements

Except for the vocabularies and rules related to the resources to be composed, a user's requirements also contain rules that link the service composition to each resource and also constrain the resources and service composition according to the need of the users. In our case study, the user's requirements are depicted in Figure 2. In the figure, statements can be seen connecting the resources such as 'Departure flight', 'Return Flight' and 'Taxi booking' to the service composition and to each other. While statements 1-20 may seem tedious, they do offer a description of the service that can be verified by

a. Vocabulary

1. Travel Arrangement *is* a Service Composition
2. Travel Arrangement *contains* a Flight *called* Departure Flight
3. Travel Arrangement *contains* a Flight *called* Return Flight
4. Travel Arrangement *contains* a Hotel Booking
5. Travel Arrangement *contains* a Taxi Booking
6. Travel Arrangement *has* a date *called* Departure Date
7. Travel Arrangement *has* a date *called* Return Date
8. Travel Arrangement *has* a time period
9. Time period *extends between* the departure date *and* the return date.
10. Travel Arrangement *has* a location *called* home location
11. Travel Arrangement *has* a location *called* destination location
12. Travel Arrangement *has* an amount *called* total cost

b. Rules

13. It is obligatory that the departure date *of* the departure flight *is same as* the departure date *of* the travel arrangement.
14. It is obligatory that the return date *of* the return flight *is same as* the return date of the travel arrangement.
15. It is obligatory that the departure flight *is from* an airport that *is in* the home location.
16. It is obligatory that the departure flight *is to* an airport that *is in* the destination location.
17. It is obligatory that the return flight *is from* an airport that *is in* the destination location.
18. It is obligatory that the return flight *is to* an airport that *is in* the home location.
19. It is obligatory that the pickup location *of* the taxi booking *is at* the destination airport *of* the departure flight.
20. It is obligatory that the drop off location *of* the taxi booking *is at* the location *of* the hotel booking
21. It is obligatory that the home location *of* the travel arrangement *is* London
22. It is obligatory that the destination location *of* the travel arrangement *is* Athens
23. It is obligatory that the pick up time *of* the taxi booking *is* 30 minutes *after* the landing time *of* the departure flight
24. It is obligatory that the departure date *of* the travel arrangement *is between* December 1, 2007 *and* December 3, 2007.
25. It is obligatory that the time period *of* the travel arrangement *is between* 16 and 18 days.
26. It is obligatory that the total cost *of* the travel arrangement *is minimal*.
27. It is obligatory that the total cost *of* the travel arrangement *is less than* £850

Figure 2. Example of a user request in SBVR.

an untrained user, and can also be reused as a resource themselves. The most interesting statements are the ones from 21 onward that build on the previous ones to express complex constraints on the length and cost of the travel arrangement. It can be thought that a company may have Travel Arrangement service composition templates ready for its employees who then only cave to add the rules relevant to their specific instance of the travel arrangement and execute the service composition as is.

D. *Extracted Queries*

Once the requirements of the user have been specified, a software system should take over and attempt to fulfill the requirements of the user. The first step in executing the service composition is to query the bounded resources. This implies that a standard RESTful querying mechanism exists within the ecosystem so the user does not have to specify something in her request. In our case study, the query to one provider for the departure flight would be similar to the following:

GET
http://rest.easyflight.com/flights[From:London][To:Athens][date:2007/12/01…2007/12/03]

The GET method is the standard method invoked in HTTP to retrieve information from a service. The query URL is constructed using similar principles as the ones found in Microsoft Project Astoria and Google's GData.

E. *Results*

Each query that is executed returns a set of results in plain XML without the use of SOAP envelopes. An example of the content of the results produced by a service query can be found in figure 3.

F. *Combinations*

Once all the queries on the bounded resources have been executed, The results are then combined according to the user's requirements to create fitting combinations. Once the combinations are in place, unbounded resources such as the 'Taxi Booking' are queried with the specific data of each combination and the results are inserted to the combinations to complete them.

G. *Evaluation*

When the combinations are complete they have to be evaluated in order for the most suitable ones to be selected for execution. Evaluations can have two phases: automated and manual evaluation. Automated evaluation is based on relative and absolute criteria entered along with the requirements. In our case study, rules 25, 26 and 27 are criteria that can help select the most suitable combinations according to the length of the time period and the total cost of each service composition. Application of these criteria falls within the automated evaluation of the generated combinations. However, if the user is interacting with the composition process directly through some user interface, the opportunity for manual evaluation as an additional step exists. This implies that the automatically evaluated combinations can be presented to the user for reordering or even rejection of certain combinations. This is of course preferred because in many cases users will have additional implicit requirements not stated which will cause them to alter the results of the automated evaluation. Of course manual composition in not always an option since the service composition itself may be a resource in the ecosystem and therefore limited in its communication with the user, that could also be a machine, by what the service interface provides. However manual evaluation remains a valuable option where it can be applied. The result of combination evaluation is not a single combination as may be expected, because there is no guarantee that the resources that comprise a combination will be available when the transaction is executed. For this reason we aim for redundancy and keep more combinations, as can be seen in Figure 4.

H. *Generated Transaction Tree*

Once the combinations to be executed have been determined, they have to be converted into a form that can be executed by the transaction model. This can be achieved by merging the combinations into a single transaction tree. The transaction tree can be represented with the notation presented in [8]. While the easiest way to do this is to utilize a serial alternative coordinator at the top level and connect to it the different combinations as separate sub-trees, this foregoes a lot of potential for optimization. Ideally, the transaction tree should utilize similarities between the combinations to create more optimal transaction trees that contain fewer leaf nodes and therefore avoid unnecessary service invocations.

I. *Executing the Transaction*

While the execution of the transaction tree itself is outside the scope of this paper, it should be noted that a RESTful protocol developed for the needs of a digital ecosystem should provide patterns or methods that allow for the transaction to be executed. In a very simple transaction model based on 2 phase commit, this would imply the need for a pattern or method that could provide locking over a resource for a limited amount of time and booking of the resource. Similar concepts have been proposed, with the most prominent being the LOCK and UNLOCK methods of WebDAV, however going beyond the basic methods of HTTP causes a need to maintain a different protocol, therefore foregoing all the advantages of the scale that the web has reached. The alternative is creating patterns that give the ability to perform these operations using the standard methods of the HTTP protocol.

VI. CONCLUSION & FUTURE WORK

Work still remains to be done, including covering the different types of service composition, other than 'Generating multiple effects'. Additionally, work should be done to examine more fuzzy results in case the initial queries do not provide results. For instance, a later taxi arrival is better than no service composition at all. Another large issue is integrating and automating a reputation system so that the providers are selected based not only on criteria relevant to the resource in question, but also their own performance. To this end, an

EasyFlight.com						
Flight No.	Departure Date/Time (Local)	Departure Airport	Arrival Date/Time (Local)	Arrival Airport	Seat Class	Price
EZF 5243	2007/12/01 06:50	LGW	2007/12/01 08:50	ATH	B	150£
EZF 5248	2007/12/02 12:40	LGW	2007/12/01 14:40	ATH	B	170£
EZF 5242	2007/12/02 17:50	LGW	2007/12/01 19:50	ATH	A	220£
EZF 5243	2007/12/03 06:50	LGW	2007/12/01 08:50	ATH	C	130£

Figure 3. Example of results returned by a service.

Position	Departure Flight	Taxi Booking	Hotel Booking	Return Flight	Composition Properties
1	**Departure Date/Time**: 2007/12/03 06:50 **Departure Airport**: LGW **Arrival Date/Time**: 2007/12/03 08:50 **Arrival Airport**: ATH **Seat Class**: C **Price**: £130	**Pickup Location**: Athens Intl. Airport **Pickup Date/Time**: 2007/12/03 09:20 **Destination**: Athens Holiday Inn **Price**: £20	**Location**: Athens **Room Class**: B **Room Capacity**: 1 **Start Date**: 2007/12/03 **End Date**: 2007/12/19 **Price**: £480	**Departure Date/Time**: 2007/12/19 15:50 **Departure Airport**: LGW **Arrival Date/Time**: 2007/12/19 17:50 **Arrival Airport**: ATH **Seat Class**: C **Price**: £170	**Departure Date**: 2007/12/03 **Return Date**: 2007/12/19 **Time Period**: 17 **Home Location**: London **Destination Location**: Athens **Total cost**: £800
2	**Departure Date/Time**: 2007/12/03 06:50 **Departure Airport**: LGW **Arrival Date/Time**: 2007/12/03 08:50 **Arrival Airport**: ATH **Seat Class**: C **Price**: £130	**Pickup Location**: Athens Intl. Airport **Pickup Date/Time**: 2007/12/03 09:20 **Destination**: Athens Hilton **Price**: £20	**Location**: Athens **Room Class**: B **Room Capacity**: 1 **Start Date**: 2007/12/03 **End Date**: 2007/12/18 **Price**: £440	**Departure Date/Time**: 2007/12/18 15:50 **Departure Airport**: LGW **Arrival Date/Time**: 2007/12/18 17:50 **Arrival Airport**: ATH **Seat Class**: C **Price**: £190	**Departure Date**: 2007/12/03 **Return Date**: 2007/12/19 **Time Period**: 16 **Home Location**: London **Destination Location**: Athens **Total cost**: £780
3	**Departure Date/Time**: 2007/12/02 12:40 **Departure Airport**: LGW **Arrival Date/Time**: 2007/12/02 14:40 **Arrival Airport**: ATH **Seat Class**: C **Price**: £170	**Pickup Location**: Athens Intl. Airport **Pickup Date/Time**: 2007/12/03 09:20 **Destination**: Athens Holiday Inn **Price**: £25	**Location**: Athens **Room Class**: C **Room Capacity**: 1 **Start Date**: 2007/12/02 **End Date**: 2007/12/19 **Price**: £475	**Departure Date/Time**: 2007/12/19 15:50 **Departure Airport**: LGW **Arrival Date/Time**: 2007/12/19 17:50 **Arrival Airport**: ATH **Seat Class**: C **Price**: £170	**Departure Date**: 2007/12/03 **Return Date**: 2007/12/19 **Time Period**: 18 **Home Location**: London **Destination Location**: Athens **Total cost**: £840

Figure 4. Combinations generated and ranked.

appropriate feedback mechanism should be built into the system that allows consumers to rate providers after the resource has been consumed and therefore evaluated.

While the work presented in this paper is still under development, we feel that we have laid an appropriate foundation for utilizing the emerging technologies such as REST and SBVR within a digital ecosystem for the purpose of declarative service composition. With combining the advantages provided by these technologies, service composition can be made accessible both to service providers and service consumers.

VII. REFERENCES

[1] U. Kuster, M. Stern, and B. Konig-Ries, "A Classification of Issues and Approaches in Automatic Service Composition," Intl. Workshop WESC, vol. 5, 2005.

[2] Hendryx, Stan "Model-Driven Architecture and the Semantics of Business Vocabulary and Business Rules" , Hendryx & Associates, 2005

[3] Object Management Group, "Semantics of Business Vocabulary and Rules Interim Specification," 2006, Online at: www.omg.org/cgi-bin/doc?dtc/06-03-02. Accessed: 25/10/2007

[4] R. G. Ross, "The Business Rules Manifesto," *Business Rules Group. Version* 2 (2003).

[5] R.T. Fielding, "Architectural Styles and the Design of Network-based Software Architectures," University of California - Irvine, 2000.

[6] D. Wu, B. Parsia, E. Sirin, J. Hendler, D. Nau, "Automating DAML-S Web Services Composition Using SHOP2," Proceedings of the Second International Semantic Web Conference (ISWC2003), 2003.

[7] L. Richardson and S. Ruby, "RESTful Web Services," O'Reilly Media, Inc., 2007.

[8] J. Yang, M.P. Papazoglou, and W.J. van den Heuvel, "Tackling the challenges of service composition in e-marketplaces," Research Issues in Data Engineering: Engineering E-Commerce/E-Business Systems, 2002. RIDE-2EC 2002. Proceedings. Twelfth International Workshop on, pp. 125-133, 2002.

[9] M.J. Hadley, "Web Application Description Language (WADL),"

[10] C.J. Date, "What Not How: The Business Rules Approach to Application Development," Addison-Wesley Professional, 2000.

[11] D. Obasanjo , "Google Base Data API vs. Astoria: Two Approaches to SQL-like Queries in a RESTful Protocol," 2007, Online at: http://www.25hoursaday.com/weblog/2007/07/13/GoogleBaseDataAPI VsAstoriaTwoApproachesToSQLlikeQueriesInARESTfulProtocol.aspx. Accessed: 25/10/2007

[12] P. Castro, "Overview: Microsoft Codename Astoria," Microsoft Corporation, 2007, Online at: http://astoria.mslivelabs.com/Overview.doc, Accessed: 25/10/2007.

[13] Google, "Google Data APIs (Beta) Developer's Guide," Online at: http://code.google.com/apis/gdata/index.html, Accessed: 25/10/2007.

[14] Microsoft Corporation, "Web Structured, Schema'd & Searchable (Web3S)", 2007, Online at: http://dev.live.com/livedata/web3s.pdf, Accessed: 25/10/2007.

[15] J. Gregorio, B. de Hora, "The Atom Publishing Protocol," The Internet Engineering Task Force, 2007, Online at: http://tools.ietf.org/html/rfc5023, Accessed: 25/10/2007

[16] Amazon, "Amazon Simple Storage Service (Amazon S3)," 2007, Online at: http://www.amazon.com/gp/browse.html?node=16427261, Accessed: 25/10/2007

[17] Y. Goland, E. Whitehead, A. Faizi, D. Jensen., "HTTP Extensions for Distributed Authoring--WEBDAV," Microsoft, UC Irvine, Netscape, Novell. Internet Proposed Standard Request for Comments (RFC), vol. 2518, 1999.

[18] R.Fielding, J.Gettys, J.Mogul, H.Frystyk, T. Berners-Lee., "Hypertext Transfer Protocol--HTTP/1.1. RFC 2616," The Internet Engineering Task Force, 1999.

[19] R. Chinnici, J. Moreau, C. Ryman, S. Weerawarana, "Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language," W3C Working Draft, vol. 26, 2004

Transaction Vectors

A Non-Interleaving Semantics for Long-Running Transactions

[OPAALS Conference 2007]

S. Moschoyiannis, A. Razavi and P. Krause
Department of Computing
University of Surrey
Guildford, Surrey GU2 7XH, UK
{s.moschoyiannis, a.razavi, p.krause}@surrey.ac.uk

Abstract—**Although long-running transactions for distributed databases have been analysed in detail, there is less formal analyses of transactions in the B2B domain. In this paper we are specifically interested in supporting long-term transactions involving collaborating businesses in a Digital Ecosystem. We first identify the core requirements for such a transaction model. Then following a discussion of some of the weaknesses of existing models, in the context of Digital Business Ecosystems, we then introduce a vector language-based model that overcomes these weaknesses.**

Keywords – digital business ecosystem; long-term transactions; semantics of concurrency; e-commerce

I. INTRODUCTION

Business transactions in a B2B context typically involve interactions between multiple partners, either service providers, or service consumers, or both. This requires that all partners behave in a coordinated manner – partners must follow some protocol to execute a transaction effectively. In this paper we describe a formal model of long-running multi-service transactions and the distributed orchestration of the underlying service executions. The formalisation of the interaction-based composition is aimed at describing the behavioural patterns services should follow in order to guarantee successful commitment or compensation within the transaction model.

The execution of a long-running transaction corresponds to conducting a business activity whose parts are offered as web services. These services need to be coordinated properly to increase confidence in a successful outcome. Therefore, each party engaging in a long-running transaction must handle not only the application-specific aspects of the various service executions but also their interactions with one another, in terms of the ordering of service invocations. The challenge is that such coordination of the underlying service executions needs to be performed in a fully distributed manner. This is a prerequisite for realising the benefits of the Service-Oriented Computing (SOC) paradigm but is also in line with the demands for local autonomy of SMEs. It is further reinforced by the nature of the supporting P2P architecture, which is based on the premise of openness or inclusiveness of the run-time environment as well as the absence of vulnerability to single points of failure. To address this challenge we need a carefully refined design of the structure of each Local Coordinator in setting up and executing a long-running transaction.

Within the database community the conventional definition of a transaction [1] is based on ACID properties: **A**tomicity – either all tasks in a transaction are performed, or none of them are; **C**onsistency – data is in a consistent state when the transaction begins, and when it ends; **I**solation – all operations in a transaction are isolated from operations outside the transaction; **D**urability – upon successful completion, the result of the transaction will persist. However, in advanced distributed applications these properties can present an unacceptable limitation and reduce performance dramatically [2].

Business transactions typically involve interactions and coordination between multiple partners. A long-running transaction comprises a number of sub-transactions or activities that involve the execution of a number of underlying services. It is often the case that internal activities need to share results before the termination of the transaction (transaction commit). More generally, dependencies may exist between activities inside a transaction due to the required ordering on the service invocations or due to the sharing of data. Booking a flight ticket before booking a hotel room at the destination is an example for the former, while booking a taxi or buying a bus ticket, depending on the remaining budget, is an example of the latter.

Further, in a digital ecosystem for business a number of distributed long-running multi-service transactions are expected to take place. Many business scenarios require that a transaction releases some results to another transaction, before it commits. For example, consider a organising the travel for a meeting, which includes booking a flight, booking a hotel, and then either go for booking a taxi or buying a ticket for the bus, depending on the remaining budget. (Note that this already indicates the need for exchange of results within a transaction.) Now if there is a really good deal for all the requirements (possible in terms of distance between hotel and meeting place, etc.), but it would require that the attendee goes slightly overspent, the 'travel booking' transaction (before finalising the bookings/purchases) may contact another transaction, the 'budget handling' transaction say, to seek approval for a slightly increased budget.

In other words, dependencies may exist across transactions due to the need for releasing partial results outside a transaction. This may not happen as often as releasing data inside a transaction but is nevertheless a central requirement for distributed long-running transactions if they are to cover a wide range of business models. Conventional transaction models such as *Sagas* [3] or the more recent models *Web Services Transactions* (WS-Tx) [4],[5] and *Business Transaction Protocol* (BTP) [6] do not provide capability for partial results and inevitably make it the business process designer's responsibility. This often means that an ad-hoc complicated transaction needs to be designed or, even worse, results in adding new transactions that do not reflect the exact needs of the business model itself, but rather are incorporated to get round the problem. It can be seen that designing transactions with full ACID properties limits significantly the coverage of business requirements and is thus not suitable for e-business in general, and digital ecosystems in particular.

It is only recently that long-running transactions, as understood in a business environment rather than in traditional database systems, have received the attention of the formal methods community. After a brief description of the basic characteristics of long-running transactions and the surrounding issues in modelling their critical aspects, we outline formal approaches to modelling transactions. The discussion focuses on which transactional aspects are covered and which are missing. We then introduce a formal model for long-running multi-service transactions in digital ecosystems for business and show how it can be used for a behavioural analysis in terms of the underlying service compositions and the compensation mechanism that needs to be deployed if some failure later in the transaction makes it necessary.

II. The Nature of transactions in Digital Ecosystems

As we have already indicated, the very nature of business – as opposed to database – transactions, opens up a different angle from which to view transactions. For example, the specification of a transaction may involve a number of required services, from different providers, and allow it to be completed over a period of minutes, hours, or even days – hence, the term *long-lived* or *long-running* transaction. Indeed, a wide range of B2B transactions (business activities [7, 8]) have a long execution period. A business transaction between SMEs in a Digital Ecosystem can be either a simple usage of a web service or a mixture of different levels of composition of several services from various service providers.

It is important to stress that the long-term nature of execution frames the concept of a transaction in a digital ecosystem, since most usage scenarios involve long-running activities. In such cases, it is impractical, and in fact undesirable, to maintain full ACID properties throughout the lifetime of a long-running transaction. In particular, as we discussed in the Introduction, Atomicity and Isolation are questionable.

Within a digital ecosystem, a large number of distributed long-running transactions take place, each comprising an aggregation of activities which in turn involve the execution of the underlying service compositions. In such a highly dynamic environment there is an increased likelihood that some transaction (or one of its internal activities) will fail. This may be due to a variety of reasons (platform failure, service abort, temporary unavailability of a service, etc.) including the vulnerability of the network infrastructure itself (platform disconnection, traffic bottleneck, nodes joining or leaving the network, etc.).

The standard practice in the event of a failure is to trigger compensating actions that will effectively 'undo' the effects of the transaction – those effects visible before failure occurred. The objective is to bring the system to a state that is an acceptable approximation of the state it was before the transaction started. However, this is not a trivial task. Recovering the system in the event of a failure of a transaction (or an activity inside a transaction) needs to be done in a way that takes into account the dependencies both inside (to ensure all dependent activities are undone) but also outside the transaction (to ensure that any dependent activities of other transactions are also undone).

Further, and when considering SOA as the enabling technology for open e-business transactions, the recovery and compensation mechanism must respect the loosely-coupled nature of the connections, since interfering with the local state of the underlying service executions violates the primary requirement of SOA. In addition, access to the local state may not even be possible in a business environment with SMEs as service providers. This is an issue that has been largely ignored by current implementations of transaction models such as *Web Services Transactions* (WS-Tx) [4],[5] and *Business Transaction Protocol* (BTP) [6].

The abortion of a transaction, even if it is successfully recovered and compensated for, can be very costly in the business environment. Rolling back the whole system in the event of a failure may lead to chains of compensating activities, which are time-consuming and impact on network traffic as well as deteriorate the performance. For this reason it is important to build into the system capability or flexibility to deal with failure. In other words, it is key to design for failure by adding diversity into the system and allowing for alternative scenarios. The idea is to get some leverage in avoiding the abortion of a whole transaction (and all other dependent transactions) by means of allowing alternative paths of execution in cases where the path chosen originally encountered a failure.

It is also important to be able to preserve as much progress-to-date as possible. If an activity (sub-transaction) of a transaction fails, it is essential to undo (roll back) only those activities that have used results of this activity, i.e. are dependent on it. It is highly desirable to avoid rollback of other activities that have produced results (committed) and are not dependent on the failed activity. These are often referred to as omitted results and do no need to be undone as that would mean they will need to be re-done (re-started, re-calculated, re-computed) once the transaction is restarted (after abortion and recovery). Addressing omitted results can have significant benefits for SMEs in digital ecosystems in terms of saving valuable time and resources.

In earlier work, we have performed an extensive review of Transaction models, such as WS-Tx and BTP, that have been designed with web services in mind, and are currently widely used in practice [9]. Apart from certain issues regarding their coordination mechanism, which is geared towards centralised control, these models do not support partial results, do not provide capability for forward recovery, and there is no provision for covering omitted results.

In the next section we will briefly review some very recent approaches to the formalisation of long running transactions. Following that, we will introduce a formal analysis of the transaction model that is being developed in the OPAALS project to overcome some of the weaknesses of the existing approaches.

III. Formal Approaches to Modelling Transactions

Not surprisingly, given the preceding discussion, a common denominator in many formal approaches to transaction modeling is the relaxation of ACID properties. This is often done by organizing a long-running transaction into a series of activities. The idea is that each activity is a discrete transactional unit of work that can either commit or abort (in which case the previously committed activities need to be undone or rolled back).

Perhaps not surprisingly, a number of researchers have taken a formal approach to modelling the primitives of long-running transactions with a version of a process algebra that draws upon the well-known process algebras *Calculus for Communicating Systems* (CCS) [10] and *Communicating Sequential Processes* (CSP) [11].

[12] set out to define a set of primitives for long-running transactions in flow composition languages concerned with structured control flows. In other words, flows that can be defined in terms of primitives like sequencing and branching (choice). This gives a formal semantics to long-running transactions, whose flow of control exhibits these features.

The approach to modelling transactions described in [12] is driven by the understanding of transactions as in *Sagas* [3]. The *Sagas* model is one of the points of reference for long-running or long-lived transactions. Nevertheless it has been criticised for its limited applicability in conducting open e-business long-running transactions [13]. That said, *Sagas* was one of the first models for long-running transactions and some of the key concepts are still relevant in any transaction model. The key idea is that valid executions of a transactional business process (or of a part of it) are those that complete all involved activities. This is essentially the Atomicity part of ACID properties.

A more relaxed version of atomicity can be achieved by associating processes with compensating activities that can recover or 'undo' the effects of previously completed activities within a transaction (partial executions). The question of course is how the compensating mechanism can be applied to ensure consistency of data at all times and take into account the dependencies that may exist both inside and outside of a transaction. Therefore, our discussion of [12] focuses on this aspect in the remaining of this section rather than the inherent limitations of the *Sagas* transaction model (they have chosen to adopt) itself.

Any partial execution of a saga is undesirable – it must either execute all its sequential actions successfully or not at all – if an action fails during the execution of a saga then the whole saga must be compensated for. Formally, a saga involving A_1, …A_n (where each A_i is associated with a compensating action B_i) is guaranteed to execute either the sequence A_1; … ; A_n (successful execution of the whole saga) or the compensated sequence A_1; … ; A_j ; B_j; …. B_1 for some $j < n$ (in case the action A_j fails, execution stops and the compensating actions are executed in the reverse order).

The formalisation of sequential sagas in [12] comprises a syntax for writing sequential sagas and a semantics that determines the outcome of a saga and how their compensations are executed in the event of a failure. The approach of [12] considers the case that the compensating actions inside a saga (the B_j; … ; B_1 part of the execution of a saga) may also fail. Hence, there are three different potential outcomes: successful execution (*commit*); failure with successful compensation (*compensated abort*); and, failure with unsuccessful compensation (*abnormal abort*).

Of course, in any transaction, there may be actions that may be performed concurrently, as well as actions that must be performed sequentially. This is handled in [12] by introducing an operator '|' denoting the parallel composition of processes. Parallel composition is understood as in the CSP [11] where sequential processes can be executed in parallel so long as they synchronise (both are ready to engage in that action) on common actions while all non-common actions from each are interleaved (that is, they can occur in either order so long as the sequencing in the process they come from is preserved).

It is important to note that in the approach of [12] to modelling long-running transactions, processes that are composed in parallel do not communicate. In other words, they do not have common actions on which they have to synchronise. This means that their respective actions are all independent and hence composition simply results in all possible interleavings of the actions from each. A similar approach to parallel composition of sequential processes is taken in *compensating* CSP [14].

As for sequential sagas, a parallel saga terminates successfully only when all its actions commit. The whole saga (all participating sequential processes in the parallel saga) will be compensated when an action is aborted. The compensations for parallel sagas are executed concurrently (instead of in the reverse order as in sequential composition).

The approach considers initially a semantics for parallel sagas where the sequential processes that comprise a parallel saga are completely independent. This has the disadvantage that if one sequential process aborts, then there is no way to immediately enforce the rest of the sequential processes to abort too. This can only be done once the rest of the processes have completed their execution.

To alleviate this problem the authors in [12] subsequently introduce two new types of results for a process within a saga: the first denoting that the process has been forced to

compensate, and it has been compensated successfully; the second denoting that the process has been forced to compensate but the compensation itself has failed.

The fact that only sequential blocks of actions are considered, and these do not communicate whether they are composed in sequence or in parallel, means that the resulting formalism is not expressive enough to capture partial results (which would require communication between sequential processes) and is even limited in capturing the dependencies that arise inside a long-running transaction. The compensating mechanism used follows exactly that of the forward actions and there is no provision to capture omitted results or forward recovery (although they introduce a nesting scheme which could be used as a means of introducing alternative scenarios) and leaves little flexibility for extension with such features in the future.

In addition, there are certain problems that arise in compensating for parallel processes due to the fact that the compensation procedure should be independent of any particular interleaving of actions. This is not possible within CSP since the parallel composition operator is defined to do just that – interleave non-common actions in any way. The fact that there is no communication between sequential processes that are composed in parallel in [12] means that the parallel composition operator does no more than simply generate the non-deterministic interleavings of the actions from each process. In fact, the extension to CSP with compensations to produce the so-called *compensating CSP* [14], paradoxically considers a non-interleaving semantics in performing the compensations for sequential processes that are composed in parallel.

The extension of the CSP formalism [11] with compensating actions has been considered in [14], [15]. The resulting *compensating CSP* has been used for modelling long-running transactions, which are understood as a sequence of isolated activities; drawing again upon the concept found in *Sagas* [3]. Again, we find some difficulties with regard to our requirements for B2B transactions. In the compensating CSP framework for long-running transactions proposed in [14], processes that are composed in parallel are not allowed to communicate. In other words, the synchronous execution of observable actions is not supported. Hence, processes that are composed sequentially (using the ';' operator described above) do not communicate, so the proposed formalism does not support any form of communication or interaction between activities inside a transaction. Since transactions are modelled by processes, which do not communicate, there is also no communication across transactions. Thus, the proposed framework does not support the release of results inside or outside a transaction.

The compensating Communicating Sequential Processes of [14, 15] introduce a cancellation semantics for compensable processes in order to examine how the effect of forward actions is cancelled by compensating actions. Central to this cancellation theory is a notion of *independence* between actions. The use of an independence relation is central to well-known *non-interleaving* models for concurrency [16, 17, 18] which do not identify concurrent execution with the non-deterministic interleaving of actions.

The cancellation function in compensating Communicating Sequential Processes [14] is intended to remove matching pairs of action-compensating action from the resulting trace of the transaction or compensable process. Further, it can be seen that a compensable process may (and typically will) have more than one trace [19]. This may be due to the choice operator or due to the parallel composition operator which generates all possible interleavings of the actions from each process. This is precisely where the independence relation comes in. It can be used to filter out different traces that represent concurrent execution of the same actions, as done in non-interleaving approaches to concurrency.

We note that the use of an independence relation on an alphabet of actions is central to well-known *non-interleaving* models for concurrency [20, 18, 17] – in fact, independence is used in these models to describe potential concurrency. This is not mentioned in [14] but the essence of the independence relation in compensating Communicating Sequential Processes (CSP) is precisely the notion of independence used in non-interleaving models for true-concurrency. (A comprehensive survey of different concurrency models can be found in [21].)

In the case of compensating CSP, the appeal to a non-interleaving semantics in examining the relation between forward actions and their compensations can be seen as a means to recover more traces than adhering to the interleaving semantics laid out up to that point. This is only possible if there is a notion of equivalence between traces – the traces before and after cancellation must be equivalent. This is not the case with the current set up in compensating CSP. We will return to this point in the next section.

As a result of prohibiting any communication, the compensating Communicating Sequential Processes of [14], [15], have no provision for partial results but also, there is little evidence and little flexibility for capturing more than the dependencies due to ordering, i.e. due to data sharing between actions or sub-transactions. Note that there is no communication in either case – sequential or parallel composition – and thus there is no sharing of data or exchange of results inside of a transaction before it commits. This also raises the question of the relevance of a independence relation in the framework – all actions are independent by default: in sequence, they execute in isolation; in parallel, they are independent since there are no common actions and hence, no communication.

The only communication that is allowed is that of synchronising on terminal events of sequential processes that are composed in parallel. This however does not provide any leverage for covering partial results. Instead it prohibits the triggering of the compensating procedure in one process as soon as a failure is detected in some other process. In other words, it does not allow to enforce the abortion of one branch as soon as failure occurs in the other branch. This is not remotely satisfactory when modelling real problems which require activities within a transaction to be executed in parallel, since it may result in a situation where one processes fails relatively early in its execution and the other process (or, even

worse, processes) have to complete their execution until they are ready to commit in order to be notified (via synchronisation on terminal events) that some process has failed. The situation becomes even worse when considering that these processes will have to run their compensation for the whole set of actions they executed while the other process had already failed. It might be worth noting that this problem has been partly touched upon in the approach of [12], when the authors talk about a 'naïve semantics' to parallel composition (which had to be extended with a 'forced to compensate' notion).

StAC (Structured Activity Compensation) [22, 23], a language in the spirit of CSP with exception handling mechanisms, has been developed as a tool for compensating CSP. The objective was to develop a language that follows CSP, and considers compensations closely related to the control flow of the executed process. However, the execution of the compensations is not part of the definition of a transaction, but rather StAC has special primitives for activating the installed compensations. Consequently, compensations are not actually related to (or triggered by) the failure or success of the activities of processes, as usually expected in flow composition languages, e.g. BPEL4WS [24]. Moreover, as pointed out in [12], there is a tight relation between data structures used by activities and the control flow of processes in StAc. This means that reasoning about processes in StAC requires low-level description of activities which is in contrast with preserving the local autonomy of participating business' platforms. Furthermore, StAC comprises a large set of operators; more than what we have seen in compensating CSP [14]. The operational semantics of these operators is given in yet another intermediate language $StAC_i$ [23], in which some usual behaviours of compensations are only achieved through a – sometimes obscure – combination of several $StAC_i$ operators. Since operators in StAC can only be understood by analysing their encodings in $StAC_i$ it is difficult to reason about the interplay between compensations, nesting (if any) and parallel composition in StAC. The promise that such issues would be rectified in compensating CSP [14] has been, at best, moderately achieved – the improvement concerns the reasoning about the effects of a transaction in a compositional way, and in this way extend the semantics of StAC to make it compositional, while the aforementioned issues remain to be addressed.

Overall, although we see good progress in the above work, there are still a number of open issues within these formalisms – at least from the perspective of Digital Business Ecosystems. One interesting point for us was the late adoption of a non-interleaving semantics for concurrency in compensating CSP. In the next section, we will outline a formalism that takes a non-interleaving semantics as its foundation.

IV. A Formal Language for Long-running Transactions in DEs

In this section, we lay the foundations for a formal model of long-running transactions showing how the subtransactions (Local Coordinators and/or basic services) are orchestrated to achieve the goal of the transaction in question. In developing the formal model we have avoided following a specific formalism (such as CSP or CCS) so as not to be constrained from the outset by the limitations of a specific approach. In the first instance, this allows an understanding of long-running transactions that goes beyond that of *Sagas* [3] and we are not forced to consider a long-running transaction as a sequence of actions.

In earlier work [9], we have described a distributed transaction model for the digital ecosystem paradigm, in which networked organisations are expected to engage in complex transactions involving a number of subtransactions (internal activities) which need to be coordinated locally in terms of the underlying service compositions. In other words, have opted for an open nested transaction model in which a transaction is understood to comprise a nested group of subtransactions whose actions (or service executions) do not necessarily have to be sequential, as in compensating CSP [14] or the approach of [12] discussed previously.

In this section we report on work in progress in providing a formal foundation for the proposed model of long-running transactions and the distributed orchestration of the underlying service compositions. Our formal semantics of long-running transactions is aimed at describing the behavioural patterns services should follow in order to guarantee successful commitment or compensation within the transaction flow manager. The proposed formal model for transactions uses ideas taken from a variety of theories for describing the behaviour of communicating systems, from Shields' vector languages [16, 20, 17] to Mazurkiewicz traces [25, 18] to event structures [26] to process algebras [10, 11]. It draws upon a vector language-based description of behaviour, which allows monitoring or recording a number of communicating entities at the same time (groups of subtransactions), and has most recently been applied to modelling interactions between components of a (distributed) system in [27]. This theory is adopted here to underpin the local coordination required for long-running multi-service transactions in a digital ecosystem.

In our model for long-running transactions in digital ecosystems, described in detail in [19], a transaction is represented by a tree structure that allows us to exemplify the local coordination that is required for the services involved to be performed in unison in accomplishing the goal prescribed by the transaction. In fact, at the heart of our transaction model are the Local Coordinators. This is where most complexities of a transactional environment are handled such as coordination, service orchestration, keeping logs for managing dependencies and implementing a recovery mechanism, but also this is where we would expect that handling the low bandwidth or dealing with the low processing power of some nodes in the underlying P2P network, would take place.

Based on the latest work on an extended service-oriented architecture for a business environment [28, 29, 30], five different types of coordinators have been considered in our model for, namely a data-oriented coordinator, a sequential process-oriented coordinator, a parallel process-oriented coordinator, a sequential-alternative coordinator a parallel-alternative coordinator and a delegation coordinator.

A. Transaction Vectors: a language-based representation

We may now start to describe long-running transactions more formally. Apart from the dependencies there is a high degree of concurrency in a transactional environment. We will find the general theory of non-interleaving representation of parallel behaviour found in [17] of great use in what follows. As mentioned before, our objective is to get a thorough understanding of the behaviour the underlying service compositions needs to exhibit for successful commit or compensation of the transaction as a whole.

In our behavioural model of a transaction it suffices to use formal notation for the leaves only. The aggregation coordinators (nodes) are manifested in the structure of the resulting formal construction, and there is no need for additional notation. A transaction T, then, is associated with a set of leaves L which consists of a set of basic services S, a set of data-oriented coordinators D and a set of delegation coordinators Dlg. Hence, $L = S \cup D \cup Dlg$. We further require that the sets S, D and Dlg are pairwise disjoint.

In this section we introduce a formal language for describing long-running transactions. The semantics is intended to describe the behaviour of a transaction in terms of its services at the deployment level, but not the low-level computations performed by the services themselves. Note that services are offered in a digital ecosystem for business from different service providers and it is important that we defer from interfering with the local state of the service execution. The service-oriented architecture for distributed transactions reinforces our interest in all environmentally observable actions inside and outside a transaction. That means it is appropriate to consider that any action within the transaction model has no significant duration, in the sense that (i) it either occurs as a whole or not at all; (ii) it occurs either wholly before, or wholly after, or wholly in parallel with, every other action.

A transaction may thus be associated with a finite set of events or *significant events* [31] or *actions* that may occur (on its subtransactions) upon activation, e.g. service invocation, initialisation, commitment, service return, release result (return), termination, abort, etc. We denote this set of actions of a transaction by M.

These actions take place on the leaves and therefore it seems appropriate to say that each leaf is in turn associated with a set of actions that may occur on that leaf, depending on its nature. We denote this set by $\mu(l)$, $l \in L$, and require that $\bigcup_{l \in L} \mu(l) \subseteq M$.

In any behaviour of a transaction T, each subtransaction on the leaves will be activated and experience a sequence of actions formed over the corresponding set $\mu(l)$, $l \in L$. This means that there a number of activation points within a transaction – essentially these are all its leaves. Following the idea that originates in Shields' vectors [16], which was subsequently extended to a more general theory of communicating systems in [17], we may describe the behaviour of a transaction by assigning such sequences to each of its leaves. This results in the so-called *transaction vectors* defined below.

Definition 1. (**Transaction vectors.**) Let T be a transaction. We define V_T to be the set of all functions $\underline{v}$: $L \rightarrow M^*$ such that $\underline{v}(l) \in \mu(l)^*$. We refer to elements of V_T as *transaction vectors*.

By $\mu(l)^*$ we denote the set of finite sequences over $\mu(l)$. Mathematically, the set V_T is the Cartesian product of the sets $\mu(l)^*$, for each l. Effectively, transaction vectors are n-tuples of sequences where each coordinate corresponds to a leaf in the transaction tree (hence, n is the number of leaves) and contains a finite sequence of actions that have occurred on that leaf.

When an action occurs on a leaf of the transaction tree, that is to say when an action associated with some subtransaction takes place, it appears on a new transaction vector at the appropriate coordinate. For example, the vector

(s_1, Λ, Λ)

describes that portion of behaviour of the transaction in which an action s_1 (e.g. service invocation) has taken place on the corresponding service allocated to the first coordinate. The vector

(s_1, s_2, Λ)

describes that portion of behaviour in which both s_1 and s_2 have happened on the corresponding services while the vector

$(s_1 s_3, s_2, \Lambda)$

describes an occurrence of s_1 and an occurrence of s_3.on the service corresponding to the first coordinate, and an occurrence of s_2 on that of the second coordinate. Nothing has happened on the service corresponding to the third coordinate.

In this sense, each transaction vector provides a *snapshot* of behaviour in which the transaction has executed the actions appearing on the vector's coordinates – the vector tells us what actions have already occurred and on which part of the transaction tree.

This vector-based description of behaviour allows recording the actions of a transaction as these occur on the multiple services involved in the execution of the transaction. Readers familiar with process algebras like CSP or CCS can understand each particular coordinate of the vector description as a sequential CSP process. In this sense, the transaction vectors can be understood as the Cartesian product of sequential processes describing each leaf in a transaction tree.

It can be seen from the examples given above that there is already an ordering among actions on a particular subtransaction (e.g. s_1 followed by another s_1). This vector-based behavioural description of transactions can also capture the orderings between different subtransactions, which amounts to actions appearing on different vector coordinates. This requires however a more careful consideration of the mathematical properties of such vectors which we briefly describe in the following section.

Before examining the mathematical properties of our construction so far, we introduce a specific kind of transaction

vectors, which is used in our model to describe actions (events or activations) within a transaction.

Definition 2. (Column vectors.) Let T be a transaction and V_T its set of transaction vectors. We define

$$A_T = \{\underline{a} \in V_T \setminus \{\underline{\Lambda}_T\} : l \in L \Rightarrow | \underline{a}(l) | \leq 1\}$$

where $|x|$ denotes the length of sequence *x*. We refer to elements of A_T as *column vectors*.

Thus, the vectors of Definition 2 are themselves transaction vectors, but have the additional constraint that each of their coordinates is either the empty sequence or a single action. For example, the vector (s_1, Λ, Λ) represents the occurrence of an action s_1 on the sub-transaction associated with the first coordinate.

We will use the term *transaction language* to refer to a subset V of all possible vectors V_T formed over a given transaction T. Hence, a transaction T comes with a language V, where $V \subseteq V_T$. The idea is that the particular set of transaction vectors for a specific transaction expresses the ordering constraints necessary in the corresponding service orchestration.

B. Order-theoretic properties of transaction vectors

We now describe the basic order-theoretic properties of transaction vectors since this is what allows us to define operations on vectors. These are important in determining the coordination of the transaction in terms of its underlying service invocations. We will see how the order structure of sets of such vectors expresses ordering constraints on actions inferred by the execution of the various subtransactions inside a long-running transaction. A detailed mathematical treatment of vectors can be found in [17]. We have seen that transaction vectors are essentially tuples of sequences. This can be exploited in defining operations on the vectors in terms of well-known operations on sequences.

First, let us establish our notation. If x and z are sequences, we write x.z for the concatenation of x and z. As is well known this operation on sequences is associative with identity Λ, where Λ denotes the empty sequence. We also have a partial order on sequences given by x ≤ z if and only if there exists a sequence y such that x.y = z, and this partial order has a bottom element Λ. It is also well-known that the operation '.' is cancellative, which means that if x ≤ z, then the sequence y such that x.y = z is unique. We shall denote this sequence by z / x. Finally, recall that if x, y, z are sequences such that x, y ≤ z, then either x ≤ y or y ≤ x.

We may now lift these well-known operations on sequences onto transaction vectors. This is done formally in the following definition.

Definition 3. (Operations on vectors.) For $\underline{u}$, $\underline{v} \in V_T$, we define

- $\underline{u}.\underline{v}$ to be the unique vector $\underline{w}$ such that $\underline{w}(l) = \underline{u}(l).\underline{v}(l)$, for each $l \in L$ (*concatenation*)

- $\underline{u} \leq \underline{v}$ iff $\underline{u}(l) \leq \underline{v}(l)$, for each $l \in L$ (*prefix ordering*)

- *glb*($\underline{u}$,$\underline{v}$) to be the vector $\underline{w}$ such that $\underline{w}(l)$=min($\underline{u}(l)$,$\underline{v}(l)$), for each $l \in L$

- *lub*($\underline{u}$,$\underline{v}$) (if it exists) to be the vector $\underline{w}$ such that $\underline{w}(l)$=max($\underline{u}(l)$,$\underline{v}(l)$), for each $l \in L$

- if $\underline{u} \leq \underline{v}$, then we define $\underline{v} / \underline{u}$ to be the unique $\underline{z} \in V_T$ such that $\underline{u}.\underline{z} = \underline{v}$ (*right-cancellation*)

Thus, the operation of concatenation on vectors is defined in terms of the concatenation of sequences appearing on their respective coordinates. For example,

$$(s_1 s_3, s_2, \Lambda).(\Lambda, s_4, \Lambda) = (s_1 s_3, s_2 s_4, \Lambda)$$

Transaction vectors can be seen to be built up from the empty vector $\underline{\Lambda}_T$ by a series of concatenations with column vectors (Definition 2) that represent actions. In fact, in describing the behaviour of a transaction we are interested only in those vectors describing (orderings of) actions that we expect the transaction to engage in during the course of its execution. This is the subset of all possible transaction vectors, over a given T, we referred to as *transaction language*.

The operations *glb*() and *lub*() of Definition 3 give the greatest lower bound and the least upper bound, respectively of $\underline{u}$, $\underline{v} \in V_T$, in the usual sense of lattices and domain theory [DaP90]. These operations are central to the treatment of concurrency in our approach and also have an important role to play in defining the properties that ensure the well-formedness of the behavioural description.

It can be shown (by an adaptation of the proof found in [27], which is in turn based on that originally perceived in [17]) that a set of transaction vectors equipped with the operations of concatenation and prefix ordering of Definition 3 forms a monoid[1] and a partial order. $\underline{\Lambda}_T$ is used to denote the empty vector which has the empty sequence on each of its coordinates. The incomparable vectors in the partial order (V_T, ≤) allow to introduce a notion of independence between transaction vectors, which is central to expressing true-concurrency within our model. This builds on earlier work on describing parallel behaviour in Shields' *behaviour vectors* [17] where the notion of independence found in *Mazurkiewicz traces* [18] is lifted onto vectors. This development is central to the modelling concurrent actions of a long-running transaction.

C. Well-formedness of the behavioural description of a transaction

In describing the behaviour of transaction we are interested in the actions (activations) on its sub-transactions. These are captured in our model using column vectors (Definition 2). Thus, instead of considering all possible transaction vectors we would like to be concerned with those obtained by concatenations with column vectors only. This gives us the

[1] Recall that a monoid is a semi-group with identity.

behaviour of the transaction in terms of activations or actions of its sub-transactions and can be used to enforce the coordination of the underlying services.

We have seen that transaction vectors are obtained by coordinate-wise concatenation (Definition 3), for example

$$(x_1, x_2, x_3).(y_1, y_2, y_3) = (x_1 y_1, x_2 y_2, x_3 y_3)$$

In such a behavioural description of a transaction, transaction vectors can be seen to be built up from the empty vector by a series of concatenations with the column vectors [27], each of whose coordinates is either empty or contains a single event/action. The study of vector languages in [17, 32] shows that in order to ensure that vectors associated with a transaction are the result of concatenations with column vectors only, the set of transaction vectors must satisfy certain properties, namely *discreteness* and *local left-closure*.

These properties are discussed in detail in [19]. Essentially discreteness imposes a finiteness constraint in the sense that it excludes infinite ascending or descending chains of actions with respect to time ordering. It ensures that situations like those resulting in Zeno-type paradoxes will never arise. Local left-closure is intended to resolve ambiguities that may arise from not having enough vectors in the transaction language to describe the course of the behaviour in question; not the start or the end, but the 'gaps' in between. This requires that every occurrence of an event is 'recorded' in the language of the transaction. discreteness and local left-closure ensure the well-formedness of the behavioural description of a transaction in our model. The idea is that in checking against these properties we may determine whether the transaction will exhibit the desired behaviour when executed or on the contrary, other non-desirable scenarios of execution are still possible. This draws upon previous work on vector languages and UML sequence diagrams in [Mos05].

D. *Where does this take us?*

Our approach towards modelling concurrent actions, actions that can happen in parallel, draws upon the concepts in *Shields' vector languages* [16, 17] and *Mazurkiewicz trace languages* [Maz77], [Maz88] where the ordering of concurrent events is considered subjective and thus is not distinguished, in contrast to CSP trace theory where it is assumed that observations are sequential in nature leading to the interpretation that concurrent events occur in either order.

Our notion of concurrency this takes up on Mazurkiewicz traces [25, 18], which introduce additional structure into formal languages in order to describe non-sequential behaviour. The additional structure is given in terms of an independence relation over action symbols, which describes potential concurrency. By departing from classic CSP concurrency, we are able to consider concurrency within a long-running transaction, and without the need to consider sequences of actions within a transaction as in compensating CSP [14]. In CSP, and related process algebras, concurrency arises through composition. Here we have not yet been concerned with composing sequences from subtransactions of different transactions, though this may also produce concurrency. We are simply describing the case that subtransactions of the same transaction engage in concurrent actions, a phenomenon common in most B2B scenarios for example.

V. Further Results and Conclusions

We have focused on the foundations of our model in the previous section. [19] contains an extensive discussion of the model, in particular demonstrating its handling of compensation and forward recovery.

Overall, by avoiding the constraints of an existing formalism, and by exploiting a non-interleaving semantics of concurrency, we believe we are able to offer a rich, yet notationally concise model.

Acknowledgment

This work has been funded by European Commission through the DBE and OPAALS projects.

References

[1] C. J. Date *An Introduction to Database Systems*. 5th Edition, Addison Wesley, USA, 1996.

[2] A. Elmagarmid *Database Transaction Model for Advanced Applications*, Morgan – Kaufmann, 1994.

[3] H. Garcia-Molina and K.Salem "Sagas", Proceedings of the Association for Computing Machinery Special Interest Group on Management of Data 1987 Annual Conference, San Francisco, California, May 27-29, 1987, pp: 249-259.

[4] F. L. Cabrera, G.Copeland, J. Johnson and D.Langworthy, Coordinating Web Services Activities with WS-Coordination, WS-AtomicTransaction, and WS-BusinessActivity. Available at: http://msdn.microsoft.com/webservices/default.aspx, January 2004.

[5] Cabrera, L. F., Copeland, G., Cox, W., Feingold, M., Freund, T., Johnson, J., Kaler, C., Klein, J., Langworthy, D., Nadalin, A., Orchard, D., Robinson, I., Shewchuk, J., Storey, T. and Satish Thatte. Web Services Coordination Framework (WS-Coordination), September 2003.

[6] Furnis, P.; Dalal, S.;Fletcher, T.; Green, A.; Ceponkus, A. and Pope, B. Business Transaction Protocol version 1.1.0, Committee Draft, November 2004. Available at: http://www.oasis-open.org/committees/download.php/9836/business_transaction-btp-1.1-spec-wd-04.pdf

[7] Cabrera L. F., Copeland G., Cox W., Feingold M., Freund T., Johnson J., Kaler C., Klein J., Langworthy D., Nadalin A., Orchard D.; Robinson, I., Shewchuk J., Storey T. and S. Thatte. (2004-b) "Web Services Business Activity Framework (WSBusinessActivity)", January 2004

[8] Razavi R.; Krause, P.J.; and Moschoyiannis S.; Digital Business Ecosystem Transaction Model. DBE Project Report D24.28, University of Surrey, 2006.

[9] Razavi R.; Moschoyiannis S.; and Krause, P.J. Preliminary Architecture for Autopoietic P2P Network focusing on Hierarchical Super-Peers, Birth and Growth Models. OPAALS project Deliverable D3.1, 2007.

[10] Milner A.J.R. Calculus for Communicating Systems. Lecture Notes in computer Science, Vol. 92, Springer Verlag, 1980.

[11] Hoare C. A. R. *Communicating Sequential Processes*. Prentice Hall, 1985.

[12] Bruni, R.; Melgratti, H.; Montanari, U.; Theoretical Foundations for Compensations in Flow Composition Languages. In *Proceedings of ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL 2005)*, pp. 209-220, ACM Press, 2005.

[13] Furnis, P. and Green, A. (2005), "Choreology Ltd. Contribution to the OASIS WS-TX Technical Committee relating to WS-Coordination, WS-AtomicTransaction and WS-BusinessActivity", OASIS WS-TX Technical Committee, 16 November 2005. (avalaible at www.oasis-open.org/committees/download.php/15808/Choreology.WS-TX.TC.Contribution.2005-11-16.doc , last availability; 06.June.2006).

[14] Butler M. J, Hoare A. C. R. and Fereira C. A Trace Semantics for Long-Running Transactions. *Proceedings 25 Years of CSP*, Lecture Notes in Computer Science, Vol. 3525, pp. 133-150, Springer, 2005.

[15] Ripon, S.; Butler, M.; Relating Semantic Models of Compensating CSP. Technical Report, School of Electronics and Computer Science, University of Southampton, 2006.

[16] Shields, M. W.; Adequate Path Expressions. In *Proc. Semantics for Concurrent Computation*, Lecture Notes in Computer Science, Vol 70, pp. 249-265, Springer Verlag, 1979.

[17] Shields, M. W.; *Semantics of Parallelism*. Springer-Verlag, London, 1997.

[18] Mazurkiewicz, A. (1988) Basic Notions of Trace Theory. In de Baker, de Roever and Rozenberg, eds, *Proceedings of Linear time, Branching Time and Partial Orders in Logics and Models for Concurrency*, Lecture Notes in Computer Science, Vol. 354, pp. 285-363, Springer-Verlag, 1988.

[19] Razavi R.; Moschoyiannis S.; and Krause, P.J. *Report on formal analysis of autopoietic P2P network, together with predictions of performance*. OPAALS project Deliverable D3.2, 2007.

[20] Shields, M. W. Concurrent Machines. *Computer Journal*, 28:449-465, 1985.

[21] Winskel, G.; Nielsen, M.; Models of Concurrency. In S. Abramsky, D. Gabbay, T. Maibaum, eds, *Handbook of Logic in Computer Science, Vol. 4, Semantic Modelling*, pp. 1-148, Oxford Science Publications, 1995.

[22] Butler, M.; Chessell, M.; Ferreira, C.; Griffin, C., Henderson, P. and Vines, D.; Extending the Concept of Transaction Compensation. *IBM Systems Journal*, 41(4):743-758, 2002.

[23] Butler, M. and Ferreira, C.; An Operational Semantics for StAC, a Language for Modelling Long-running Business Transactions. In *Proc. Coordination 2004*, Lecture Notes in Computer Science, Vol. 2949, pp. 87-104, Springer, 2004.

[24] Business Process Engineering Language (BPEL) Specification, version 1.1. Availabel at: http://www.ibm./com/developerworks/library/ws-bpel

[25] Mazurkiewicz, A. Concurrent Program Schemes and their Interpretation. Technical report DAIMI PB-78, Aarhus University, 1977.

[26] Nielsen, M.; Plotkin, G.; Winskel, G.; Petri Nets, Event Structures and Domains, part 1. *Theoretical computer Science*, 13:85-108, 1981.

[27] Moschoyiannis, S. Specification and Analysis of Component-Based Software in a True-Concurrent Setting. PhD Thesis, University of Surrey, 2005.

[28] Yang, Jian; Papazoglou, M P. and Heuvel, W-J van den (2002), "Tackling the Challenges of Service Composition in E-Marketplaces", Twelfth International Workshop on Research Issues in Data Engineering: Engineering E-Commerce/E-Business Systems, 2002. RIDE-2EC 2002 (IEEE Computer society), pp:125-133.

[29] Papazoglou M. P. (2003) , "Service-Oriented Computing: Concepts, Characteristics and Directions", 4th International Conference on Web Information Systems Engineering (WISE'03) , Rome, Italy, 2003.

[30] Papazoglou M. P., Traverso P., Dustdar S., Leymann F. and Kramer B. J. Service-Oriented Computing Research Roadmap. In *Dagstuhl Seminar Proceedings 05462, Service-Oriented Computing (SOC)*, pp. 1-29, 2006. Available at: http://drops.dagstuhl.de/opus/volltexte/2006/524 [Last accessed: 19 Apr 2006]

[31] Singh, M.; A Customizable Coordination Service for Autonomous Agents. In *Proc. 4th Int'l Workshop on Intelligent Agents – Agent Theories, Architectures and Languages*, Lecture Notes in Computer Science, Vol. 1365, pp. 93-106, Springer-Verlag, 1997.

[32] Shields, M. W. and Moschoyiannis, S.; An Automata-Theoretic View of Software Components. Technical Report SCOMP-TC-02-04, Department of Computing, University of Surrey, 2004. Available at: http://www.computing.surrey.ac.uk/personal/st/S.Moschoyiannis/

Interpretation Logics

[OPAALS Conference 2007]

Ossi Nykänen
Institute of Mathematics, Hypermedia Laboratory
Tampere University of Technology
Tampere, Finland
ossi.nykanen@tut.fi

***Abstract*—Integration of incompatible axiomatic theories is a concrete obstacle in developing logic-based applications. This challenge is faced, e.g., when integrating legacy databases, rule-based knowledge systems, and domain-specific ontologies. Further, many applications do not exploit the full range of properties of the underlying logic languages that typically leads into unnecessary complex, implicit overhead in reasoning. We propose a generic framework called interpretation logics for integrating mismatching knowledge bases, and explicitly designing the related interpretation system. We claim that the integration of formal theories based on a linguistic transformation and application-specific, asserted interpretation axioms, with respect to interpretation logic, is not only feasible but also desirable, due to the common challenges of application development. We outline the conceptual framework, demonstrate its applicability with an example, and discuss implementations with respect to Semantic Web technologies.**

Interpretation logics, Knowledge transformation, Semantic interoperability

I. INTRODUCTION

Logic provides a system for mechanically working with knowledge. The usefulness of logic builds upon the two fundamental tasks of mathematics: the descriptional task and the deductional task [10]. The abstract knowledge engineering process captures the motivation of computerized knowledge systems well: decide what to talk about; decide on a vocabulary of predicates, functions, and constants; encode general knowledge about the domain; encode a description of the specific problem instance; and pose queries to the inference procedure and get answers [21]. While this is straightforward in principle, serious problems usually arise in applications, at least when different systems are to be integrated.

Despite the evident hype, logic-based models are today commonly found in mainstream applications such as software engineering, configuration, medicine, digital libraries, and Web-based information systems [2]. Prominent new areas of work include applications for web portals, multimedia collections, corporate web sites, design documentation, agents and services, and ubiquitous computing [8]. Visions of modern applications range from publishing business rules to discover the potential of enterprise collaborations to aggregating the family photo albums [19][23].

Indeed, the standardization of the representation languages and the ease of publishing and accessing data over the Internet, have made it possible to compile and semantically integrate logical data and exploit it in a variety of local applications. However, strict integration of different axiomatic and slightly incompatible theories soon becomes an obstacle in practical application development. This happens, e.g., when integrating legacy systems, or when combining rule-based knowledge systems and domain-specific ontologies.

In this article, we present a generic framework called interpretation logics for integrating a series of available, mismatching knowledge bases, and explicitly designing the related interpretation system. Intuitively, interpretation logics may be perceived as an abstract design pattern, with logic-specific characteristics of more general interfaces and proxy design patterns (see, e.g., [6]). Due to its popularity in software engineering, the study of design patterns is emerging in several research topics, including ontology design (see, e.g., [1]).

The basic idea is that instead of mixing the available logical data within a single "rich enough" logical framework, applications should be based on explicit design decisions of what formulas to include and with what logical axioms. Thus, the interpretation theory is not simply a "passive" mixture of the theories of the available knowledge bases, but an "active" interpretation of them, involving application-specific choices.

The main motivation for our work stems from the research needs of a large European Network of Excellence Project OPAALS (Open Philosophies for Associative Autopoietic Digital Ecosystems; see http://www.opaals.org/) coordinated by London School of Economics and Political Science. The overall objectives of the OPAALS project are to build a sustainable interdisciplinary research community in the emerging area of Digital Ecosystems (DE, http://www.digital-ecosystems.org/) and to develop an integrated theoretical foundation for DE research. In the project, the problem of integrating mismatching knowledge bases is faced in two major research activities: language evolution and visualization. In the former research activity, interpretation logics provides a method for working with bottom-up ontologies, i.e., interpretation theories built with the contextual organization of the available knowledge bases. In addition, interpretation logics provide a concrete framework for studying the integration of domain ontologies and business rules. In the latter activity, interpretation ontologies (or theories) are used for mapping the available knowledge bases onto visualization models [18]. This allows abstracting the concepts of the few general-purpose

This work has been supported by the European Comission (IST network of excellence project OPAALS of the sixth framework program, contract number: FP6-034824).

visualization models from the numerous specific-purpose knowledge models.

We claim that working with interpretation logics not only provides a concrete, easy-to-deploy framework for integrating logical models (because tools can be chosen more freely), but also significantly clarifies application development. This is due to the fact that designers are allowed to make transparent design decisions about the reasoning process in local applications. This in turn helps making designs more understandable, at least by providing an intermediate venue for pinpointing the potential misunderstandings.

In principle, one can easily argue against this approach by saying that interpretations based on transformations are not needed since there are several good and standardized general-purpose knowledge representation languages. However, a short tour on the Semantic Web applications (see, e.g., [9][3]) of Resource Description Framework (RDF), Web Ontology Language (OWL) and Semantic Web Rule Language (SWRL), reveals that there are known practical problems with the "one size fits them all" approach. For instance:

1. Useful knowledge representation languages have often been developed in parallel, without a tight integration of the theory or the tools. For instance, the interplay of rule systems and ontology systems in reasoning applications is often defined only ad hoc. Consider, e.g., the management of the asserted and the inferred knowledge in Protégé with a DIG-compliant reasoner (for an overview of Protégé and OWL, see, e.g., [13][11]).
2. Integrating legacy data with ontologies is often challenging, due to the fact that the ontology languages seriously limit the use of expression of the language. For instance, OWL DL ontologies do not allow user-defined properties between individuals and classes [22].
3. Standard logics may include axioms that are not needed for every application. For instance, the Jena reasoner provides an explicit option of omitting some of the usually non-needed but expensive consequences of reasoning [20].
4. There are design errors and inconsistencies in the application theories. For instance, the RDF Schema is still commonly misused for "checking predicates' constraints" rather than asserting the valid consequences of their usage.
5. The availability of correctly working reasoning tools is a concrete problem; the available tools typically support only few language/theory/logic combinations. For instance, at the time of writing, the standardization efforts for a Semantic Web rule language are still underway [7].
6. In many cases, the modeling decisions of the legacy systems are fundamentally incompatible which means that explicit and constructive transformation processes are needed at any case. For instance, integrating local models typically requires actions related to the global naming of resources.

The above kinds of restrictions have led into introducing documented workarounds. For instance, the relatively simple restriction 2 above, which follows from the fact that OWL DL models classes as sets, is explicitly dealt as a Semantic Web best practice on representing classes as property values [16].

As suspected, the notion of interpretation logics is not tied to any particular logic system. However, for our purposes, implementing interpretation logics as Prolog programs has turned out to be a good compromise. There are several established tools available for various platforms and embedding Prolog interpreters to applications is a common practice. The need for using, e.g., Prolog tools explicitly may well change as the standard Semantic Web tools mature. Nevertheless, the availability of tools is an important issue in application development.

The interplay of ontologies and rule systems also seems to support this design decision. While it is generally agreed that, e.g., ontology systems and rule systems genuinely complement each others (neither subsumes the other as a logic unless we choose "unusually rich" ontology or rule systems), Prolog seems sufficiently expressive for capturing many useful OWL (DL) ontologies (see, e.g., [24]).

Because of its expressive power and availability, Prolog has also been suggested as a implementation strategy for the SWRL rules [15]. Popular Prolog systems, such as the SWI-Prolog, also include readily-available packages for managing Semantic Web data [25]. Wielemaker's and his colleagues work also includes, e.g. implementations of Semantic Web query languages and ontology editors [26][27]. As such, SWI-Prolog has been used for implementing Semantic Web reasoning in successful large-scale projects such as the MuseumFinland [12]. The specific challenge of managing heterogeneous ontologies has been extensively studied, e.g., in terms of ranked mappings of descriptions of resources onto OWL DL ontologies with appropriate complexity [4].

Finally, while implementing declarative logic programs is relatively straightforward, the computational complexity of logical queries is typically very high. Indeed, while the notion of interpretation logics might be perceived more as a design pattern than a theory, application developers face the practical problem of writing both expressive and sufficiently fast logic programs. Because of the theoretical and practical significance, the complexity and expressive power of logic programming and description logics has been extensively studied (see, e.g., [2][5]). The results indicate that while tractability and completeness can be achieved with suitable restrictions, the complexity of practical applications tends to be non-polynomial. Intuitively this means that large applications slow down very quickly. While interpretation logics can not obviously solve this problem, their explicit nature enables clarifying the complexity of applications. The basic method is including only the particular deductions (whose expected complexity is known and accepted) that are explicitly required by the target application.

The rest of this article is organized as follows: Section 2 describes the basic concepts and general definitions in more detail, by depicting interpretation logics in a relationship with a series of given initial knowledge bases. The basic idea of applications is then illustrated with an example in Section 3, with respect to specific Semantic Web technologies. Section 4 finally concludes the article with technical remarks and discussion.

II. Basic Concepts

In order to be able to discuss interpretation logics, we need to establish two main concepts: logic and theory. Interpretation logics and theories build upon these concepts by providing an explicit venue of language transformations, asserted interpretation axioms, interpretation systems, and finally concrete implementations.

A. Characterization of Logic Systems

We call a system L a logic (system) if it introduces and associates the following three meaningful components:

1. L establishes an object language of well-formed formulas (wff), denoted by F(L).
2. L introduces an interpretation system R(L).
3. L explains its concepts and reasoning process in terms of a meta language M(L).

The language F(L) serves as the principal medium of capturing asserted knowledge with respect to L. If F(L) is a formal language we may consider F(L) as a potentially infinite set of strings over a finite alphabet, usually defined in terms of a finite grammar. Examples of wffs include ground wffs such as facts and rules, and schematic wffs, typically intuitively simple string-rewriting functions. Schematic wffs are usually operated in the meta language, conditionally outputting ground wffs "upon request", based on valid parameters. (In implementations, schematic wffs might be implemented in terms of macros or pre-processing systems.)

The interpretation system R(L) essentially provides a way of reasoning, i.e. finding out whether a set of wffs B intrinsically follows from another set of wffs A (often denoted $A \rightarrow B$ or $A \supset B$). The reasoning system typically includes an algebraic component that allows equating and thus simplifying wffs that appear as different words in the language but have the same meaning with respect to the interpretation system. When R(L) is expressed formally, it typically introduces a set-theoretic or truth-functional interpretation theory which aims justifying, e.g., the soundness and completeness of the inference method. Further, the tractability and costs of inference may be explained with respect to worst-case or expected value estimates of computational complexity.

The meta language M(L) is needed for humans for communicating and analyzing the former two components in a sufficiently clear and understandable way. In most cases, meta language is some natural language such as English, equipped with some mathematical notation.

Note that according to the above characterization, a system may be considered "logical" even if it is not "correct", "consistent", etc.

We are now ready to define our second main concept, theory. A theory T_L with respect to a particular logic L, is simply a collection of wffs, $T_L \subset F(L)$. Further, when a theory T_L, is interpreted, we may think that the underlying logic L establishes a subsumed (and often potentially infinite) set of implied wffs that are not explicitly asserted by T_L. (Note that interpretation systems typically explain this behavior in terms of models of potential worlds rather than linguistically with the implied wffs.) Any changes in the theory or in the interpretation system usually change the set of implied wffs of the theory.

In short, a theory provides a logical justification for both the explicitly asserted wffs (which can be usually read from the set T_L when the theory is finite) and the implied wffs (whose existence needs to be verified with proofs). While the set of wffs is in principle arbitrary, useful theories are usually used to describe complex applications with few and understandable wffs, providing a concise and consistent system for making rigorous but rich deductions. While the concept "truth" is often found in the semantics, a theory does not really need to make particular ontological commitments. A theory simply captures the aspects of the application the designers have found useful.

A logic and hence a theory may be considered formal or informal. The notion of formality typically accounts to the level of detail and consistency in describing the object language and the method of inference. Examples of well-known formal logics include propositional logic, first-order predicate calculus (and the related logics such as pure and full predicate logic, modal logic, temporal logic, fuzzy logic, and probabilistic logic), RDF Schema, and OWL (and its sublanguages OWL Lite, OWL DL, and OWL Full) (for a list of core Semantic Web references, see [9]). Explicit rule languages that could be accounted as logics include e.g., abstract Horn Clause Logic and concrete systems such as Datalog, Prolog, SWRL, and their numerous relatives (see, e.g., [5] [15]).

Not every useful logic need to be considered formal, though. The best known examples of useful informal logics include common-sense reasoning and, e.g., semi-formal systems that operate with formally defined wffs with a reasoning procedure but without a detailed or an written out interpretation theory. Note that semi-formal and formal systems may in effect provide similar kinds of reasoning services, e.g., subsumption and satisfiability, even if the "quality" of reasoning might not be explicitly known in the case of the semi-formal systems. In practice, many kinds of algebraic structures, string-rewriting systems, and automata may be considered semi-formal logics in this sense. With some limitations, we may also conceptualize e.g. mathematical proofs as semi-formal logic systems, even if the object language is not usually fully formalized and reasoning depends heavily on structures and mathematical intuition expressed only on the level of the meta language.

The reasons for working with semi-formal logics vary. Perhaps the most common motivation for not aiming for a

completely formal theory is the complexity of the task, the estimated cost/gain ratio of the effort, and the fact that the occasionally resulting inconsistencies due to poor design may be tolerable in applications. As a consequence, formalization often results only after the informal approach has for some reason been found insufficient in application development.

For purposes of this article, it is interesting to record two well-known observations.

1. Theories do not usually exploit all of the available structures of the object language, nor require all the aspects of the interpretation system.
2. Applications that claim to work with a particular logic, are sometimes in fact working with a (slightly) modified logic, perhaps adding or removing some wffs considered as axioms.

The first point implies that in some cases, theories could be captured with a simpler logic. For instance, certain applications of full predicate logic may be expressed in suitable description logic. The second point implies that in some cases this might also be unintentional, e.g., due to misunderstandings. For instance, a logical theory may include a contradiction or convenient extra axioms whose rigorous implications (usually nullifying the usefulness of the theory in a strict sense) are neglected due to the lack of systematic consistency checking (tools).

We claim that both of these issues are faced commonly in application development. While these observations might simply be considered as poor modeling decisions (using an overly complex modeling language), or errors in reasoning (not conforming to the claimed logic), they hint the existence of useful interpretation logics, in which these selections are made knowingly, as active design decisions.

B. Interpretation Logics and Interpretation Theories

Assume a list semi-formal or formal theories $\langle T_i \rangle$ with the associated logics L_i. Let us call these the initial knowledge bases.

Define an interpretation theory T_I (with respect to an interpretation logic L_I) as follows:

1. Assume L_I is a suitable logic.
2. Establish a list of language transformations g_i that filter and adapt the interesting formulas from the theories of $\langle L_i \rangle$ to the object language of L_I.
3. Assert a suitable set of wffs denoted by D, and set interpretation theory $T_I = \cup_i g_i(\langle T_{Li} \rangle) \cup D$. The set $D \subset T_I$ is called the set of interpretation axioms of T_I.

An interpretation logic L_I is defined in association with an interpretation theory, by establishing $R(L_I)$ and $M(R_I)$ so that L_I appropriately captures the intended reasoning application.

In applications, the interpretation logic is typically selected first with a rough idea of its reasoning capabilities. Then, an interpretation theory is compiled and enriched with appropriate axioms. Finally, the interpretation of the interpretation logics is refined to match the needs of the target application.

The usefulness of the above definitions is based on the idea that useful knowledge bases include wffs that can provide a rough basement for the application-specific formulas of L_I. According to this design stance, the user may choose to utilize the formal linguistic constructions as a part of his or hers own logic system freely, adding interpretation axioms when convenient.

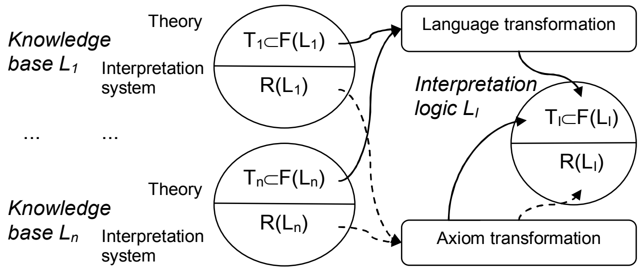


Figure 1. An outline process for working with interpretation logics

The basic setting of interpretation logics discussed above is illustrated in Figure 1. The role of the meta language(s) is not explicitly depicted since we may think that it is implicitly present in the whole setting. Further, the process of asserting the interpretation axioms is symmetrically depicted as a kind of transformation (represented by dotted lines in the figure), even if it typically appears asymmetrically in applications, with a somewhat less mechanical flavor.

Assuming the knowledge bases $\langle L_i \rangle$ are compatible with matching languages and interpretation systems, constructing the interpretation logic may be nearly trivial. For instance, assuming $\langle L_i \rangle = \langle L_1, L_2, L_3 \rangle$ where L_1 and L_2 are propositional logics and L_3 is a pure predicate logic, with the associated theories $T_1 = \{A, A \wedge B\}$, $T_2 = \{A \vee B\}$, $T_3 = \{\exists x: C(x) \rightarrow A\}$, we easily get a meaningful propositional logic L_I with a theory $T_I = \{A, A \wedge B, A \vee B, \exists x: C(x) \rightarrow A\}$ with some familiar interpretation theory and method of inference (which might allow simplifying the theory into some shorter, and "simpler", but semantically equivalent form, such as $\{A \wedge B, \exists x: C(x) \rightarrow A\}$).

When the language and the method of inference of the interpretation logic are sufficient for completely capturing the knowledge bases, we say that the interpretation logic is conformant. Completely conformant interpretation logics (as above), when applicable, are usually uninteresting in the sense that they simply aim introducing a maximally rich logic for capturing the knowledge bases. As expected, most useful interpretation logics are non-conformant, e.g., modifying the language and/or the interpretation system. This means that the designer establishing the interpretation actively selects and modifies the formulas for the language of the interpretation logic, and decides which axioms are included in the reasoning process.

An important special case of interpretation logics is provided by interpretation ontologies. Here the term ontology refers to a suitable ontology language, typically a restriction of some predicate logic. Interpretation ontologies are important simply because of the pivotal role of OWL (DL) in many concrete applications.

Note that the interpretation system of the interpretation logic may also be weaker than the interpretation system of a knowledge base, and still be able to meaningfully capture the key applications semantics. This is usually due to the fact that all theories do not fully exploit the properties of the underlying logics. However, the language transforming process may also map a wff in $F(L_i)$ into multiple wffs in $F(L_I)$, perhaps "writing out" some significant deductions. This is particularly the case when language transformations are allowed to process schematic wffs. In other words, the language transformation process effectively includes information about the intended reasoning process of the interpretation logic. (Of course, in most cases, e.g., writing out all deductions as ground wffs is impossible due to the large or potentially infinite number of them etc.)

Clearly, non-conformant interpretation logics actually change something in the language and/or in the interpretation of it. The (iterative) process of constructing useful non-faithful interpretation logics must be managed with care, requiring substantial understanding of the knowledge bases and the target application.

III. Example

Let us next consider a simple example that aims for combining three knowledge bases, K_1, K_2, and K_3 in terms of an interpretation theory in Prolog. To get a flavor of interpretation logics of useful applications, the wffs of initial knowledge bases include statements about individuals, ontology, and if-then rules using Semantic Web technologies. Using Semantic Web technologies simplifies the task of language transformations and the selection of the interpretation logics since the languages share the principal modeling characteristics.

A. Initial Knowledge Bases

Our target application involves capturing (research) interests of different people, based on data provided by registered users of a knowledge-sharing system. To keep our example straightforward, we consider only a very simple use case. We would like to provide a search mechanism for finding out people based on their interests (and vice versa), and to discover potential "weak associates", i.e., people working with similar topics, perhaps without knowing about each others' work.

Our first knowledge base K_1 includes two statements about individuals in RDFS (see, e.g., [14]) written in the Turtle syntax:

```
<http://www.foo.org/person_Paul>
 <http://xmlns.com/foaf/0.1/topic_interest>
<http://archive.astro.umd.edu/ont/Science.owl#Math>.
<http://www.foo.org/person_Susan>
 <http://xmlns.com/foaf/0.1/topic_interest>
<http://archive.astro.umd.edu/ont/Science.owl#Algebr
a>.
```

Strictly speaking, the first statement roughly says "there is something called (...`Paul`) that exists with a relationship called (...`topic_interest`) with something called (...`Math`)". A natural interpretation might be that Paul is interested in mathematics. The second statement is similar, with a natural interpretation that Susan is interested in algebra.

If we really would like to interpret the statements in RDFS (which is too expressive since RDFS-specific constructs are not actually used), there are more points to be made. In particular, according to RDF Semantics there is, e.g., no assumption that this is all there is nothing more to be known about the resources, nor that the names would actually refer to different things. Further, the interpretation system states that by default, each knowledge base of RDFS statements implicitly includes the so-called RDF and RDFS axiomatic triples. In practice, these include not only ground wffs such as `rdf:type rdfs:domain rdfs:Resource`., but also schematic wffs, for instance rdf:<<N>> rdfs:domain rdfs:Resource. where <<N>> stands for an arbitrary identifier of a membership property, e.g. `_1`.

Our second knowledge base K_2 (that is an excerpt of a real ontology by E. Shaya (see http://archive.astro.umd.edu/ont/index.html)) includes statements in OWL DL, written in the RDF/XML syntax (see, e.g., [22]):

```
<rdf:RDF xmlns:rdf ="http://www.w3.org/1999/02/22-
rdf-syntax-ns#" xmlns:rdfs="http://www.w3.org/2000/
01/rdf-schema#" xmlns:owl="http://www.w3.org/2002/
07/owl#"><owl:Ontology rdf:about="http://archive.
astro.umd.edu/ont/Science.owl#"/>
 <owl:Class rdf:ID="Albegra">
   <rdfs:subClassOf rdf:resource="#Math"/>
 </owl:Class>
<owl:Class rdf:ID="Math"/>
</rdf:RDF>
```

Intuitively, these statements assert that the given ontology includes a class Algebra that is a subclass of the class Math. Again, strictly speaking, we have not asserted that the two classes are actually different things (but in principle OWL would allow us to do so), a set of axiomatic triples is implicitly assumed, etc. Further, the knowledge base K_2 accounts as an OWL Lite ontology since it only exploits a restricted set of the OWL language. (Actually declaring that the classes are disjoint would have resulted an OWL DL ontology.)

Finally, our third knowledge base K_3 includes a rule in SWRL, written in the "human readable" SWRL syntax (see [15]):

```
hasInterest(?p1, ?s) ∧ hasInterest(?p2, ?s) →
hasWeakAssociate(?p1,?p2)
```

Intuitively, this rule asserts that whenever two persons share a common interest, they become weak associates. Again, strictly speaking, the rule simply asserts that whenever the conditions specified in the antecedent hold, then the conditions specified in the consequent must also hold, a set of rule axioms is assumed, etc.

B. Issues of interpretation

We would now like to establish an interpretation theory K_I in Prolog for purpose of an application that wishes to integrate the knowledge bases K_1, K_2, and K_3 which are theories in RDF Schema, OWL DL, and SWRL logics. Intuitively, we would like to use the knowledge bases for reasoning about the

interests of Paul and Susan, perhaps to find out whether the interests of Paul and Susan overlap.

However, certain known issues exist that make simply aggregating the knowledge bases problematic. For instance:

1. The statements in K_1 relate Paul and Susan with resources that are classes according to K_2. Relating individuals with classes is not allowed for an object property in OWL DL.
2. The names in the rule of K_3 do not match with the names found in K_1 and K_2.
3. The knowledge bases include lots of implicit axioms, but assert, e.g., no disjointness relations.

In a concrete application, we might have a priori knowledge about these things that we would like exploit in the interpretation. In particular, for purposes of our particular application, we would perhaps like to give up the open world assumption and drop several unnecessary wffs, to make reasoning more straightforward, discarding uninteresting deductions along the way.

Indeed, while with certain assumptions, we might be able to capture the knowledge bases K_1, K_2, and K_3 with sufficiently rich conformant theory and interpretation logic (something like "OWL Full + SWRL" logic), we would not usually wish to do so. At least we would like to glue the useful names together (e.g. the properties `topic_interest` and `hasInterest` are equivalent), add disjointness axioms (Paul and Susan are names of different things), and ensure the tractability of our system, thus effectively ending up with a new theory.

Note that we could in principle solve these problems by redesigning the knowledge bases and harmonizing the modeling. However, in practice, the knowledge bases might not be under our control, or they might include other, more important use cases.

With these issues in mind, we will next compile our interpretation theory K_I in terms of a Prolog program (some other logic system might applicable as well). After deciding basic modeling strategy, the work involves making a series of language transformations and asserting appropriate interpretation axioms.

C. An interpretation logic in Prolog

Let us first decide encoding for the knowledge base K_I. A general and a well-known approach is encoding RDF statements as triples (we omit the URL prefixes for brevity; in a concrete application the name `'Paul'` would be represented as `'http://www.foo.org/person_Paul'`, some literals might be associated with datatypes, the knowledge base might include anonymous terms [nodes], etc.):

```
t('Paul','hasInterest','Math').
t('Susan','hasInterest','Algebra').
```

Note that we changed the name of the predicate, due meta-reasoning and selection (`topic_interest` and `hasInterest` are equivalent but we are only interested in deductions involving the name `hasInterest`).

Then, let us assert the terminological information related to the second knowledge base K_2:

```
t('Algebra','subClassOf','Math').
```

Note that we, e.g., discarded the information about the existence of an ontology and that Algebra and Math are classes since we have no use for this piece of information in our application. Further, we have not included any RDF(S) axiomatic triples so far because our application does not have any use for them.

Finally, we transform the third knowledge base K_3 into a Prolog rule:

```
t(Vp1,'hasWeakAssociate',Vp2) :-
t(Vp1,'hasInterest',Vs),t(Vp2,'hasInterest',Vs).
```

In this case, we have again discarded axioms of the SWRL logic from our interpretation theory.

Now it is time to assert some interpretation axioms of our own. The objective is of course a definition of a useful system that captures the aspects of interest from the initial knowledge bases, for purposes of the target application (a real application would obviously need a more elaborate description in the meta language).

Let us first assert a procedure that essentially states that the `subClassOf` predicate is transitional. We can achieve this e.g. by introducing a support procedure `transitiveSubClassOf`:

```
t(A,'transitiveSubClassOf',B) :-
t(A,'subClassOf',B).
t(A,'transitiveSubClassOf',B) :-
t(S,'subClassOf',B),t(A,'transitiveSubClassOf',S).
```

In general, formulations of transitive properties typically include potential pitfalls in logic programming, resulting to endless queries. Of course, writing support procedures in triples format is not strictly necessary.

We shall next assert that being interested in something particular implies interest in the topic in general. For this, we exploit the transitive definition of the subclass property:

```
t(Vp,'hasInterest',Vsuper) :-
t(Vsub,'transitiveSubClassOf',Vsuper),t(Vp,'hasInter
est',Vsub).
```

This concludes the definition of our interpretation theory K_I.

We may now use out interpretation logic system for working with the interpreted knowledge. In the case of a Prolog interpreter, the basic tool is making queries. For instance, we might be interested in who is interested in what. The `query ?- t(X,'hasInterest',Y).` now returns:

```
X = 'Paul' Y = 'Math';  X = 'Susan' Y = 'Algebra'; X
= 'Susan' Y = 'Math'; No
```

Of course, we can ask more explicit questions such as "What is Paul interested in?"

We may also ask that who have weak associations based on mutual interests. The query `?- t(X, 'hasWeakAssociate', Y).` seemingly does the trick. It returns:

```
X = 'Paul'  Y = 'Paul'; X = 'Paul'  Y = 'Susan'; X =
'Susan' Y = 'Susan'; X = 'Susan' Y = 'Paul'; X =
'Susan' Y = 'Susan'; No
```

In other words, we get all the conceivable combinations of persons that are interested in same things, either based on a direct fact, or transitive rule based on the available taxonomic knowledge of the topics of interest.

A nice thing about our application is that it includes only the reasoning potential more or less explicitly stated by the interpretation theory K_I. and modifying the interpretation is easy. For instance, if we would like to further refine our interpretation by discarding symmetric properties, we could simply rewrite `hasWeakAssociate` as follows:

```
t(Vp1,'hasWeakAssociate',Vp2) :-
t(Vp1,'hasInterest',Vs),t(Vp2,'hasInterest',Vs),Vp1\
=Vp2.
```

Finally, from applications' perspective, it is sometimes beneficial to map the deduced results further onto some other interpretation logics. For instance, should the deductions outlined above be integrated with Semantic Web applications, we might assert the Prolog query results in terms of a suitable RDF vocabulary. When the queries include only concatenated triple terms $\wedge_j$ $t_i(p_{j1}, p_{j2}, p_{j3})$ as in `?- t_1(Vp_11,Vp_12,Vp_13), t_2(Vp_21, Vp_22,Vp_23), ..., t_n(Vp_n1,Vp_n2,Vp_n3).` we may trivially write out the retrieved variable configurations in some RDF format. For instance, a result configuration `X = 'Paul' Y = 'Susan'` of the query `?- t(X, 'hasWeakAssociate', Y).` yields `Paul hasWeakAssociate Susan.` (with the exception that applications would of course use global URI names). Transforming this into, e.g., RDF/XML, is straightforward.

D. From convenience methods to knowledge interfaces

Interpretation logics do not need to follow the linguistic or interpretative conventions of the initial knowledge bases. For instance, we may assert convenience methods as support procedures, e.g., classifying names in order to be able to easily list all the known persons and their properties with a query `?- t(X,'type','Person').` (Note that in this case, asserting the required wffs can be achieved either with the language transformations or with the interpretation axioms.)

Various kinds of convenience methods may be used to encapsulate the internal complexity of the initial knowledge bases. This design perspective allows using interpretation theories as knowledge interfaces, enabling implementing knowledge views.

We may also exploit the language components of the initial knowledge bases in a setting of non-standard logics, e.g., probabilistic and fuzzy logics. It is important to realize that this not only enables integrating implied wffs according to non-conformant crisp reasoning, but also induced wffs, due to statistical or imprecise reasoning. Effectively, this allows integrating logical theories with predictions (e.g. via manually designed probabilistic models or machine learning algorithms) and wffs due fuzzy reasoning (e.g. classifications and deductions associated with fuzzy models etc.).

For instance, a natural application for fuzzy logic exists in our example: simply saying that because Susan is interested in Algebra she is also interested in Math is too vague. It seems natural to think that weak associations between people, based on their interests, may have varying degrees, according to the closeness of the topics of interest. In particular, assuming a tree-like taxonomy of interest, a person that asserts being interested in the top concept (now Math) would participate in every weak research group according to R_I!

This observation leads into a notion for computing degrees for weak associations, e.g., based on the (shortest) distance of topics in a topic graph (a taxonomy tree in our example). Of course, the intuition of the degree of weak association may well be captured with different kinds of definitions of fuzzy membership.

Coming up with extensions like this does not necessary mean giving up Prolog etc. For instance, one may use the language of Prolog to capture wffs associated with fuzzy models, and implement reasoning procedures of fuzzy logic [17]. However, the interpretation system might need re-evaluating, giving explanations of the newly computed results in the meta language. In fact, this interplay of interpretation system and meta language is already present in our simple example, e.g., when considering whether Susan and Paul are different individuals and how this should be interpreted in the application.

IV. Conclusion

In this article, we have presented a generic framework called interpretation logics for integrating a series of available, mismatching knowledge bases and explicitly designing the related interpretation system. We conclude the article with few notes about implementation and applications.

When working with RDF/XML data and Prolog, relatively natural implementation architecture exists. The knowledge bases are first identified as data sources and mapped onto RDF/XML, e.g. with the help of GRDDL techniques. This raw data is then queried and filtered into an appropriate intermediate RDF/XML form using query and transform processors, effectively performing the language transformation step (with the help of, e.g. SPARQL, XQuery 1.0 and XSLT 2.0). The transformed RDF/XML data is serialized into a Prolog program (as triples), and associated with the interpretation axioms, e.g. again in terms of XSLT processing. The Prolog interpreter is then consulted for interesting queries. Finally, the results are applied as such, or transformed back into the RDF/XML format, perhaps to be integrated with the original knowledge bases or data sources. (About the referenced acronyms and technologies, see, e.g., http://www.w3.org/TR/.) In applications, practical challenges include user interface design, Unicode tool support, modeling issues and ensuring the speed of queries, error management and understandable reporting, and applications-specific issues such as the logical interpretation of inferred data in applications.

To support these activities, we have implemented a system called *tinker* that is capable of reading RDF/XML data sources,

carrying out the reasoning subtask with asserted axioms, and outputting conclusions in RDF/XML, as described above. Considering our OPAALS visualization use case, tinker can be fitted into the visualization pipeline architecture (see [18]), as a part of a complete series of pre-processing, query, reasoning, and transforming components. In practice, this also requires implementing an additional wrapper layer on top of the tasks outlined above.

Finally, while the framework of interpretation logics does not solve the issues of, e.g., transparency, expressiveness, and non-polynomial complexity in logical modeling, it should help clarifying the reasoning component in applications. In particular, the introduction of interpretation logics makes the abstract dialogue of theories in different logics in applications fairly concrete. Consider, e.g., the question (*): What is the relationship between and the consequence of two theories T_1 and T_2 (e.g. an SVBR rule system and an OWL DL ontology)? The answer ($T_1 = T_2$, $T_1 \subset T_2$, $T_1 \supset T_2$, or some partial inclusion, resulting inconsistency, etc.) depends upon the properties of the theories and the underlying logics and the interpretation on the level of the meta language(s). However, in applications, the question (*) typically appears in a form: What is the relationship of two theories T_1 and T_2 *with respect to a particular application*? Since few applications need every aspect of the theories, a useful amalgam might be constructed in terms of a suitable interpretation logic system (e.g. an easily accessible Prolog interpreter), by explicitly escaping the pathologies of the theories, without the need of introducing highly specialized reasoning tools in every application.

REFERENCES

[1] M.E. Aranguren, Ontology design patterns for the formalisation of biological ontologies, Ph.D. Thesis, University of Manchester, faculty of engineering and physical sciences, 2005. Available at http://www.gong.manchester.ac.uk/ docs/MPhilThesis.pdf

[2] F. Baader, D. Calvanese, D.L. McGuiness, D. Nardi, P.F., and Patel-Schneider, Description Logic Handbook, Cambridge University Press, 2002.

[3] D. Beckett, Dave Beckett's Resource Description Framework (RDF) Resource Guide, Planet RDF, 2007. Available at http://planetrdf.com/guide/

[4] C.F. Da Silva, L. M´edini, S.A. Ghafour, P. Hoffmann, P., and P. Ghodous, Semantic Interoperability of Heterogeneous Semantic Resources, In Electronic Notes in Theoretical Computer Science 150, 2006, pp. 71-85.

[5] E. Dantsin, T. Eiter, G. Gottlob, A., and Voronkov, Complexity and Expressive Power of Logic Programming, ACM Computing Surveys, Vol. 33, No.3, September 2001, pp. 374-425.

[6] E. Gamma, R. Helm, R. Johnson. J., and Vlissides, Design patterns: elements of reusable object-oriented software, Addison-Wesley, USA, 1994.

[7] S. Hawke (Ed.), Rule Interchange Format Working Group Charter, World Wide Web Consortium, 2007. Available at http://www.w3.org/2005/rules/wg/charter.html.

[8] J. Heflin (Ed.), OWL Web Ontology Language Use Cases and Requirements, W3C Recommendation, 10 February 2004, World Wide Web Consortium, Available at http://www.w3.org/TR/webont-req/.

[9] I. Herman (Ed.), W3C Semantic Web Activity, World Wide Web Consortium, 2007. Available at http://www.w3.org/2005/rules/wg/ charter.html.

[10] J. Hintikka, What Do Logic and Computer Science to Do with Each Other? In Logic, Mathematics and the Computer, Publications of the Finnish Artificial Intelligence Society, No. 14, 1996, pp. 85-88.

[11] M. Horridge, H. Knublauch, A. Rector, R. Stevens, and C. Wroel, A Practical Guide To Building OWL Ontologies Using The Prot´eg´e-OWL Plugin and CO-ODE Tools (Edition 1.0), The University Of Manchester, 2004. Available at http://www.co-ode.org/resources/ tutorials/ProtegeOWLTutorial.pdf.

[12] E. Hyvönen, M. Junnila, S. Kettula, E. Mäkelä, S. Saarela, M. Salminen, A. Syreeni, A. Valo, and K. Viljanen, Finnish Museums on the Semantic Web: The user's Perspective on MuseumFinland. Proceedings of the Museums and the Web 2004, Toronto, Canada. Available at http://www.archimuse.com/mw2004/papers/hyvonen/hyvonen.html

[13] H. Knublauch, O. Dameron, and M.A. Musen, Weaving the Biomedical Semantic Web with the Protégé OWL Plugin, Proceedings of the First International Workshop on Formal Biomedical Knowledge Representation (KRMED04), Whistler, Canada, 2004. Available at http://protege.stanford.edu/plugins/owl/publications/KRMed2004-protege-owl.pdf.

[14] F. Manola, E. Miller, and B. McBride (Eds.), RDF Primer, W3C Recommendation, 10 February 2004, World Wide Web Consortium, Available at http://www.w3.org/TR/2004/REC-rdf-primer-20040210/.

[15] G. Newton, J. Pollock, D.L. McGuinness, Submission request to the World Wide Web Consortium: Semantic Web Rule Language (SWRL), World Wide Web Consortium, 2004. Available at http://www.w3.org/ Submission/2004/03/.

[16] N. Noy (Ed.), Representing Classes As Property Values on the Semantic Web. W3C Working Group Note 5 April 2005, World Wide Web Consortium, Available at http://www.w3.org/TR/swbp-classes-as-values/.

[17] O. Nykänen, An Approach to Logic Programming with Type-1 Fuzzy Models Using Prolog. Proceedings of the IADIS International Conference of Applied Computing, San Sebastian, Spain, 2006, pp. 201-208.

[18] O. Nykänen, M. Mannio, J. Huhtamäki, and J. Salonen, A Socio-Technical Framework for Visualising an Open Knowledge Space, In Proceedings of the WWW/Internet 2007, Portugal, to appear.

[19] OMG. Semantics of Business Vocabulary and Business Rules (SBVR), Second SBVR Interim Specification without change bars dtc/06-08-05. Object Management Group, 2006. Available at http://www.omg.org/ docs/dtc/06-08-05.pdf

[20] D. Reynolds, Jena 2 Inference support. Sourceforge documents by the Hewlett-Packard Development Company, 2007. Available at http://jena. sourceforge.net/inference/index.html.

[21] S. Russel, and P. Norvig, Artificial Intelligence; A Modern Approach, Prentice Hall, USA, 1995.

[22] M.K. Smith, C. Welty, and D.L. McGuinness (Eds.), OWL Web Ontology Language Guide, W3C Recommendation 10 February 2004, World Wide Web Consortium, Available at http://www.w3.org/TR/owl-guide/.

[23] R. Troncy, J. van Ossenbruggen, J.Z. Pan, J.Z., and G. Stamou (Eds.), Image Annotation on the Semantic Web, W3C Incubator Group Report 14 August 2007, World Wide Web Consortium, Available at http:// www.w3.org/2005/Incubator/mmsem/XGR-image-annotation/.

[24] V. Vassiliadis, Thea: A Web Ontology Language - OWL Library for [SWI] Prolog, SemanticWeb.gr, Greece, 2007. Available at http://www.semanticweb.gr/TheaOWLLib/Thea%20OWL%20Lib.pdf

[25] J. Wielemaker, G. Schereiber, and B. Wielinga, Prolog-based infrastructure for RDF: performance and scalability. In Proceedings ISWC'03, Florida. USA, 2003.

[26] J. Wielemaker, An optimised Semantic Web query language implementation in Prolog. Proceedings of the ICLP 2005, Barcelona, Spain.

[27] J. Wielemaker, G. Schreiber, and B. Wielinga, Using triples for implementation: the Triple20 ontology-manipulation tool. Proceedings of the ISWC2005, Galway, Ireland.

Mediating the Dialectic Relations between Indigenous Knowledge and Identity: Lessons from DEAL Project

[OPAALS Conference 2007]

Debashis Pattanaik
OPAALS Research Laboratory
Indian Institute of Technology Kanpur,
Kanpur, India
debas.p@gmail.com

Runa Sarkar
Department of IME
Indian Institute of Technology Kanpur,
Kanpur, India
runa@iitk.ac.in

Jayanta Chatterjee
Department of IME
Indian Institute of Technology Kanpur,
Kanpur, India
jayanta@iitk.ac.in

Abstract*—The world today is confronted with large scale issues around indigenous knowledge. The growing interest in the indigenous knowledge is primarily due to the uncertainties surrounding everyday life which has challenged unbridled belief in scientific knowledge and rationality. The thirst to assimilate indigenous knowledge to formal knowledge has resulted to a situation where identity and culture of the indigenous are threatened. Under these conditions the ICT has a greater role in both accommodating these knowledge into formal knowledge system as well as in maintain the culture and identity of the indigenous. Based DEAL observations we argue that an ICT facilitated architecture such as Kisan Blog has potential socio-technical features to accommodate such knowledge in to formal knowledge by maintaining the identity and culture of the indigenous. A technology which best support the normative structure is likely to bring more acceptability and benefits to the community who uses it.

Keywords-indigeneous knowledge, audio blog, knowledge, identity, ICT, DEAL

I. Introduction

The world today is confronted with large scale issues around indigenous knowledge. The unprecedented quest for the production of wealth has given rise to profound uncertainties in all spheres of life (Beck 1996). The continuing production of uncertainties has challenged unbridled belief in scientific knowledge and rationality. Indigenous knowledge has gained attention as an alternative form of approach for reorienting the knowledge system. From being marginalised in the dominant development paradigm, the indigenous are now at the centrestage of debates that surround their knowledge systems. This has affected the indigenous peoples. While at one hand it has given them recognition in the form of an acceptance of the indigenous knowledge; on the other hand, this has raised questions about their cultural survival or identity. This assumes great importance, as knowledge of particular group has become the tool for cultural identification in today's world (Pattanaik and Jha 2007). Historically indigenous knowledge based on an indigenous worldwide has been under the control of indigenous peoples. Traditionally these knowledge are passed down form generation to generation through oral histories, story telling, picture writings songs and ceremonies (Barnes and Danard 2007).

II. On the Concept of Indigenous

Around the world, indigenous people today face difficulties in gaining adequate recognition. Furthermore, indigenousness as a concept fails to attain the status it deserves (Miller 2003). Maybury-Lewis (2002) observes that there is no clear cut distinction between indigenous people? "The term is ambiguous because a majority of the peoples are "indigenous" in the sense of having been born [there] and descended from those who were born [there.]" Most of the definitions describe indigenous peoples as those social groups that are culturally identified and maintain historical continuity with their ancestry. This continuity is in the forms of their organization, culture, self-identification, and languages (Dei 2000a, 200b; Semalio & Kincheloe 1999). There are other definition which describes them as descendants of those who have been marginalized by major powers and dominant groups, the Basarwa of Botswana and the Maasai of Kenya and Tanzania represents such indigenous people (Mammo 1999; Maybury Lewis 2002; McGovern 1999; Wangoola 2000). However in general the term refers to those group of people who have maintained a lifestyle that sets them apart from the rest of society, and have

traditionally been subordinated and marginalized by unequal economic, political and social structures (Wangoola 2000). In this paper we use the term indigenous more congruent with the definition given by United Nations (UN) and form an ecological perspective. The term in the UN context essentially refers to 'those who maintain a collective identity through association with specific territories' (McIntosh 2000). According to UN:

> Indigenous communities, peoples and nations are those which having a historical continuity with pre-invasion and pre-colonial societies that developed on their territories, consider themselves distinct from other sectors of the societies now prevailing in those territories or parts of them. They form at present non-dominant sectors of society and are determined to preserve, develop and transmit to future generations their ancestral territories, and their ethnic identity, as the basis of their continued existence as peoples, in accordance with their own cultural patterns, social institutions and legal system (UN 1996).

An ecological perspective views the indigenous people as 'the original stewards of the environment, holding the land of their ancestors in trust for future generations' (McIntosh 2000). In the Indian context whereas the Minas of Rajasthan, who are also dominant in their area, can be referred to as a tribe, the Todas of Tamilnadu, or the Santals and Kondhs of Orissa can be described as indigenous people. "Some of the indigenous people might be more 'advanced' than others, but what makes them distinct is their closeness to nature and comparative geographical isolation" (Pattanaik and Jha 2007).

III. ON THE CONCEPT OF INDIGENEOUS KNOWLEDGE

Like the term indigenous peoples, indigenous knowledge (IK) also has multiple descriptors. A majority of them refer IK as the unique, local knowledge existing within and developed around the specific conditions of local people resulting from their long-term geographical residence. This knowledge is part of their cultural heritage and histories. It portrays their way of life and serves social interests. Such knowledge systems are cumulative and represent generations of experiences (Kinuthia 2007).

In general usages, two terminologies are employed to characterise local knowledge: traditional and indigenous. They are often used interchangeably; however, there is a difference between the two. Though there is no agreed definition of traditional knowledge, it may be seen as a subset of indigenous knowledge (Pattanaik and Jha 2007).

> Traditional knowledge is generally expressed in the communities and encompasses expressions of folklore, religion (e.g. sacred places, plants, animals), crafts (e.g. developments of technologies for producing textiles, food), agriculture (e.g. management of ecosystems, development of plants and animals with specific properties), and medicines (e.g. herbal products**).** However indigenous knowledge is no different from traditional knowledge except that the holders are indigenous people rather than non-indigenous communities embodying traditional life styles (WZB: nd).

Gernier defines indigenous knowledge as the unique, traditional and local knowledge existing within and developed around specific conditions of people indigenous to a particular geographic area. Indigenous knowledge innovates from within and internalises. By and large it is context specific and uses and adapts external knowledge to suit the local situational needs (Nuffic 2000).

The characteristics of indigenous knowledge may be summarised in the following manner. It is generated within communities. It is location and culture specific. It is the basis for decision-making and survival strategies. It covers the critical issues of primary production, human and animal life, and natural resource management. It is dynamic and based on innovation, adaptation, and experimentation. It is oral and rural in nature. It is not systematically documented (Nuffic 2000). Further, the other qualities that separate indigenous from other forms of knowledge is their custodial rights, i.e., equity in sharing, collective ownership and rules governing secrecy and sacredness (Puri 2000). In general, indigenous knowledge refers to the cultural and intellectual heritage of the indigenous people (McIntosh 2001).

Recognition of the embedded nature of culture to the situational, historical and political context supports the argument that both the term "indigenous" and "indigenous knowledge' are not a detached concepts. It can then be inferred that the ability to use these knowledge forms is pertinent to peoples' understanding of themselves and their world, and have an influence on their identity and everyday practices (Dei 2000a; 2000b; Robyn 2002; Semali 1999). The history of colonization and Western world view of what constitute knowledge during the recent past has led to instances where indigenous populations were excluded and deprived of their right to this knowledge (Wangoola 2000). This has led to many ethic based identity struggle during recent past.

IV. KNOWLEDGE: COMPETING WORLD VIEWS

Broadly there are two paradigms of knowledge: one that can be referred as Northern techno-centric; and the other that

exists within local or indigenous communities. Within a Northern techno-centric paradigm, knowledge is seen as a commodity, which can be valued and traded. The concept of commodity is primarily based on scarcity that emphasises individual ownership and long-term planning. Contrary to it, within indigenous communities, knowledge is seen as integral part of existence. Indigenous knowledge is based on theories of abundance, subscribing to the idea that nature has capabilities to fulfil all human needs. Further, in these societies, property including knowledge is held communally under a custodianship (Manek and Lettington 2001). The Northern Techno-centric orientation to individual ownership and long-term planning in the domain knowledge are considered meaningless (Pattanaik and Jha 2007). However modernity in its quest for rationalisation has resulted in various forms of crises. The uncertainties in all spheres of life have demonstrated the limitations of formal scientific knowledge. This has led to an unbridled quest for alternative forms of knowledge. According to Todd:

> Western culture seems to want everything, to go everywhere. Wants that seem endless….The desire to know, seek new experience, take new journeys, create light, has somehow grown from a flame to a forest fire that burns everything in its way …..Of course, in a world with a legacy of colonialism, the hunger of western culture is threatening and frightening. We have had to feed that hunger, with the furs of animals and flesh of fish and the gold and silver of our lands and ourselves as fearsome mysteries in the West's drama of itself (Todd 1996).

Most of this knowledge today is taken out of their cultural context. There are large scale appropriation of indigenous knowledge and symbols for the purpose of wealth accumulation through various means ranging from commercial art markets to internet. Indigenous knowledge and symbols in this new world order become commodities. This control of indigenous knowledge and symbols at the one end limits the expression and perception of indigenous worldviews, while at the other it threatens their cultural survival. In addition, these controlled representations deprive the indigenous from their right to their knowledge system and symbols (Barnes and Danard 2007).

V. RELEVANCE OF INDIGENOUS KNOWLEDGE

The World Conference on Science (Budapest 1999) recommended that scientific and traditional knowledge should be integrated in interdisciplinary projects dealing with links between culture, environment and development particularly in the areas of the conservation of biological diversity, management of natural resources, understanding of natural hazards and mitigation of their impact. Similarly the UN Conference on Environment and Development has also emphasised on the practice and knowledge of indigenous people for better environmental management to achieve sustainable development (UN 1992). There is a growing consensus around the world that indigenous knowledge system can be utilised as an alternative method for sustainable regional development (Cox 2000).

There are two views regarding the use of indigenous knowledge. While the first paradigm accepts the supremacy of Western scientific notion of knowledge the second rejects hierarchical division of knowledge (Sharma 2001). Within the first paradigm all forms of knowledge including indigenous knowledge which does not follow the assumptions of formal scientific notion are considered as irrational and 'deserves to be lost' within the civilizational mission of the modernity (O'Neil 2000). The second view on the contrary accepts possible alternatives and recognises indigenous knowledge as one of the possible alternatives. It lays emphasis on equity of knowledge, that is, an integration of indigenous knowledge with formal science (Sharma 2001). According to this view, equity of knowledge not only serves the purpose of conservation of biological diversity and preservation of traditional knowledge but it provides empowerment, security and opportunity to the indigenous people.

The World Bank has pointed out that in the emerging global knowledge economy, a country's ability to build and mobilise knowledge capital is equally essential for sustainable development as the availability of physical and financial capital (World Bank 1997). The basic component of any country's knowledge system, more so in the case of developing countries, is its indigenous knowledge. It encompasses the skills, experiences and insights of people, applied to maintain or improve their livelihood. It is also the social capital of the poor, their main asset in the struggle for survival, to produce food, to provide for shelter or to control of their lives (Nuffic 2000). It would perhaps then be correct to say that adopting indigenous knowledge will provide security to not only to the formal scientific knowledge but also to the nation-state as a whole. By capitalising on the collective wisdom of formal and traditional sciences, we shall be able to help people address the problem of declining common property and to manage the risk they face in everyday life because of the destruction of the resource base (Pandey 2000).

Localized relevance of IK, therefore, has significant bearing on the organization of knowledge and ICT can play a crucial role in its incorporation and preservation (Dei 2001b; McGovern 1999; Sillitoe 2002). However three broad aspect of the indigenous knowledge potentially complicate their integration: (a) the uniqueness of IK in a particular culture does not imply consensus with other knowledge structures; and (b) in this highly transient "global" lifestyle, it is not uncommon for IK to be influenced by non-indigenous knowledge; (c) above all use of IK as commodity threatens their cultural survival/identity.

VI. CONFLICTS AT THE FRONTIER OF CULTURAL IDENTITY

In a discussion on culture and identity, Laroui has defined cultural identity: "Sometimes it is race, sometimes language, sometimes religion, sometimes nationality, sometimes it is culture in the anthropological sense sometimes it can even be dress. Cultural identity then is what differentiates" (Laroui 1998). As indigenous knowledge is different from formal knowledge system, it sets the indigenous people apart and gives them a separate identity. (Pattanaik & Jha 2007).

The process of modernisation is dialectical in nature. It both universalises inside and outside the society as well as differentiates socially and psychologically at all levels of everyday life. In this course of differentiation cultural identity has become an exiting fact for everyday life. The individuals as well as groups become conscious of their cultural identity. It manifests itself profoundly where there is competition, or any form of dominance (Pattanaik & Jha 2007). Cultures sought to be defended are defined, analysed and neatly categorised to fit the situation or agenda (Laroui 1998). By and large indigenous communities around the world are economically poor and socially disadvantaged. They continue to exist in what may be described as colonial situations. Under these circumstances they are most likely to assert their identity. One of the ways through which they can do it is by conforming to their knowledge system (Pattanaik and Jha 2007).

Generally this knowledge is tacit in nature. Traditionally such knowledge systems have been the property of the community. Most of these communities live in geographically isolated areas. This gives them a form of political and cultural autonomy where community gatherings and participation remain important. It is important to note that knowledge systems within these communities are considered as their sacred ancient past, it is a part of their survival mechanism and also constitute an important element of what is regarded as their own self. Traditionally, the elders of the community play the role of the guardian of this knowledge. Only certain community members adhering to the principles of the community have access to it in this way it is preserved and passed down from generation to generation (Chapin 1991).

In the recent past the growing concern over indigenous knowledge system has made them conscious of protecting their knowledge systems. There are movements all around the world to protect their symbols and knowledge system. Many of these movements are a reaction to the process of modernisation. Examples of organisations that have led some of these movements are the Australian Aboriginal Progressive Association and the Association of the Indigenous people of the North (Bannister and Barrett 2001).

The reaction to the process of modernisation is due to two main reasons. Firstly, in many cases this knowledge is utilised without giving any credit to the indigenous people. In addition some of this knowledge that is acquired by the outsider is subsequently brought into modern world and traded as commodities (Bannister and Barrett 2001). Hence the sacred character of this knowledge that the indigenous people ascribe to it is threatened or lost. Secondly, modernity in its own virtue creates spaces for one to assert oneself or one's community. Thus with the penetration of modernisation the indigenous people today become more conscious about their identity. Since the knowledge system is a part of their identity they tend to guard it from encroachment. These factors have given rise to a situation where indigenous people pull back from sharing the knowledge with the outside world. As on out come of the overall process the integrative force that modernisation seeks to promote is checked (Pattanaik & Jha 2007).

VII. A FEW OBSERVATIONS FROM PAST EXPERIENCES

Evidence from all around the globe indicates that indigenous peoples are increasingly uncomfortable sharing cultural knowledge outside their communities due to uncertain negative consequences (Chapin 1991). This is both profound in physical arena as well as in ICT facilitated interfaces.

The experience of Kuna gives an insight to the difficulties that are experienced at the physical interface. In the early 1980s an internationally funded project was undertaken among the Kunas of Panama for the management of the forest area of the Kuna territory. One of the basic aims of the project was to integrate indigenous knowledge with formal scientific knowledge. Traditionally the elders of the Kuna are the guardians and custodians of the Kuna knowledge system. The elders neither showed an interest in the project nor did they share their knowledge with the project. The reasons for this were two-fold. "First, for the Kuna, the knowledge was the 'Way of the Great Father', which is sacred, and this 'sacred lore is a soul of their identity'. They did not want to part with this. Secondly, the invasion of modernisation has fragmented the Kunas: the difference between the young and the old had already become wide." There were disagreements over the sacred character of the tradition among the young and the old. The elders accused the young of not being 'genuine'. The young people consider these as curious folklore whereas for the elders these are living documents about their world in which they live. As a result, the older generations separated themselves from the rest and kept the knowledge system confined to themselves (Chapin 1991). Where tradition is strong, people see no need to preserve esoteric knowledge. They simply live their culture. But the process of modernisation has challenged and even eroded the traditions among the Kunas (Pattanaik & Jha 2007).

The observations at the ICT facilitated interfaces show more of similar picture. ICT broadly refers to computers, software, networks and related systems that allow users to access, analyze, create, exchange and use data, information and knowledge. If successfully implemented and maintained, the infrastructure brings together people in different places and time zones with the multimedia tools for data, information and knowledge management (Herselman & Britton 2002).

There has been effort to capture IK through the use of ICT tool; however the use of ICT has shown its own limitations. Firstly it does not provide the opportunity to know the ways indigenous knowledge continues to live in the lives of the present generations and those of the future. Secondly, many websites which has been developed to promote indigenous knowledge and symbols ignore the importance of the indigenous people's traditional roles in taking care of these knowledge and being responsible for protecting and transmitting their knowledge and history (Barnes and Danard 2007).

Indigenous peoples have always seen themselves as connected to all of life and all of relations. The contemporary use of knowledge and symbols of the indigenous in the cyberspace today fabricates this connection. The artificial structure of cyberspace reflects primarily Western thought, which is fragmented and disconnected, and strives for dominance over humanity to find meaning (Todd, 1996). "The Web" fundamentally embodies this alienation of Western thought (Barnes and Danard 2007). Information presented on the internet is distanced from its context. Perhaps this lack of context makes it appear neutral. This supposed neutrality possibly makes the information more acceptable. But for indigenous peoples, who are represented and defined by non-indigenous peoples on the internet, the information is not neutral or acceptable when it is filtered through perspectives which promote bias and reinforce stereotypes (Barnes and Danard 2007).

Colonization and its ongoing practice in society and through information technology, produces distance between the individual and their environment, between those doing the representing (mainstream) and those being represented (indigenous people), between indigenous peoples and their communities, between indigenous peoples and their knowledge and art, and between indigenous peoples and their sense of self in all aspects including physical, emotional, mental and spiritual. The outcome is that the history, culture, identity and knowledge of indigenous peoples are being eroded, erased and reconstructed. The genuine loss of indigenous knowledge and diversity of thought is perhaps shifting the world towards a singular hegemonic structure. The value of life is being replaced by the value of commodities and resource accumulation for the dominant society.

Misapplied ICT tools can led to repercussive consequences; when it is introduced to indigenous groups it brings along mass media, popular culture and global languages that can potentially conflict with local traditions. Yet paradoxically, the same technologies also provide users with new tools that can be used to preserve, promote and strengthen their culture and identity (Lieberman 2003). The challenge with ICT is to make aware the users about the potential benefits and alternatives available in ICT facilitated interface (McGovern 1999; Sillitoe 1999). Successful participation is impractical when potential users do not know the alternatives and benefits (Herselman and Britton 2004).

VIII. SUGGESTION FOR INDIGENOUS KNOWLEDGE INTEGRATION

The question then is how best to access and integrate these knowledge systems for the betterment of humankind? A developmentalist approach emphasises a 'melting pot' framework. In this approach different group are assumed to submerge under a greater collective force. Smaller groups are expected to sacrifice their interests in the larger interest of the society. Development alone becomes the key issue (Pattanaik and Jha 2007). In contrast, a rights-based model emphasises progress rather than unprecedented development. This model focuses on development along with the rights of the groups involved in it intact. A right-based approach emphasises the creation of a public space. Policies are made on the basis of mutual agreement of all the actors involved in it. From this perspective appropriation of indigenous knowledge is based on the approval of the indigenous people, safeguarding their rich cultural heritage and maintaining their identity (Pattanaik and Jha 2007).It is on this context that the ICT has a potential towards integrating indigenous knowledge to the formal knowledge structure by maintaining their right and identity along with their cultural heritage.

It can be argued that socio-technical processes that address the needs of indigenous peoples are bound to be more effective and meaningful to the indigenous people. Thus, transferring and sharing of knowledge should not be about coming up with technological fixes to [their] problems, or passing along ICT for [them] to adopt. It should be about acknowledging that they have their own effective knowledge, resources and practice management systems (Sillitoe 1998). Incorporating IK into larger domain of knowledge assumes that coexistence of different knowledge structures is conceivable, and that they can complement each other. To avoid fallacious dichotomization of knowledge structures, it is important to understand that the past continues to influence the present and the present influences the narration of the past. Because different knowledge systems represent different points on a continuum, it is necessary to work toward synthesis of the different systems, both indigenous and non-indigenous. ICT dissemination should be a continuous search for jointly negotiated advances rather than as a top-down imposition. It should seek systematic accommodation of IK into formal knowledge system. While this is not an easy task, it is one that requires formulation of strategies that meet demands and

challenges (Sillitoe 1998; Viergever 1999). DEAL experience of using ICT tool to accommodate tacit knowledge provides a path towards this.

IX. ROLE OF ICT IN PROMOTING IDENTITY AND RIGHTS OF THE INDEGENOUS

An application of ICT that could have particular usefulness for indigenous peoples is a system for cultural preservation. According to Dillon (2004) and Jones (2001) ICT can serve the following purposes when applied to indigenous communities in right way, (a) intrinsic individual development, (b) improving the economy of the people, and (c) immersion into a culture. As part of the cultural immersion, technology needs to be introduced in a fashion so that the user can understand it and be able to participate in it at some level. Technology in this way can stands along the side of the culture of the indigenous. The socio-technical issues involved in it must ensure that the users are exposed to processes that acknowledge the sources of empowerment and disempowerment in society. It should create space and capacity for the users to engage in self reflective knowledge production. Above all the ICT interface should allow users to produce and control knowledge about themselves and their communities. As a result they will be able to resonate with their culture and traditions and contribute to a universal knowledge system in a process that is viewed as both intergenerational and holistic (Cavallo 2000; Mosha 1999; Reiser 2001).

X. LESSONS FROM DEAL: INSIGHTS TO RESOLVING THE DIALECTIC BETWEEN IDENTITY, RIGHTS AND KNOWLEDGE

Facilitating meaningful interface at the ICT front is likely to have a positive impact on the process of integration of indigenous knowledge and conflict of identity. This requires that there should be provision for reciprocal flow of ideas, information and mutual decision-making (Mundy and Compton 1995; Grenier 1998; Viergever 1999). The ICT has an inherent capability of preserving explicit knowledge; it has difficulty with how to treat tacit knowledge. ICT output, no matter how well represented, is usually one-dimensional. Digital archiving of information encompassing text, audio, graphics and video is only the first step to cultural preservation. The second is placing the information in a meaningful knowledge management system where it can be used and maintained by the community (Michael and Dunn 2007). DEAL experience provides some insights towards this.

The focus of DEAL project was to create a digital knowledge base on agriculture and rural livelihood domain by involving various actors in the co-creation of content process. It also aimed at building a social network through the facilitation of ICT tools among various actors working in the domain. DEAL assumed that enablement of a co-creation of content process will eventually constitute a social network leading to an electronic community and finally to a self-sustaining community network in the agricultural domain in the condition that, the infrastructural facilities require to construct community network will be available to the actors as time proceeds. With this framework it involved various agricultural scientist, village level extension worker and experts, agricultural universities, research institutes, Non-Governmental Organizations (NGOs), international bodies working in agriculture and rural livelihood domains.

The broad objective of the project was to initiate recursive, reflexive and self-reinforcing knowledge creation and network building process. DEAL tried to resolve the complexities within the structure of knowledge at its initial stages of inception. Creating a knowledge space and content management system in the domain required that both the explicit and tacit form of knowledge are captured and integrated. It conceptualized the two streams of knowledge (explicit and tacit) as "Gyandhara" and "Ganagyan." The Gyandhara was based on the assumption of formal-scientific knowledge, where as Ganagyan focused on all the localized, everyday and context based knowledge which is tacit in nature.

In order to create a content management system in the stream of Gyandhara, DEAL solicited the participation of a few KVKs working in the region. With the help of the Zonal Coordination Unit IV of ICAR five KVKs of the region (KVK Daleep Nagar, Kannauj, Pratapgarh, Raibareily and Unnao) were selected to participate in the project. The aim was to promote a space where experts of the domain are expected to participate in sharing and exchange their knowledge in a digital architecture. It was proposed that the scientist at the KVKs will provide the content for the portal and use the content of other experts and scientists working in the domain (particularly of other participating KVKs) that will be available at the portal which will be enabled through internet. This it self was a barrier as only two of the KVKs has internet connection (one with high band width and the other low). As an alternative an agricultural scientist was deployed at each KVK by DEAL to facilitate the process of content creation. The agricultural expert of the DEAL manually and verbally collected the content which was then edited and added to the portal. It was proposed that the agricultural scientist will update the content at the KVKs computer manually through data transfer in a CD. The process at the beginning worked smoothly. But as the complexity of the portal grew the agricultural scientist found it difficult to update. Thus a computer expert was appointed to update the content at the KVKs. At present the content is collected from the agricultural scientist at KVK by the agricultural experts manually and the updating of the portal at the KVK are done manually.

However what interest most to us and to the IK is the content management system that was developed to capture and digitize tacit knowledge Ganagyan. Due to its very nature Ganagyan contrary to the Gyandhara required different orientation. The first and foremost task in this venture was to

collect the tacit knowledge from the local actors. For this the agriculture expert of DEAL visited the local actors and in the beginning collected them in the text format. It is noteworthy to mention that a majority of the local actors when visited queried about solutions to their difficulties in everyday practices relating to agriculture and livelihood issues. Looking at this complexity DEAL decided to develop a blog at its portal, where the users can put their questions and get answers to it. Simultaneously it was felt that developing a text based blog might not be of much help to the local actors as majority of them have no skill in computer. Thus it was decided to develop an audio blog for the same purpose. The Figure 1 shows the architectural design of the audio blog named as Kisan Blog at DEAL portal.

The Kisan blog is based on audio interface. A person interested to put up any question, can do the same either by recording it in an electronic device or directly through a microphone attached to a computer linked with internet. At the top of the page a link is provided containing information on how to use the blog. For posting any query the user has to log on into the page. It can be done by clicking on the option login at the webpage. It has a fixed login and a password for common user which is given at the bottom of the login page. Each participating KVKs has been allotted with separate logins. This has done to ensure their identity. Once a user login to the page he can post his query either directly or can upload a file already recorded on an electronic device. The usual time period for direct recording is 250 sec, for upload the audio file has to be below 2mb in size. The Figure 2 below shows the recording procedure step 1at the Kisan Blog. After the recording is done, the user can check the same for quality, clarity etc. by clicking in the option play.

There are additional features to improve the quality of recording which can be accessed by clicking the right mouse button. Once the recording is done the user can submit the same by clicking the submit button. Hence a new page appears where the user can give a title to his audio clip and any other additional information relation to it on text format. An on_screen key board (in Hindi) is available for the same purpose. He also can provide identity such as name, place etc in text format. This supplementary information appears at the blog in text format along with the audio clip when it is transmitted in air. When this is done it is automatically stored at the server of DEAL.

However to be on air it requires an administrators permission. The administrator has separate login id and follows the same login procedure. He then filters the question and puts it on air. The filtering is usually done by the agricultural experts of DEAL. This has been done with an intention to ensure that the questions asked and the answers provided are valid. Once on air the query appears on the blog site with title, identity and the audio. Users interested in answering the query can do so by clicking on the option "number of suggestions" which appears at the bottom left of the query.

FIGURE 1. ARCHITECTURAL DESIGN OF THE KISAN BLOG

FIGURE 2. RECORDING PROCEDURE: - STEP 1

FIGURE 3. RECORDING PROCEDURE: STEP - 2

To answer a query one can follow the same recording method. Answer a query does not require any login. The names of the most recent users along with their identity who provide suggestions are categorized and appear at the top of the main page of the Kisan Blog. This both ensures authenticity of the suggestion as well as it protects the identity of the persons and provides a form of recognition to them.

XI. IMPLICATIONS FOR INDIGENOUS KNOWLEDGE

If developed an audio blog such as Kisan Blog has potential capabilities to capture and digitize indigenous knowledge simultaneously protecting the identity and culture of the indigenous due to following reasons: (a) it allows capturing the tacit knowledge in its pure form. The distortion of the knowledge does not occur as it is mostly in audio format and is directly added to the portal (b) it is based on easy to use and easy to learn mechanism (c) it guarantees the identity of the content providers and gives recognition to them.

XII. CONCLUSION

ICT facilitated content management provides access to a wider range of extending possibilities. It has ability to draw more

peripheral participants, and provide access to a wider set of peoples. However one can not deny the importance of infrastructural resources, social structures and norms where the both the actors and socio-technical processes operate. Cultural preservation cannot be achieved by ICT alone: it also requires the spiritual element behind the history to be actively reinvigorated into a community to make its presence felt in a long-lasting manner. Culture is something that is alive and ever-changing. In brief, it is not machinery that transforms society, repairs institutions, builds social networks or produces democratic culture; it is people who make this happen. What has been presented in this paper is a way forward. Only by getting communities involved in the development of applications, ICT adoption is likely to bring its myriad benefits to the everyday life we live in.

ACKNOWLEDGEMENT

We greatly acknowledge the help rendered by Sugatha Chaturvedi, Project Associate, for going through the manuscript and collecting materials for the paper. We appreciate Anupam Bhattacharya, Project Associate, who has developed and conceptualized the Kisan Blog. All the images presented in the paper are provided by him.

REFERENCES

[1] A. Jones, "Theme issue: developing research in technology education," Research in Science Education, Vol. 31, 3-14: 2001.

[2] Abdallah Laroui, "Cultural identity and modernity," International Social Science Journal, 118: 551-559, 1998.

[3] A. E. Lieberman, "Taking ownership: strengthening indigenous cultures and languages through the use of ICTs," 2003. Retrieved January 2, 2005, from INTERNET:

http://learnlink.aed.org/Publications/Concept_Papers/taking_ownership.pdf.

[4] B. G. Miller, Invisible Indigenous: The Politics of Nonrecognition. Lincoln, NE: University of Nebraska, 2003.

[5] D. Cavallo, "Emergent design and learning environments: building on indigenous knowledge," IBM Systems Journal, vol.39(3-4), 768-781, 2000.

[6] D. Maybury-Lewis, Indigenous Peoples, Ethnic Groups, and the State (2nd Ed.). Boston: Allyn and Bacon, 2002.

[7] Debashis Pattanaik and Munmun Jha, "Knowledge and identity among indegeneous people," The Oriental Anthropologist, vol.7(2): 199-212, 2007.

[8] Deep Narayan Pandey, "Equity of Knowledge," 2000 from INTERNET:

http://www.sciencemag.org/cg1/elleters/28//5450/44.

[9] G. J. Dei, "Rethinking the role of indigenous knowledge in the academy," International Journal of Inclusive Education, vol.4(2): 111-132, 2000a.

[10] G. J. Dei, "African development: the relevance and implications of "indigenousness," in G. J. Dei, B. L. Hall, and D. G. Rosenberg (Eds.), Indigenous Knowledge in Global Contexts: Multiple Readings of Our World (pp. 70-86). Toronto, ON: University of Toronto, (2000b).

[11] Ian S. McIntosh, "Are There Indigenous People," Cultural Survival Quarterly, vol.24 (3): 4-7, 2000.

[12] Ian S. McIntosh, "Intellectual Property Responsibilities," Cultural Survival Quarterly, vol.24 (4): 4-5, 2001.

[13] Jim O.' Neil, "Scientific method vs. 'knowledge,'" 2000, from INTERNET:

http://www.sciencemag.org/cgi/eletters/287/5450/44.

[14] Judy Iske-Barnes, and Deborah Danard, "Indigenous Knowledges and World View: Representations and the Internet," in L. E. Dyson, M. Hendricks and S. Grant (Eds.), Information Technology and Indigenous People (pp. 27-37). London: Information Science Publishing, 2007.

[15] Katina Michael and Leone Dunn, "The Use of Information and Communication technology for the Preservation of Aboriginal Culture: The Badimaya People of Western Australia," in L. E. Dyson, M. Hendricks and S. Grant (Eds.), Information Technology and Indigenous People (pp. 170-174). London: Information Science Publishing, 2007.

[16] Kamal Puri, "Is traditional or cultural knowledge a form of intellectual property?" Oxford Electronic Journal of Intellectual Property Rights, 2000, from INTERNET:

http://www.oiprc.ox.ac.uk/EJWPO100.pdf.

[17] Kelly Bannister and Katherine Barrett, "Challenging the status quo in ethnobotany," Cultural Survival Quarterly, vol. 24 (4): 10-13, 1995.

[18] L. Grenier, Working with Indigenous Knowledge: A Guide for Researchers. Ottawa, ON: International Development Research Centre, 1998.

[19] L. M. Semali and J. L. Kincheloe. "Introduction: what is indigenous knowledge and why should we study it?," n L. M. Semali and J. L. Kincheloe (Eds.), What is Indegenous Knowledge: Voices from the Academy (pp. 3-57). New York: Falmer, 1999.

[20] L. Robyn, "Indigeneous knowledge and technology, American Indian Quarterly, vol. 2 (6), 198-220, 2002.

[21] L. Todd, "Aboriginal narratives in cyberspace," in M.A. Moser and D. Macleod (Eds), Immersed in Technology: Art and Virtual Environments (pp. 179-194). Cambridge, MA: MIT Press, 1996.

[22] M. Herselman, and K. G. Britton, "Analysing the role of ICT in bridging the digital divide amongst learners," South African Journal of Education, Vol. 22(4): 270-274, 2002.

[23] M. Vierger, "Indigenous knowledge: an interpretation of views of indigenous people," in L.M. Semali & J. L. Kincheloe (Eds.), What is Indigenous Knowledge: Voices from the Academy (pp. 333-343). New York: Falmer Press, 1999.

[24] Mac Chapin, "Losing the way of the great father," New Scientist, Vol.131 (1781): 40-44, 1991.

[25] Mita Manek and Robert Lettington, "Indigenous knowledge rights: recognizing alternative world views," Cultural Survival Quarterly, vol.24 (4): 8-9, 2001.

[26] Nuffic, "About indigenous knowledge," 2000, from INTERNET:

http://www.nuffic.nl/ik-pages/about-ik.html.

[27] P. Dillon, "Trajectories and tensions in the theory of information and communication technology in education," British Journal of Engineering Education, Vol.52(2): 138-150, 2004.

[28] P. Mundy and J. L. Compton, "Indigenous communication and indigenous knowledge," in D.M. Warren, D. Brokensha, & L. Jan Slikkerveer (Eds).The Cultural Dimension of Development: Indigenous Knowledge Systems (pp. 112-123). London: Intermediate Technology, 1995.

[29] P. Sillitoe, "Globalizing indigenous knowledge," in P. Sillitoe, A. Bicker, and J. Potteir (Eds.), Participating in Development: Approaches to Indigenous Knowledge (pp. 108-138), London: Routledge, 2002.

[30] P. Wangoola, "Mpambo, the African multiversity: a philosophy to rekindle the African spirit," in G. J. Dei, B. L. Hall, and D. G. Rosenburg (Eds.), Indigenous Knowledge in Global Contexts Multiple Readings of Our World (pp. 265-277). Toronto, ON: University of Toronto Press, 2000.

[31] Paul Alan Cox, "Will tribal knowledge survive the millennium?" Science, Vol. 287: 44-45,2000 http://www.sciencemag.org/cgi/content/full/287/5450/44.

[32] R. Reiser, "A history of instructional design and technology: Part II: A history of instructional design," Educational Technology Research and Development, vol.49 (02), 57-67, 2001.

[33] R. S. Mosha, The Heartbeat of Indigenous Africa: A study of Chagga Educational System. New York: Garland, 1999.

[34] S. McGovern, Education, Modern Development, and Indigenous Knowledge. New York: Garland, 1999.

[35] Shailendra Sharma, "Should the local knowledge survive? For whom?," 2001, from INTERNET: http://www.sciencemag.org/cgi/content/full/287/5450/44.

[36] T. Mammo, The Paradox of Africa's Poverty: The Role of Indigenous Knowledge, Traditional Practices, and Local Institutions: The Case of Ethiopia. Lawrenceville, NJ: The Red Sea, 1999.

[37] Ulrich Beck, Ulrich, Risk Society. London: Sage, 1996.

[38] United Nations, "Standard setting activities: evolution of standards concerning the rights of indigenous people," Economic and Social Council, E/CN.4/Sub.2/AC.4/1996/2,1996, from INTERNET: http://www.unhchr.ch/Huridocda/Huridoca.nsf/e06a5300f90fa0238025668700518ca4/2b6e0fb1e9d7db0fc1256b3a003eb999/$FILE/G9612980.pdf.

[39] United Nations, "Declaration on environment and development," General Assembly, A/CONF.151/26 (Vol. I),1992, from INTERNET: http://www.un.org/documents/ga/conf151/aconf15126-1annex1.htm.

[40] Wanjira Kinuthia, "Instructional design and technology implications for indigenous knowledge: Africa's perspective," in L. E. Dyson, M. Hendricks and S. Grant (Eds.), Information Technology and Indigenous People (pp. 3-57). London: Information Science Publishing, 2007.

[41] World Bank, "What is indigenous knowledge?, 1997' from INTERNET: http://www.worldbank.org/afr/ik/basic.htm.

Interface, Perception and Visualization

[OPAALS Conference 2007]

Piazzalunga, Renata
Perception and Cognition
Research Institute for Technology and Innovation - IPTI
São Paulo, Brazil
renata@ipti.org.br

Keywords: perception, interface, visualization, phenomenology.

The interface must be conceptualized as an entity communicating with the individual (Laurel, 1991). That is, designing an interface does not simply mean giving shape to information connected to a software program, but exploring the communication mechanisms established between a virtual entity, as revealed via the interface, and an actual entity: the individual who is interacting with the program.

One of the lines of research in our group is concerned with investigating the relationship between architecture and digital spaces (Piazzalunga, 2005). This is relevant because it sheds some light on how we treat the interface.

The first discussions about environment modeling showed us that the graphic interface is nothing more than the space of action and interaction of the user. In this sense there is a resemblance between the creation process of concrete, architectonic space and digital spaces. In this way, we were able to conceive the interactive relationship between user and space in a more comprehensive way. We also noticed that the user of virtual spaces not only interacts with the environment through the interface but he/she also lives, experiences, perceives and "mentally inhabits" virtual space.

If we consider, for example, concrete traditional school settings, classrooms should be designed in line with the pedagogical principles of each educational context, as well as with a series of contingent architectural conditions such as thermal comfort, acoustic comfort, number of students per class, lighting conditions, aesthetic qualities and other considerations to ensure the coherence of the conceptual process of this space. If we draw a comparison with a virtual learning environment, all the previously mentioned general aspects of the model (co-existence of distance education and traditional instruction; e-learning environment; collaborative and interactive learning environment; mapping student's activities) were considered in the process of creating our environment.

In addition to this, the development of an interface project for cyberspace must consider the peculiar dynamics of its configuration, as well as the properties to be implemented that are possible and accessible from a technological point of view. That is, from the spatial configuration point of view one of the peculiarities consists of the fact that this space is constantly submitted to new configurations and re-configurations of its forms.

In the case of virtual environments, the interface serves as a gateway to access users' cognition of such systems, given that the mental processing involved in the interpretation of the informational contents depends on the stimuli that lead to perception motivated by this interface. So, we might state that interface design should be connected to "cognitive ergonomics", metaphorically speaking, that takes into account the interaction between the individual and the virtual environment, based on the exchange of stimuli between users and cognition, via the interface

From this viewpoint, we support the thesis that the designing of really qualified interfaces leading to significant experiences in the virtual space, should necessarily concern the correct understanding of the perceptive process, a trend widespread in contemporary science of relating cognitive science to phenomenology. We are therefore here discussing the understanding of perception through the perspective of phenomenology. Two other models should be remembered: the serial and the connectionist models. Although important, they do not interest us here as they do not consider subjective experiences on the part of the individual. They are conceived from a representational view of perception. They are based on the principle that perception consists of taking characteristics of a pre-determined world and constructing a copy, or internal image of this world in the perceptual apparatus. Reviving phenomenology seems an appropriate strategy in line with current expectations, given that the principle of its conceptual formulation: *the analysis of consciousness in its intentionality*, functions as a counterpoint to the representational formulation of perception.

The formulations made by researchers such as the philosopher Hubert Dreyfus and the biologist Francisco Varela are examples of this relationship between the cognitive sciences and phenomenology. The idea that forms the basis of the thinking of these researchers is that the spirit

forms an organic unit with the body and with the environment. This assumption is very important and must be considered in the creation of virtual environments.

Hubert Dreyfus holds to the thesis that Husserl can be considered as the precursor of the classical theses on Artificial Intelligence. Varela develops a research program referencing classical phenomenology known as the "naturalization of phenomenology". At the beginning, the intention of the project appears to be to create a natural science, a "neuro-phenomenology", incorporating concepts of phenomenology and concepts of the cognitive sciences.

The fundamental assumption orienting phenomenological research subverts the natural attitude characterized by the belief in the absolute external nature of the world's things. For phenomenology, it is **consciousness** that constitutes the meaning of the world's objects. Consciousness is revealed through the phenomenon: the consciousness of the perceptive experience involved in the observation of the phenomenon and the opening of consciousness to the world, that is to say, the intentionality of consciousness. When for example, I observe a scene of a child riding a bicycle, from the start I am paying attention to the movement in this action: I observe that he turns the handlebars and pedals, etc. But, generally, I am not aware of the perception involved in watching the child's activities, in other words, my perceptive experience involved in the observation of the phenomenon: I am concentrated, involved, thoughtful, etc. My gaze is concentrated on the external action and, therefore, on the direct perception of the object, and not on the internal action, on the perception of the essence of the phenomenon. One could say that this change in our way of looking is, in a simplistic way, what the phenomenological method proposes.

As a philosophical school, phenomenology is not a localized movement nor can it be reduced to a system. "It is always thought of as, even among its philosophers, as a research in movement that can never be summarized as a finished list of precepts and rules" [1].

This being so, we can take the general fundamentals of the phenomenological doctrine introduced by Edmund Husserl (1859) as a reference. In 1900, Husserl published *Logical Investigations I*, with which he became considered the founder of phenomenology as a philosophical method. The fundamental assumption of phenomenology, ***the intentionality of consciousness***, was developed by Franz Brentano (1838), of whom Husserl was a student. Husserl's originality consists of having adopted this assumption for the consolidation of a philosophical method. The general fundamentals of phenomenology were to be dealt with by starting with the presentation of the key concepts, around which Husserl based his method, related to the specific sense of how phenomenology conceives its philosophy: as the *analysis of consciousness in its intentionality*.

For phenomenology, consciousness is always the consciousness of something. It is the source or beginning of the other realities in the world. For Husserl, consciousness is a chain of lived experiences, with each experience having its own essence, which in turn defines the way in which the object is revealed to the consciousness. These essences are acts of the consciousness, such as perception, remembering, emotion, etc. Thus, "the analysis of consciousness is the analysis of the acts by which consciousness itself relates to its objects; or, the ways by which these objects are revealed to consciousness. The acts of consciousness, or the ways by which the objects are revealed to consciousness form the *intentionality of consciousness*" [2]. That is, the intentionality of consciousness is the way in which consciousness opens itself to the world of experiences in order to extract its essences.

Pure phenomenology is not a science of facts, but of essences (eidetic science). To get to the essences, once must avoid the affirmation or recognition of the reality and assume the attitude of a spectator, interested in only gathering the essence of the facts, through which consciousness reports to reality. Consciousness should, therefore, assume an attitude of a *disinterested spectator* in relation to the world. This distancing is possible through a methodological artifice created by: the ***epoché***. The *epoché* consists of a phenomenological practice; an internal or mental gesture of reduction (eidetic), which leaves the existence of the world in suspension. Through *epoché* one can reach the realm of subjectivity, distancing oneself from the world and assuming the role of a disinterested spectator of the facts, thereby creating a methodological strategy to reach the essences. The attention of the disinterested spectator is turned not toward the world of his reality, but to the phenomena that reveal this world to the consciousness.

According to the phenomenological analysis, the object is not part of the lived experiences. The subject that gives intent to the object does not become an integral part of the object; neither does the object given intent become part of the subject. "The world that in the attitude of *epoché* becomes a transcendental phenomenon is understood from the start as the correlation of occurrences and intentions, of acts and subjective faculties, in which the sense of its unity is constantly changing and which progressively assumes other senses" [3]. Husserl attributes the crisis in the sciences to not taking into account the realm of subjectivity.

The trend widespread in contemporary science of relating phenomenology to cognitive science consists of consolidating a model that emphasizes a practical dimension of phenomenology, giving it potential from a pragmatic dimension.

Supporting an approach that reconciles at least three of these models in a new vision of the perceptive process, Hubert L. Dreyfus says in *The Current Relevance of Merleau-Ponty's Phenomenology of Embodiment:* "I will suggest that neural-network theory supports Merleau-Ponty's phenomenology, but that it still has a long way to go before it can instantiate an intentional arc" [4]. This suggestion by Dreyfus results from the analysis of how Merleau-Ponty thinks of his phenomenological subject. Merleau-Ponty's phenomenal body, our cognizant apparatus

which makes us conscious of the world, is not only a psychic entity, and not only a physiological one, but rather it is a synergic system that takes on and connects the functions of a physical, psychic and cultural order, that define the general movement of a being in the world. “Consciousness projects itself in a physical world and has a body, just as it projects itself in a cultural world and has habits” [5].

The intentional arc is understood as the personification of the interconnection between perception and action, in the phenomenological context. It is the intentional arc “that unites the senses, uniting the senses with intelligence, and sensibility with the motor functions” [6]. In other words, it is the intentional arc that certifies acts of consciousness, certifies experiences of the phenomenological being in the world. According to Merleau-Ponty: “the life of consciousness – cognizant life, the life of desire or the perceptive life – is supported by an ‘intentional arc’ that projects around us our past, our future, our human environment, our physical situation, our ideological situation, our moral situation, or else leads us into placing ourselves under all these aspects” [7].

The instantiation of the intentional arc of which Dreyfus speaks perhaps occurs in cyberspace, which would promote ideal conditions for its occurrence, as it is a domain in space where phenomenological action and perception gain the status of being one at the same time as they are constituents of this same space.

Since the interface measures the informational content offered the user, putting into play his understanding, what he receives depends on the information displayed and on the user’s perception of it. Therefore, within this perspective, the “ideal” interface would be that which is able to neutralize in the most efficient way possible the differences between the external and the internal, it would be that which manages to touch the user’s perception most directly in such a way that the user feels his body to be an integral part of this simulated context. The experience brought about by means of the interface should touch the essence and being in the most organic, natural and intuitive way possible.

References

1] Begout, Bruce. Un air de famille. La Phénoménologie: une philosophie pour notre monde. In MAGAZINE LITTÉRAIRE. Paris, No. 403, Nov. 2001, pp. 20-22.

“Elle s’est toujours pensée, même chez sés épigones, comme une recherche en mouvement qui ne pouvait jamais donner lieu à une liste close de préceptes et de règles”.

2] Abbagnano, Nicola. História da Filosofia. Lisboa : Editorial Presença, Volume XIV, 1993. p. 93.

3] Idem, ibidem. P. 93.

4] Dreyfus, Hubert L. The Current Relevance of Merleau-Ponty's Phenomenology of Embodiment. 1996.

Available at: http://hci.stanford.edu/cs378/reading/dreyfus-embodiment.htm
Accessed on: 18/10/2003
“I will suggest that neural-network theory supports Merleau-Ponty's phenomenology, but that it still has a long way to go before it can instantiate an intentional arc”.

5] Merleau-Ponty, Maurice. Fenomenologia da Percepção. São Paulo: Martins Fontes, 1999. p.192.

6] Idem, ibidem. p.190.

7] Idem, ibidem. p.190

The Impact of DEAL on Community Networks : A Case Review

[OPAALS Conference 2007]

Radhika Rajagopalan
Research Associate
Indian Institute of Technology, Kanpur
Kanpur, India
radhika@iitk.ac.in

Runa Sarkar
Industrial & Management Engineering
Indian Institute of Technology, Kanpur
Kanpur, India
runa@iitk.ac.in

***Abstract*— The DEAL (Digital Ecosystem for Agriculture and Livelihood) project is a step towards addressing issues related to designing an ICT intervention that leads to sustainable development. This paper studies the impact of DEAL on existing community networks in the field agricultural extension in rural areas around Kanpur. We find that while DEAL has not led to a rapid expansion of the network, it has successfully deepened ties and created new weak links between existing actors. As a result, it is expected that this intervention would positively affect social capital and lead to the formation of a digital ecosystem in the agricultural knowledge domain.**

Keywords- Digital Ecosystem, Weak links, Community Networks, Agriculture

I. THE DE APPROACH IN A COMMUNITY NETWORK: THE CASE OF THE DEAL

The DEAL (Digital Ecosystem for Agriculture and Livelihood) project is a step in towards addressing these issues by assembling a technology enhanced agricultural extension intervention in a DE framework. Conceived by IIT K and funded by Media Lab Asia, DEAL is an ICT enhanced network built on an existing framework of tele-centers in rural institutes, village schools, village level agriculture extension centers (KVKs) and other deployment partners. The project aims to create a digital knowledge base by involving the various actors in the existing system in the content creation process and making this knowledge accessible to farmers and other agricultural practitioners. The entire process of content creation and dissemination is capable of self generation, node independence and self-sustainability using an electronic medium. The moderating node in this system is IIT Kanpur- providing the collaboration and collation technology platform, skills and resources to assist knowledge flows through the network. The presence of Government agencies helps build trust in the network. The agricultural experts and educational institutions are responsible for verification of content generated.

Field deployment of the DEAL project was between December 2006 and June 2007. Following this, a study was conducted at 4 KVKs in September 2007, to assess the effect DEAL has had on information flows. A total of 20 agricultural scientists from across KVKs and 5 project team members from IIT Kanpur were interviewed. We elicited responses from actors how exposure and use of different facets of the DEAL project altered their relationships with existing nodes, or if there was a deletion / addition of new nodes. Each KVK scientist were asked to describe the existing links each KVK had with different actors in the extension system, and how they viewed the potential of the DEAL in enhancing their access to information flows in the network. The questions about DEAL were open ended and unbiased, and respondents were encouraged to give their honest impressions and opinions about the project, its strengths and weaknesses, the potential for forming new associations, the benefits thereof and lacunae in implementation

Table 1 lists the members who are part of the network (actors), both before and after the DEAL intervention with their respective role.

This work has been funded by The European/ Union's 6th Framework Programme of research under the project Open Philosophies for Associative Autopoietic digitaL ecosystemS (OPAALS)

TABLE I. ACTORS IN AGRICULTURAL EXTENSION

NODE	ROLE PLAYED
ICAR	Government Body
ICDS	Government Body
ICRISAT	Research Institute
IIPR	Research Institute
NSI	Research Institute
CSA	Educational Institution
NDU	Educational Institution
ZCU	Zonal Co-ordination Body
KVK(P)	KVK (agricultural extension centres)
KVK(D)	KVK
KVK(R)	KVK
KVK(K)	KVK
SAC(P)	Scientific Advisory Committee
SAC(D)	Scientific Advisory Committee
SAC(R)	Scientific Advisory Committee
SAC(K)	Scientific Advisory Committee
ZF1	Zilla (Ditrict) Line Functionaries
ZF2	Zilla Line Functionaries
ZF3	Zilla Line Functionaries
ZF4	Zilla Line Functionaries
IITK	Educational Institution
PNU	Educational Institution
Fr1	Farmers
Fr2	Farmers
Fr3	Farmers
Fr4	Farmers
KV1	Kisan Vidyalaya (village school)
KV3	Kisan Vidyalaya
KV4	Kisan Vidyalaya
Pvt	NGO
NBFGR	Research Institute
BK	Bank

TABLE II. EXPLANATORY NOTE

ICAR	INDIAN COUNCIL OF AGRICULTURAL RESEARCH
ICDS	Integrated Child Development Services
ICRISAT	International Crops Research Institute for the Semi-Arid Tropics
IIPR	Indian Institute of Pulses research
NSI	National sugar Institute
CSA	Chandra Shekar Azad Agricultural University
NDU	Educational Institution
ZCU	Zonal Co-ordination Body
KVK	Krishi Vigyan Kendra
KVK(P)	KVK at Pratapgar
KVK(D)	KVK at Dileepnagar
KVK(R)	KVK at Rae Barelli
KVK(K)	KVK at Kannauj
SAC(P)	Scientific Advisory Committee
IITK	Indian Institute of Technology Kanpur
PNU	Pant Nagar Agricultural University
KV1	Kisan Vidyalaya
NBFGR	National Bureau of Fish Genetic Resources

The following network diagram, prepared in NetDraw[1], represents the ties that were present before the implementation of DEAL.

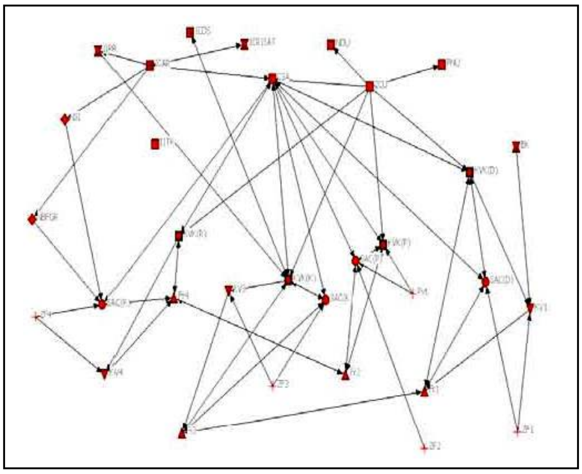


Fig.1: Network Ties before DEAL

The shapes of actors in the network are based on their role, i.e. KVKs, Research Institutes, Government agencies, etc. The thickness of lines between agricultural experts within the same KVK, or between a farmer and his respective KVK are examples of strong ties, while links between KVKs and NGOs are examples of weak ties. In the course of the analysis, we refer to preexisting , structure based links that individual nodes supply information to or draw information from (or both), as the ‘strong ties’. By this definition, all links sanctioned by the structural framework of the agricultural extension system can be denoted as strong links. However in practice, most of these channels are too infrequently used by nodes to be significant. To tighten our definition of strong ties, the agricultural scientists at each KVK were asked to indicate which of the available structural links were mandatory. Apart from that, they were asked to list the nodes in the said network that they had received information inputs from. In theory, though all KVKs can, by the extension structure, seek the help or advice of any national research or educational institute that are in the same zone , through the Zonal co-ordination unit. Thus, the potential for extended links is inherent in the system, but without frequent use these remain links only on paper. For instance, IIPR is linked to all the KVKs through the Zonal co-ordination unit, but only one KVK (at Dileepnagar) has directly consulted with experts from the institute. Similarly, there exist links between the KVKs and educational institutes like PNU and NDU, but these links are more or less dormant. The network shows the information flows within and across community. Here, the community is understood in terms of the village unit. So, within community linkages are those between actors in the same village – for eg, between the farmer of a village and the respective village KVK while across groups’ links includes links between actors from different villages – like the link between farmers of different villages. By the strata of operations classification, information flows between members of the same functional role also qualify– IITK is a member of the educational institutions group, KVK are part of the villages’ level functionaries, and the ZCU, ICAR are all implantation and monitoring agencies. In this above network diagram, we have represented the different sources of agricultural information and the interrelations, both formal and informal, between them. Formal links are characteristic of the reporting relationship between actors – for instance, in the case of a KVK and the ZCU (Zonal Co-ordination Unit), and informal links are characteristic of the social relations between actors – like relations between farmers of adjoining villages. We can characterize the reporting relationships between members into different layers – administrative, academic and functional. One observation here is that while there are well established and clearly defined relationships between members from the different layers, there are very few formal ties between the members of the same layer. For example, the relationship between the ZCU and a KVK, or between a KVK and farmer close and well directed, but there exist no direct links between the 4 KVKs. Communication is routed through the ZCU, and is conducted face to face at periodic zonal meetings.

Fig 2 represents the ties after implementation of the DEAL project. As is evident, IITK is the only completely new actor introduced into the framework. Its integration into the network is represented by the arrows between it and other nodes, signifying an increase in information flows.

[13] Software from Analytic Technologies

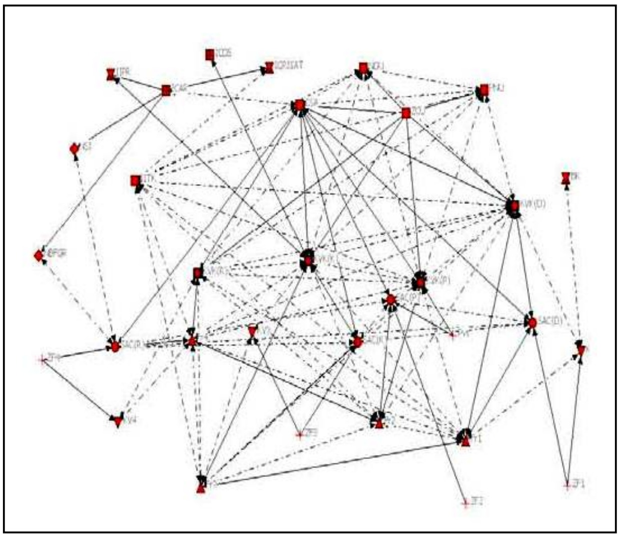


Fig.2: Network Ties post DEAL

The dotted lines represent ties that have been formed due to content co-creation and sharing by partners facilitated by IITK through DEAL, while the solid lines represent the preexisting network ties. By implication, ties formed through DEAL are mostly weak links. These are voluntary clusters of members who are from different groups.

Groups in the network can be understood at 2 levels – one, at the geographic level , which consists of members of different types (farmer, KVK , research institute) at a specific location , and the other is related to functional relationships. These could include academic ties, administrative reporting relationships (financial flows) or operational ties, for example, between KVKs. Linking together all the actors in dynamic relationships helps retain both strong and weak ties. We present here salient results[2] from the analysis done using Ucinet[3]. The total number of ties increased from 77 to 183, and no old ties were displaced. No old actors in the network were deleted after implementing the DEAL, while only one completely new node (IITK, the implementer) was added. What was observed was that several weak links were introduced between existing nodes, signifying greater interaction (and hence innovation), and a deepening of community relations. Thus, the ICT intervention has led to the enhancement of social capital (Granovetter, 1985, Coleman, 1988)

Another indicator of this increased interaction is the group reciprocity measures increasing from 0.3585 to 0.7745 from the pre DEAL to the post Deal scenario. Figures 3 and 4 depict the state of reciprocal ties between members across different layers. The red lines denote reciprocal ties and the blue lines the non reciprocal ties.

[2] Details of study available at IITK Deal site

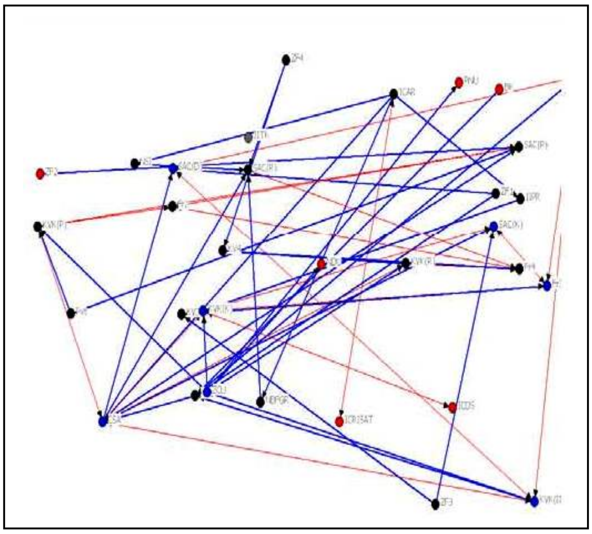


Fig.3: Reciprocal Ties pre DEAL

In the pre-DEAL scenario, except for the informal links between farmers of neighboring villages, the other links represented in the network are structurally determined. There are very few reciprocal ties between members of the same layer – for instance, the links between PNU and IIPR are both indirect and non-reciprocal. This lack of reciprocity across layers reflects the top-down nature of the reporting ties between actors from different layers – like in the relationship between the ZCU and a KVK. While a top-down approach is time and cost effective for information dissemination, in an extension setup it causes the network to become more centralized. However, studies in network architectures have shown that a centralized network is ineffective for knowledge sharing (Fahey and Prusak, 1998; Markus, 2001) as it is resource intensive, error prone and more crucially, does not potentially encourage re-deployment of the stored content. In the Indian context, with the Government reducing public investment in agricultural extension as well as privatizing its input system, there is a need to make extension and the overall technology transfer system more demand-driven and responsive to farmer needs. To achieve this, a more bottom-up approach is needed which empowers farmers and allows them to more effectively articulate their problems and needs to the research-extension system.

This is an issue that the DEAL aims to assess by providing opportunities for weak ties formation. The DE design of the system places special emphasis on voluntary participation, and as more members access the network the number of weak ties increases, and as these ties are mutual and voluntary, the reciprocal ties between the members of the same layer are positively affected.

Figure 4 shows the reciprocal ties that have developed after the implementation of DEAL. Here again, the red lines denote reciprocal ties and the blue, non reciprocal ties.

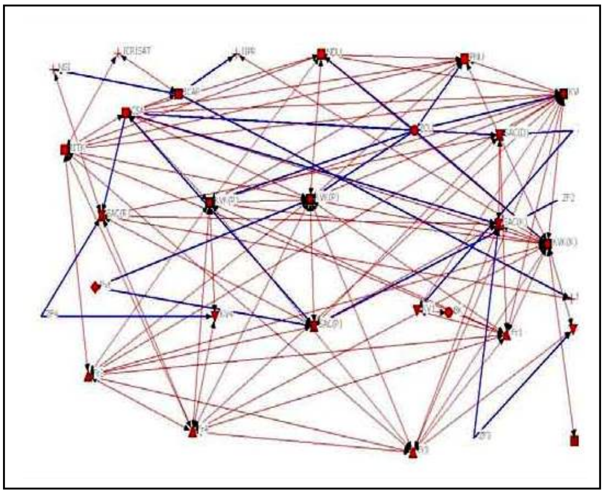


Fig.4: Reciprocal Ties post DEAL

As seen in the figure, there is an increase in reciprocal ties, with a simultaneous decrease in the number of non reciprocal ties. Increased reciprocity has a positive impact on content creation, while increased collaboration between members further enhances reciprocity.

We have seen how DEAL has increased the opportunities for reciprocal ties. We know observe the effect of DEAL on information flows within groups and between groups. Here, we make a further note on the ties within and between the different layers. Unlike conventional ICT interventions , which adopt either a top-down or bottom- up approach, DEAL focuses on increasing ties between members in the same layer, while also building links across the different layers. Figure 5 illustrates the links within and between groups in the pre DEAL scenario. The blue lines represent ties within groups and the red show ties between groups.

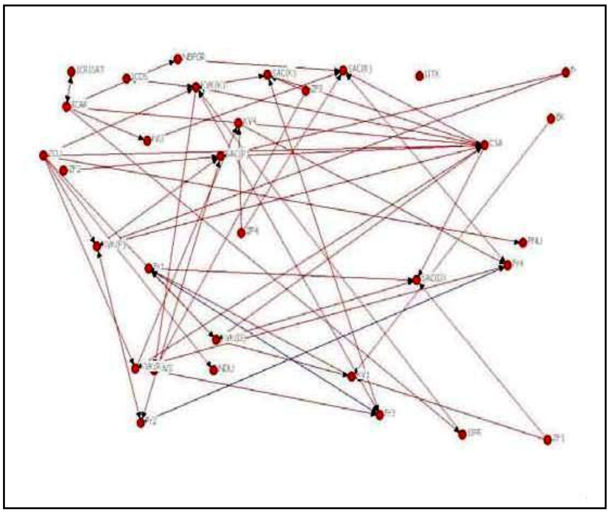


Fig.5: Pre DEAL links within and between groups

In the pre DEAL scenario, one feature of existing extension framework is that there are few links between members of the same layer. For instance, as the interviews revealed, there are very few direct informal links between individual KVK. Majority of the relationships that a node has is with members from the same geographic community, but very ties with members having the same role in the network. Thus, the local social capital stays locked into the local loop keeping it out of the network.

Figure 6 shows the effect of DEAL on the links within and between groups, that is horizontal information flows in the post-DEAL scenario.

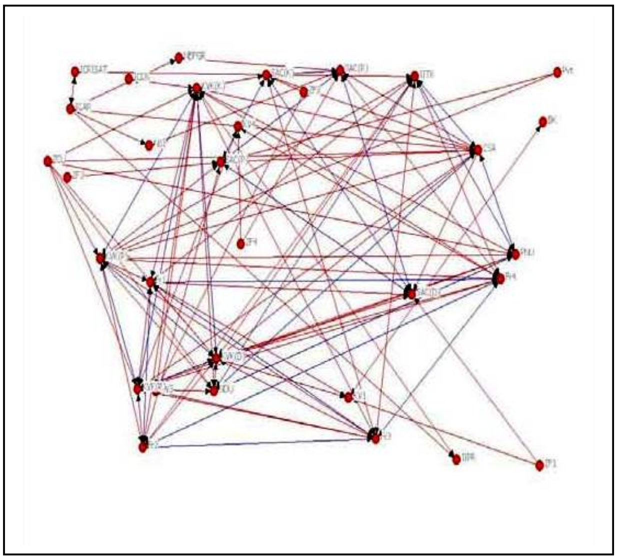


Fig .6: Post DEAL links within and between groups

There can now be seen several weak links between different nodes, which are at the same horizontal level. For example, DEAL had provided a platform for different KVKs to share their extension experiences with each other through hosting a website for each of them. Horizontal ties between farmers from disperse geographical areas are enhanced through the use of the kissan blog for sharing agricultural experiences. This is a distinguishing feature of a digital ecosystem as creating these ties lead to shared norms and values which make ICT interventions successful. In the case of DEAL, this would fulfil the goal of sustained and voluntary content co-creation.

CONCLUSION

This paper was an attempt to study the impact of DEAL on existing community networks in the field agricultural extension in rural areas around Kanpur. Through analyzing reciprocal ties between the various actors, we found that while DEAL has not led to a rapid expansion of the network, it has successfully deepened ties and created new weak links

between existing actors. We expect that since this intervention has led to increasing ties both between levels as well as among levels, it would positively affect social capital and lead to the formation of a digital ecosystem in the agricultural knowledge domain.

References

[1] Ben-Porath, Y., The F-Connection: Families, Friends, and Firms and the Organization of Exchange. Population and Development Review, (Mar., 1980),. Vol. 6, No. 1.: p. 1-30.

[2] Borgatti, S.P., Everett, M.G. and Freeman, L.C. 2002. Ucinet for Windows: Software for Social Network Analysis. Harvard, MA: Analytic Technologies.

[3] Chopde V.K, D.Parthasarathy , Building Social Capital: Collective Action, Adoption of Agricultural Innovations, and Poverty: Reduction in the Indian Semi-Arid Tropics.

[4] Chatterjee, J. and T. V. Prabhakar (2005). On to action- Building a digital ecosystem for Knowledge difussion in rural india. International conference on Knowledge Management, North Carolina, USA

[5] Coleman, J.S., Social Capital in the Creation of Human capital. The American Journal of Sociology, 1988. 94

[6] Fahey, L. and L. Prusak (1998). "The eleven deadliest sins of knowledge management." California Management Review 40(3): 265-76.

[7] Granovetter, M., Economic Action and Social Structure: The Problem of Embeddedness. The American Journal of Sociology, Nov., 1985,. Vol. 91, No. 3.: p. 481-510.

[8] Markus, M. L. (2001). "Toward a Theory of Knowledge Reuse: Types of Knowledge Reuse Situations and Factors in Reuse Success." Journal of Management Information Systems 18(1(Summer)): 57-93.

The Agent-based Digital Business Ecosystem:

towards the semantic web

[OPAALS Conference 2007]

A. Razavi, S. Moschoyiannis and P. Krause
Department of Computing
University of Surrey
Guildford, Surrey GU2 7XH, UK
{a.razavi, s.moschoyiannis, p.krause}@surrey.ac.uk

***Abstract* — by proposing two levels of virtualization, our approach offers the basic characteristics for semantic web on the Digital Business Ecosystem. On one hand, distributed transactions have been modeled in a virtual private transaction network (which supply the semantic links between business partners in a business activity at the same time), on the other hand, virtual service network, as a connected network, provide any software agents virtually to have a link to any other service provider. We distinguish transactions from services (and service providers) by considering Virtual Private Transaction Networks (VPTNs) and Virtual Service Networks (VSNs). These two virtual levels are optimised individually and in respect to each other. The effect of one on the other, can supply us with stability, failure resistance and small-world characteristics on one hand and durability, consistency and sustainability on the other hand.**

Keywords – semantic web, business transactions, P2P networks, e-commerce

I. Introduction

The Web has yet to realise its full potential. In this paper we focus on work to enable small to medium enterprises (SMEs) to engage in collaborative e-commerce, in a way that maintains an open competitive environment free of (the current) dominance by large organisations.

In achieving this goal, we see a convergence between our work and the goals of the Semantic Web. We argue that our work is foundational to realising the potential of adding semantic content to Web 2.0 (aka "Web 3.0). In section 2 of this paper, we establish the connection between our existing research and the key components of the Semantic Web. The main body of the paper elaborates our framework for supporting Digital Business Ecosystems of SMEs. Finally, we return to summarise how our work contributes to enriching the future of the Web.

The current paper, in contrast with conventional Semantic web literatures, does not talk about derived technologies, potential progresses or discussing about microformats, natural language search, data-mining, machine learning, and artificial intelligence technologies. This approach on one hand shows the enormous improvements on DE by applying the virtualization and more importantly tries to point out more controversial subject which is questioning the feasibility of the Semantic web. We show the feasibility of concept of Semantic web, on a business environment (DE), by using software agent. On the other hand, our virtualizations in the network, offer specific ability for this software agent, not only model its business activities by transactions, but also connect to any other user (or software agent).

II. The content, links, and transactions

According to W3C, the Semantic Web is an evolving extension of the World Wide Web in which web content can be expressed not only in natural language, but also in a format that can be read and used by software agents, thus permitting them to find, share and integrate information more easily [20]. Perhaps more precisely, W3C director Sir Tim Berners-Lee originally expressed the vision of the semantic web as follows:

"*I have a dream for the Web [in which computers] become capable of analyzing all the data on the Web – the **content**, **links**, and **transactions** between people and computers. A 'Semantic Web', which should make this possible, has yet to emerge, but when it does, the day-to-day mechanisms of trade, bureaucracy and our daily lives will be handled by machines talking to machines. The 'intelligent agents' people have touted for ages will finally materialize*." [ref: Berners-Lee, Tim; Fischetti, Mark (1999). Weaving the Web. Harper San Francisco, chapter 12. ISBN 9780062515872- ref: 24]

There are three important components in W3C's definition; content, links and transactions. First, we clarify these components in a Digital Business Ecosystem and then the design and complexities can be discussed. A Digital Business Ecosystem, as a service-oriented environment, should enable businesses (SMEs) to engage in distributed business transactions [4]. The structural atomic components of each business transactions are web services, which are described by some description language readable by computer programs as well as (sometimes less easily) by humans (description languages such as WSDL, SDL).

As we are involving with service providers in the deployment level [12], the content will be web-services and especially their descriptions on one hand and any composition of them as "on-the-fly-services" [11] on the other hand. The links provide access to/from service providers to consumers and/or other business to execute a transaction. Transactions are business activities between businesses (and consumers).

The complexity is that a Digital Business Ecosystem typically:

- is not a connected network.
- has a very dynamic nature. Especially because of the significant involvement of SMEs, this dynamicity will be in all levels of the semantic web: the nature of web services can change (content); the relationships between participants may change (links); and the nature and quantity of business activities between participants can change too.
- does not have potential for creating all sort of links between different content or service providers (this is one of the side-effects of it not being a connected network,).
- is relying on centralised controls for business activities (frameworks such as ws-businessactitivities, BTP and using ws –coordination framework).
- is not fully resistant against failures and errors.
- is not stable, with traffic bottlenecks and other uncontrolled parameters easily affecting the environment.

A. *First Step towards Sematic Web*

As a first step, we purpose a business network to enable networked organisations to engage in distributed business transactions [4] that realise their core business activities. If such a network is to support B2B interactions between different businesses it should be fully distributed (no central point of control for transaction or network operations), should also offer a consistent model for performing transactions and theoretically and practically each node should be able to have a link to the other node. This means it should be highly resistant to fragmentation.

In addition, given the nature of the internet, there is always the possibility of failure at the transaction level, which should be recoverable and such a procedure must be supported and assisted by the underlying network. The ability for choosing alternative paths/scenarios of execution is another important issue on the transactional level, as well as requirements for feasibility to create links between any two participants.

To reach such a goal, in our earlier approaches we introduced a fully distributed transaction model [1],[4] and localised coordination framework [3]. Furthermore by introducing distributed recovery management and concurrency control, the system's resistance against failure has been increased and a self recovering model has been designed [13]. To integrate the model into such a network, which has high connectivity, and to provide a dynamic topology, we introduce an agent based model. Each local agent is responsible for integrity and consistency of its data and local state. These software agents together provide virtual levels which offer a self-organised network, which reacts against changes during time and is able to change itself for adaptation or recovering against failures.

B. *Software agents on behalf of people (SMEs)*

In general, the Digital Ecosystem's software agents need to be able to:

- Gather and store local knowledge (local contents);
- Collect and accumulate knowledge from outside (external links);
- Manage content;
- Promoting the stored contents for outside and provide external links;
- Process business activities and transactions.

As we are working in a service-oriented business environment, the contents are services (their descriptions) and the structural descriptions of on-fly-services (composite services). Meanwhile we have not considered any limitations for using a customized XML or mixture of OWL, RDF and some derived XML from a XML schema for contents (we may address these issues in our later works).

III. Structure of a DE's Software agent

Figure 1, shows an overview of the local agent structure. Such a system includes a Local Web Services Informer, a Local Service Repository, a Web Service Information Investor, a Global Service Repository, a Web Services Promoter and a Local Coordinator. We describe each entity in more detail in the following;

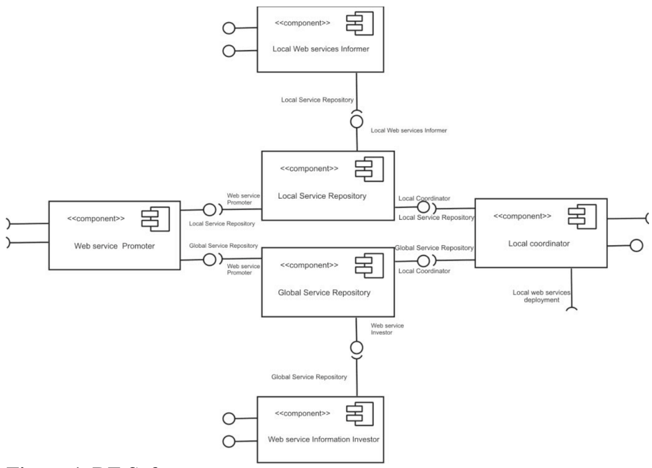


Figure 1. DE Software agent structure

A. *Local Web Services Informer*

A Digital Ecosystem is supposed to support any type of web service, with any protocol in a loosely-coupled manner (local autonomy for SMEs). In addition, it should support a proper commit protocol for long-running transactions (business activities as discussed in [1],[3],[4]).

In order to provide the Initiator of a transaction with the basic view about the web service which is to be to be used on its transaction, as well as the limitations / restrictions of the particular web service, we need to gather information about the

web service. This information can be provided by each web service after its creation in some description language such as WSDL and/or can be provided manually by the service provider (SME or any business) that provides the web service.

Furthermore, service providers may regularly change their web service protocol, parameters, etc. (in the case of SMEs, this is highly expected); therefore the possibility for updating this information is necessary too. As a result, we need to provide two interfaces for keeping this information in the local agent, as the component also requires an interface to the local repository.

B. *Local Service Repository*

The Local Service Repository keeps information about each local web services in the platform (SME). This information is some description of each web service (for example it can be in SDL or WSDL) and any extra information such as availability, last updates and so forth (which may help other SMEs to have a clearer picture of that particular web service), can be included too. In the first place (as a component-based approach), the Local Service Repository should provide an interface to the Local Web Services Informer, e.g. for accessing the local web service description records. The next interface provided by the Local Service Repository gives access to the Local Coordinator to use the web service descriptions for creating and running a transaction.

Any updates, modifications or even the creation of web services should be promoted (at least for other partners with whom they are collaborating in running a transaction). That's the main reason the Local Service Repository requires an interface to another component, namely the Web Service Promoter, whose purpose is to promote the web services to other agents.

C. *Web Service Information Investor*

The structure of the local agent we considered in Figure 1 looks symmetric for both the local and the global view of web services. Therefore the Web Service Information Investor, as a symmetric component for Web Service Informer, does a similar job but this time for global web services.

It provides two interfaces for creating a new web services record and updating the current web services. In addition, it requires an interface to the Global Service Repository (the symmetric component to the Local Service Repository).

D. *Global Service Repository*

Similar to the Local Service Repository, the Global Service Repository provides two interfaces: one for the Local Coordinator to access the web services record descriptions and the other one for the web services Investor to make changes on the Global Service Repository.

The first interface plays a critical role for the Local Coordinator in making the decision about the protocol and the method for applying it on the transaction model. At the other side of the local agent is another participant (SMEs) which may change its web services descriptions regularly and even service availability can be an issue too. The second interface is important too, as updating the Global Service Repository is crucial.

The Global Service Repository should be able to inform the Web service promoter, as soon as any changes occur for its records. That is why it requires an interface to the Web service Promoter.

E. *Web Services Promoter*

The Web service Promoter is an important part of the local agent, as it reflects the situation of the web services of a local agent and the web services of any other connected agents to that particular agent. This can be done by using two interfaces, which are provided for the Local Service Repository and Global Service Repository respectively. Meanwhile the Web Service Promoter requires two interfaces from the other agent to be informed of the latest situation of its local web services and any other web services, which are communicating with it.

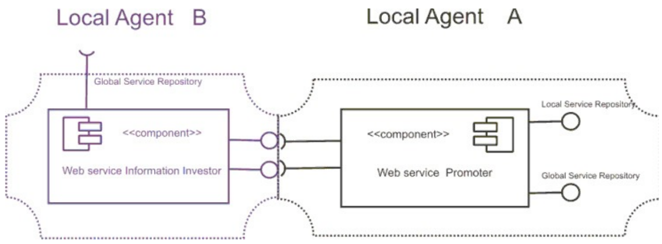


Figure 2. Web service Promoter connections

In fact two interfaces for this component should be provided by the Web Service Information Investor of the other agent. Figure 2 shows this connection between Local agent A and Local agent B. When any changes happen for some records of the local or the Global Service Repository, they use the Web service Promoter's interfaces. The Web service Promoter in turn can use the interfaces provided by the Web Service Information Investor in Local agent B, and the Web Service Information Investor at agent B can update its Global Service Repository if needed (because in some cases it could be done already). As a result, the Global Service Repository of agent B will use the same interfaces for the Web service Promoter at agent B and this will be done for any connected agent to Local agent B. In this way, any changes on connected agents can be updated quickly.

F. *Local Coordinator*

The kernel of the local agent is the Local Coordinator. Other components provide information for a Local Coordinator (on the local machine or even for a remote agent). The Local Coordinator facilitates our transaction model to be applied for complicated business activities (long-running transactions) as well as simple transactions.

Generally the Local Coordinator requires an interface from the Local Service Repository for gathering the information about local web services which enables it to provide the preparation and commit phase in a two phases commit (2PC)

protocol. This normally can be handled by a transaction content in response to a transaction request (Script).

For communicating with another agent (its Local Coordinator), the Local Coordinator as well as *providing* an interface, *requires* an interface from the remote agent too. The Local Coordinator also requires an interface from the Global Service Repository, especially when it acts as an Initiator of the transaction. This makes it possible to create the transaction script based on the knowledge of the other agents' web services. Ultimately, it requires the interface from its local web services to able to invoke them.

IV.Digital Business Ecosystem Network

Our previous work [3] has described a distributed model of multi-service long-running transactions which has been designed for open collaborations within a community of businesses (SMEs) in digital ecosystems. The transaction model provides the capability for efficient recovery management, in terms of preserving as much progress-to-date as possible (omitted results), and provision for alternative scenarios or paths of execution (forward recovery). Using local coordination not only avoids any violation of local autonomy but also provide a fully distributed model for the transactions to be executed.

We now look at the durability and reusability of the transaction and stability of it (as the dynamic nature of SMEs) as areas for improving the expected quality of a DE. It is important to avoid the abortion of the transaction even when some (or even all) participants are temporarily disconnected. This problem has been solved when one of the participants (at each nested part of the transaction) is/are disconnected. But we try to provide a highly reliable environment which can cope with dynamicity of SMEs and high probability for their disconnections.

It is very important to keep the result of a successful or unsuccessful transaction (even by considering regular unavailability of initiator and other participants). Another issue is lookup algorithm. Meanwhile we might provide even more knowledge about the unavailability of SMEs (or their services) based on their natural behaviour. This can be argued by the probability of fragmentation on such a network (ref: D3.1, D3.2). Our goal is not just providing a network structure for answering these requirements but also reusing fragmented network structures provided by the transactions to create a fully connected network.

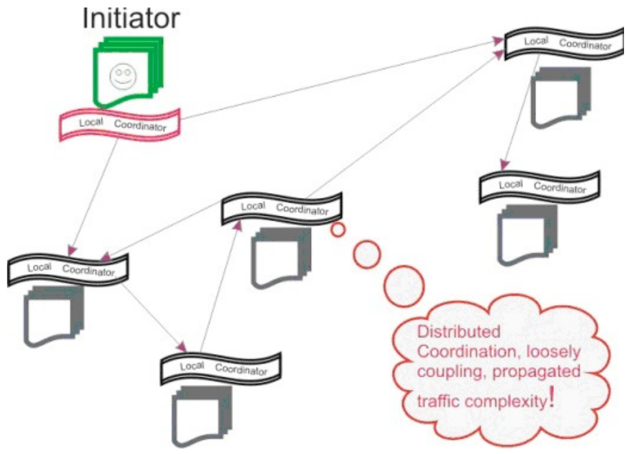


Figure 3. Distributed coordination for executing transaction

Figure 3, the actual execution of a transaction is in a fully distributed manner with each participant using its local coordinator. This creates a temporary network between service providers which will disappear after the transaction is finished. However, this network has some unique characteristics which are worth keeping for later use. This temporary business network inherits all of the transaction model properties (ref: D3.1, D3.2); in one hand, it is loosely-coupled which gives full local autonomy to platforms, and on the other hand it has resistance to failure and can be recovered - it can even handle short-term disconnections as the traffic is not focused on a centralised point. Meanwhile the platforms involved in a transaction are in a related domain and there is probability for them to do similar transactions again.

A. Virtual Private Transaction Neotworks

We call the temporary network created by a 'Transaction' a "Virtual Private Transaction Network" (VPTN), as apart from the participants in the transaction, naturally they are shared with other platforms and normally they are not created as an actual network. Because of the specific properties of these networks we consider a component for keeping them and adding the ability for re-using them.

Figure 4 shows a collection of these networks. Clearly, VPTNs are a collection of fragmented networks, apart from occasional overlaps between different transactions which is also transparent.

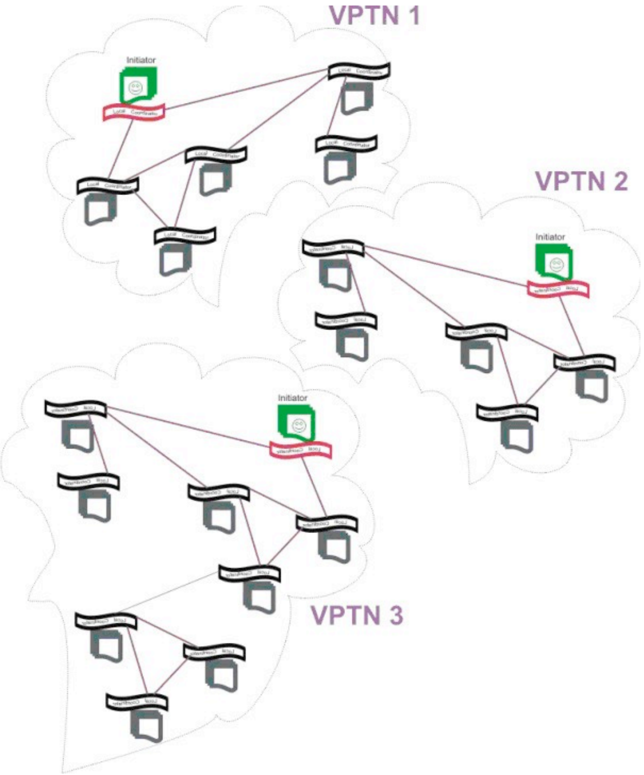


Figure 4. Fragmented VPTNs

The Global Service Repository's (introduce in III.B) main duty is to keep our virtualization permanent according to the life-time of each business agent. Therefore these VPTNs will be saved by it. In the aspect of implementation, the simplest option which we considered is using a database but we consider the adoption of this to other forms (such as some standard schema which may show the connection and relationships and reflect the semantic of the association in a standard way). An example is given in Figure 5; the Local Coordinators are connected to the Global Service Repository, they can update links and information about other participants and their services can be saved. In this way, VPTNs can be saved permanently.

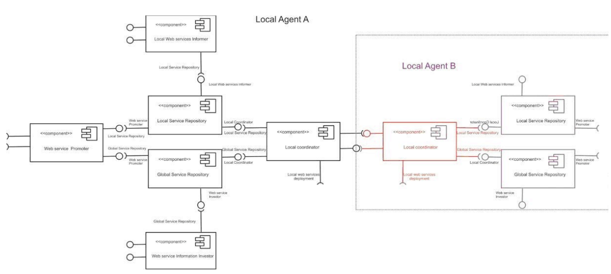


Figure 5. Global Service Repository connections

It is important to mention, keeping this information is not the only duty of 'Global Service Repository'. Actually the 'Global Service Repository, by using a 'Web Service Information Investor', tries to update its information about other platforms (web services) based on the architectural requirements (which may be clearer by the end of this paper) and at the same time, by using 'Web service Promoter', tries to transfer its records to the other platforms for creating a more stable network (which will be explain in the rest of this chapter). Figure 5 can explain these connections between 'Global Service Repository' and 'Web service Promoters' and 'Web Service Information Investor'. (there are more here in connection with semantic web, which will be added).

B. A measurement for Platform stability

In order to increasing the life time of each VPTN and avoid business transaction abortion during temporary disconnection of some participant in a VPTN, we need to increase the stability of VPTNs over time. We try to find some measurement of stability for each node in the network. It turns out that the probability for availability of each node is the important factor. For this reason, in this section we try to analyse this factor by using service availability and later on, we can use this to ensure better stability in the network. Our purpose is to find a good candidate in each VPTN for keeping information about the private network in its repository and in this way have maximum stability of the network in time (for example, it may be able to keep this information for some hours). Then in the next part we try to improve this network towards a fully permanent network.

It would be unreasonable (and not feasible) to expect nodes to be online all the time and thus stability is determined on the basis of declared availability.

For finding a more precise and computable measurement for node stability, first we introduce an important property for each node, called Expected Availability Time (EAT). This is the time when the node is expected to be available and online in the network (Figure 6 shows an example of EAT for a node in the network). The node stability is then calculated as the actual availability of the node against this expected time. These are typically different, since during its EAT the node may be disconnected. These disconnections will reduce stability (reliability) of the corresponding node in the final selection. This notion of stability can be simply calculated as below:

$$NodeStability = \frac{EAT - DisconnetionPeriods}{EAT}$$

It can be seen that the closer NodeStability gets to 1 for a node, the more stable the node is (which can be understood as more reliable or predictable). For calculating the stability function of a node, in the first instance we use its participants in a transaction (other nodes in the same VPTN) to check its availability behaviour. At the moment, we have considered EAT as a part of service provider's (SMEs) business model which is provided by each of them when they join to the network (and they may change it, if the nature of their business changes). It should be noted that other approaches can be considered for calculating the EAT - for example, it is possible to use an algorithm based on the network neighbourhoods for calculating EAT which would allow it to vary over time.

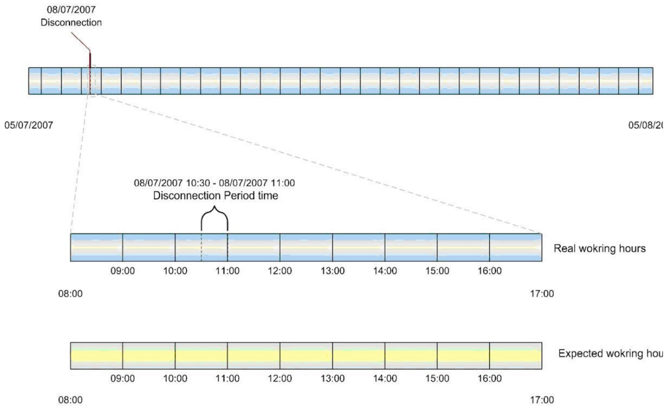


Figure 6. Expected Availability Time

By finding the most stable node in each VPTN, we can rely on that node for keeping uncommitted transaction information or even saving the commit transaction's results for providing better global durability and reusability. But still one may ask, what will happen if all nodes in a VPTN (including the most stable one) are disconnected together?

C. Virtual Service Network

Apart of the unanswered question about a total disconnection in a VPTN, the virtualization in the transactional level provides us with a fragmented network but still does not fulfil the semantic web's link feasibility from any node to the other. For answering this requirement, we provide another virtual level, which offers a connected network between services (service providers). Figure 7 shows our different level of virtualizations.

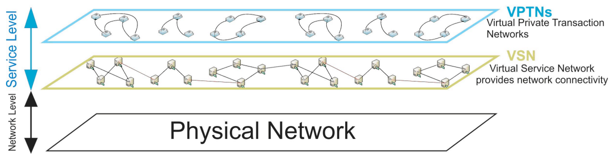


Figure 7. Virtualization in Digital Business Ecosystems

An important question will be: how can this network be created without external controls and how can software agents manage it based on the dynamic character of a Digital Business Ecosystem?

V. Increasing connectivity in a dynamic environement

Clearly the simplest way for provide a fully connected network is by connecting VPTNs together. The best candidates for connecting VPTNs together are the most stable nodes in each VPTN. Figure 8 shows a demonstration of this connected network. In this paper, we mostly focus on the result of such a network (which shows the characteristics and necessities of it) and do not focus on the birth model.

By connecting VPTNs, we mean the 'Global Service Repository' of each candidate in each VPTN will be connected to the candidate of the other VPTN. In this way we can improve the stability of the connected network (the maximum time for the network to be alive). However, we cannot warranty full stability of the network and still cannot avoid the occasional fragmentation (because even in the best case, it is dependent on each platform's availability and if the total online time of all stable nodes cannot cover 24 hours, our network will collapse for some period of time, precisely that in which all of them are not available).

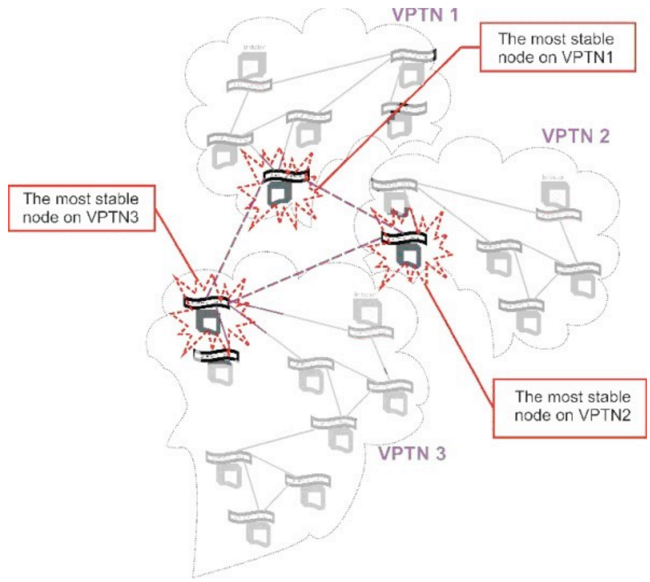


Figure 8. Connecting VPTNs

A. Super peers and permanent nodes

The conventional solution to the problem of fragmentation, which has been used by several P2P networks ([9], [10]), consists of introducing an extra layer to the network; the so-called super peers. Actually the super peers are decentralised servers, which are intended to provide reasonable connectivity and avoid the fragmentation in the network. Depending on the size of the network, the protocol used and the number of super peers, each super peer manages a number of nodes and can check their availability. At the same time, each super peer provides a strong link to the other super peers and in this way the design ensures that there is low probability for fragmentation.

The primary necessity for having super peers is providing stable nodes which are online all of the time. This means super peers are expensive nodes with costly maintenance requirements. It should also be noted that their resources are used for facilitating network operation management tasks. When considering such a solution for a digital ecosystem environment involving SMEs, the question arises as to who is going to provide such nodes? Apart from feasibility issues, most SME business models militate against this.

Additionally, during peak time the pressure of high traffic can result in a bottleneck on super peer nodes and because of the connectivity role of super peers, the whole VSNs and consequently VPTNs (and the corresponding business activities of SMEs) will come under serious risk. It could be argued that the problem may be countered by providing maximum facilities and additional resources on super peers, but this may address the problem only temporarily. Powerful

super peers will still need to be online and monitor the whole network at all times, processing redundant data and producing overheads waste at off-peak times of the network while they will be continuously under pressure at peak time while the network grows (more nodes join).

Moreover, and even if it were possible to find suitable SMEs willing to provide permanent nodes as super peers, these may change their business model and after some time may not find it useful to provide a permanent (and expensive) node anymore. It is not advisable and may not even be possible to force small-to-medium enterprises to be constrained into a static business model and stable behaviour for the sake of stability of the DE infrastructure.

Perhaps even more importantly, the super peers solution results in a static topology for the network as these nodes are pre-selected and their role is pre-determined in the network. This is by no means satisfactory in a highly dynamic environment of a digital ecosystem where the idea is that the network topology changes continuously to adapt to its very usage and demands of the participating entities. The evolving nature of the DE is intended to reflect the congestion of network packages and nodes that change from time to time.

It transpires that dealing with change and adapting to ever changing requirements is unavoidable in the content of a digital business ecosystem. This leads to thinking about a design solution that provides a dynamic topology that continuously evolves to echo changes in the participating entities or nodes. Our approach to the business network design is based on clusters of nodes for providing permanent clusters, rather than permanent nodes as is the case with super peers, and is described in the following sections.

B. Permanent Cluster

As mentioned before, in contrast with conventional super peers, we try in our network design to move towards a more dynamic architecture which does not rely on just a few permanent nodes. Central to our approach is finding permanent clusters on the network. More specifically, we are identifying aggregations of stable nodes, where node stability is determined as in the previous section. For doing so, the most stable nodes from different time zones must be chosen, in a way that they cover 24 hours. In fact, we are trying to find permanent clusters through the most stable nodes.

The important part in determining permanent clusters is discovering different aggregations of these time zones which can cover 24 hour availability. Any union of the stable nodes in the aggregations (which provides 24 hour availability) are actual permanent clusters. Figure 9, shows the simple situation in which the most stable nodes have been selected from two sets of time zones which can cover 24 hour service availability to form permanent clusters.

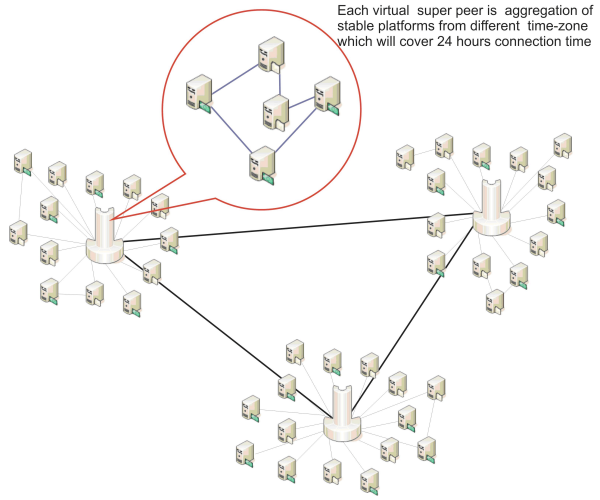


Figure 9. Permanent Clusters and Virtual Super Peers on DE

C. Virtual Super Peers

By using stable nodes from permanent clusters, as is shown in Figure 9, we can create Virtual Super Peers (VSPs) which are effectively permanent clusters of nodes in the network. These can provide the desired stability for the network. The strong connection between the virtual super peers themselves on the one hand and the connection between them and their nodes on the other, decreases the probability for fragmentation. Depending on the level of reliability required for the network, it is possible to include further redundant stable platforms from each available time zone. For example, in Figure 9 we have included two stable nodes from one time-zone and three stable nodes from the other one (the green and creamy signs show different time zones).

In this manner, the good connectivity can cause more reliable transactions at the VPTNs level. Meanwhile the traffic is spread over the virtual super peers and there is less risk of bottlenecks at peak times. Nodes within a virtual super peer need to keep information only about nodes in their cluster and about neighbouring VSPs so at off-peak time the amount of redundant information processing is reduced dramatically as compared to the classical super peers solution.

Since choosing stable nodes is a dynamic process (it is done based on the stability function, EAT to Disconnection period of a node during EAT, whose value varies over time) the virtual super peers are also formed dynamically. This means the topology can change from time to time and new nodes can be added to the permanent clusters as the structure of virtual super peers changes. A node can become part of a virtual super peer, when its node stability increases and overcomes some threshold. In contrast, nodes that are super peers may not be able to cope with the increased number of connections they get, and possibly increased number of transactions they perform, and so lose their virtual peer status. Within a digital ecosystem for business, SMEs would be expected to invest at that time (in hardware, processing power, band-width etc.) and become again part of a virtual super peer in future. It is in this sense that the topology evolves to reflect the usage and demands of the participants who benefit from

and contribute to the ‘sustainability’ of the network.

Additionally, network congestion can change the maximum level of node stability (Section…) which in turn affects the selection of the most stable nodes in forming the permanent clusters. High congestion of packages can increase or decrease network reliability (higher traffic on few virtual super peers can potentially create a bottleneck and even cause fragmentation). In a digital business ecosystem, the best part of the traffic is the result of business activities which are effectively long-lived transactions. These have been virtualised in VPTNs and therefore, using the effect of VPTNs for making VSPs and their client nodes, can increase the stability of each virtual super peer.

Furthermore, we expect a reasonable cluster coefficient on the account of having VPTNs as the main building blocks, which we have seen are formed from a transaction. This means its nodes are in relevant domains – by connecting them to several VSPs we actually increase the probability for that. We also expect a fair distribution degree on the account of propagating links to VSPs. This means that instead of being concerned with individual links for each node, aggregate links of VSPs come into play.

Finally, reusing business activity results (or service-on-fly as result of composite services [11]) and explorative service composition [12] are other factors which can be considered for higher performance within a digital business ecosystem and can provide potential for creating so-called virtual vendors.

D. Parametric algorithm for choosing VSPs

In the first step, the most stable node in each VPTN (participants of a transaction) should be selected for keeping vital information about the transaction and its participants. In this sense, the network provides a level of durability without any extra cost from participants and it covers omitted results, a problem relating to preserving as much progress-to-date as possible in the event of aborting a transaction. An extended lock mechanism for recovery management in long-lived business transactions has been described in [13] which contains further details on omitted results as well as other aspects of compensation in long-running transactions.

The best candidates for connecting VPTNs together are the most stable nodes in each VPTN. Figure 9 shows the internal structure of each VPTN and the connection between VPTNs. The internal structure of VPTN contains a lot of information from the transaction level such as log structures, lock schemes for ensuring consistency in recovery mentioned above, local coordinator design, formal analysis of the required interactions and compensations, along with alternative scenarios for forward recovery.

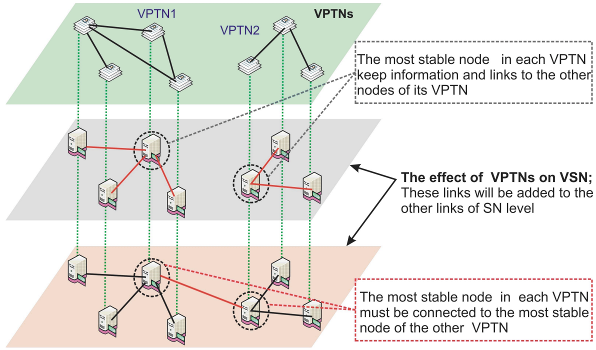


Figure 9. Optimisation from VPTNs to VSN

The direct effect of connecting VPTNs together is to raise the cluster coefficient of the network. Conversely, connecting the most stable nodes of VPTNs together provides the opportunity of choosing the best candidate locally between these stable nodes for the permanent cluster. Choosing nodes of the permanent cluster in this way, results in a virtual super peer that provides fair traffic distribution at the VSN level (each virtual super peer will take care of its local VPTNs). The main concept behind forming permanent clusters stays the same, i.e., selecting the most stable nodes from different time zones which can cover 24 hours online time.

VI. The Network in Practice

The important question is about feasibility of applying basics of the semantic by using these virtualizations. On the other hand, we want to show, it is possible for any agent to not only deploys transactions in a collaborative way, but also traverse the virtual network without any fragmentations. For doing so, our transaction schema is a heterogeneous model [ref: D3.1, D3.2, ieee paper] which is able to work with different standard and protocol. On the other hand, we considered a general structure for our local agent [D3.2 and recall III] to provide ability for working with XML, XML Schema, RDF, RDF Schema and OWL.

However, the most important question will be about the practical stability of the network. As the most stable node in each VPTN is the best candidate for keeping the transaction information, the corresponding business activities will have increased levels of reliability. The fact VPTNs are used initially in the design of the business network, and are connected through their most stable nodes which are determined dynamically, allows in most cases the candidate platform to avoid the full rollback or compensation of the transaction when some participants of the long-lived transaction get disconnected within its du-ration. This can be considered directly in the design of the recovery mechanism for such a transaction model, as done for example in [13] for the distributed transaction model de-scribed in [3]. Another expectation of the network design we have proposed is that the dynamic topology resulting from the selection of virtual super peers, which relies on the stability measurement of each individual platform in each VPTN, reduces the probability of fragmentation. Certain evolutionary models studied in biology exhibit some characteristics of this network design. Meanwhile some practical simulations can compare the theoretical

behaviour and practical status of the network in different situations. These aspects are discussed in further detail next.

A. The Network in Practice

We have seen that we are dealing with a highly dynamic environment where there is no central point of control and a high probability for failure (in a transaction or the network itself). In the design of a business network for this environment we have considered a dynamic, ever changing topology. It would however be desirable to be able to somehow guide the way this topology evolves. Considering the requirements of DE for business, we propose to draw upon the evolutionary growth of metabolic (signal transduction) networks, as studied in the work of Rzhetsky and Gomez (e.g. [14], [15]) in designing the birth and growth model for an autopoietic P2P network to sup-port long-running business transactions in the OPAALS project (see [1], pp. 77-94). It turns out that the evolutionary growth in molecular networks exhibits scale-free characteristics while it also has some interesting properties with respect to network connectivity. More specifically, the frequency of vertices connected to exactly k other vertices in metabolic (signal transduction) networks follows a power-law distribution. The distribution function degree is equivalent to:

$$P(k) = c.k^{-\gamma}$$

where c is a normalising constant and diverges across networks (but usually has a value between 1-3; in our simulation was 2.34) and the network follows the classical Barabasi-Albert model [16]. This growth model says that the total number of network vertices is more than three times the number of nodes, which shows the connectivity (even with-out using VSPs) to be quite good. But there were specific weaknesses which do not seem to be solved without introducing the VSPs conceptual model.

B. Achilles heel of the network

The basic network as a scale-free network follows the power law distribution, which means most nodes will have a few links and a few nodes will have a large number of links (see Figure 10). This may result in a high dependency on a few number of nodes which have a large number of links. Such nodes actually play the role of hubs in a typical scale-free network. Thus, the network would already be vulnerable since any smart attack on hubs (or even a series of accidental failures), may cause fragmentation on the network (creating islands in the network), As a consequence, any running business transactions will be discontinued and a fragmented network can be extremely costly to repair (de-fragmentation) as discussed before.

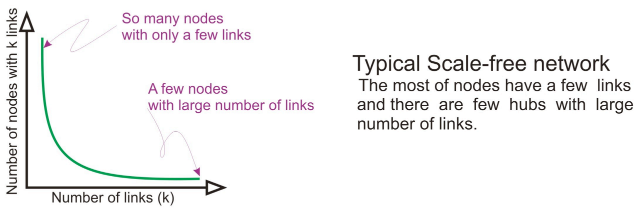


Figure 10. Power Law Distribution

Another problem has to do with the inconsistency of such a model with the dynamicity of a digital ecosystem and the versatility of SMEs business models. Hoping to have stable and permanent hubs to warranty the stability of a few individual nodes is in contrast with the very nature of an ecosystem. Also, SMEs may not provide stable and permanent nodes for hubs at all. Therefore, there is possibility for fragmentation even without any external attack or physical failures. As a result of this, the network may suffer regular transition periods between exhibiting the characteristics of a scale-free network and those of a fully random network (with potential islands).

C. Agent-based DE network's experience

Our experimentations and results show the proposed model to use the advantages of a conventional scale-free network, but at the same time has built-in capability for coping with events which typical scale-free networks are vulnerable against. This is depicted in Figure 11 (our network is DBE/OPAALS scale-free network).

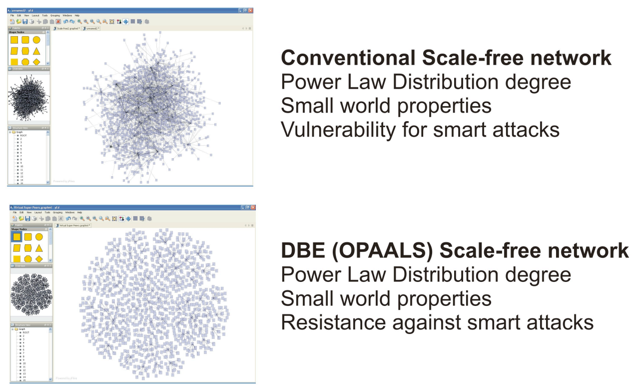


Figure 11 Digital Business Ecosystem network in comparison with a conventional network

The major effect of our proposed model is that the actual hubs are virtual super peers. As mentioned in Section V, VSPs are aggregations of several nodes that create a permanent cluster. The dynamic nature of VSPs makes them adaptable to changes in business models or more generally the versatile nature of SMEs. In addition, the fact 'platform stability' varies over time allows for the formation in virtual super peers to also change (if the corresponding platform stability changes).

VII. Conclusion and further work

We have provided some level of virtualization which offers the basic characteristics for semantic web. On one hand, distributed transactions have been modeled in a virtual private transaction network (which supply the semantic links between business partners in a business activity at the same time), on the other hand, virtual service network, as a connected network, provide any software agents virtually to have a link to any other service provider. We have introduced this network as a P2P network, based on the notion of virtual super peers that equips the network with ability for resisting against different types of failure, which support our semantic map of a

connected network between services which supports transactions and it is readable by computers as well as human.

In contrast with a conventional scale-free network, attacking a few high degree nodes may not destabilize the whole network. This is because in the first step each virtual super peer is made up from several different platforms, from different domains even. At the same time, the dynamic nature of ‘permanent clusters’ allows for each participant in VSPs to be easily substituted by another stable platform (recall the discussion in Section V).

In addition, platform failures or traffic bottlenecks may not fragment the network or lead it to a transition state between the random graph and its original topology. Actually, even if a node of a permanent cluster fails or experiences high traffic, this will only cause a substitution of the node with another stable node in the closest level. Our actual simulations show the network to follow a fractal model around virtual super peers which can vary depending on the size of the permanent clusters.

The resulting characteristics of this network foster an environment that potentially enhances the ability of SMEs to compete by means of virtual vendors. Smart composition of services and reuse of profitable (and successful) transaction results can lead to the creation of virtual retailers in place of large enterprise vendors that wish to dominate the infrastructure and rip any potential benefits. Furthermore, the stable network in a collaborative environment can be used in the continuous creation and sharing of knowledge in the form of business models.

A. Web 2.0

This approach has been inspired by web 2.0 and used its presumptions and views; we assumed the web as a platform, in our model Network effects created by an architecture of participation [see 17], we considered the business processes are modeled by transactions and business models are enabled by content and service syndication, we assumed Software above the level of a single device by the mean of leveraging the power of the "Long Tail" (Businesses with distribution power) [see 18,19].

Web 2.0 also has been described as an environment where users generate and distribute content, often with freedom to share and re-use. One perceived result is a rise in the economic value of the Web and it has been one of our primary aims in designing distributed transaction and providing the reusability for participants in the transactional virtualization (VPTN). But this was the start of our approach and the current experiences and results are inline to moving forward towards next step which it is 'Semantic Web'.

B. Future of the web and roadmap for our approach

Already discussion about the next generation of Web (Web 3.0) has been started:

“*Web 3.0, a phrase coined by John Markoff of the New York Times in 2006, refers to a supposed third generation of Internet-based services that collectively comprise what might be called 'the intelligent Web' — such as those using semantic web, microformats, natural language search, data-mining, machine learning, recommendation agents, and artificial intelligence technologies — which emphasize machine-facilitated understanding of information in order to provide a more productive and intuitive user experience.*” [20], [21]

Apart of our roadmap towards ‘Semantic web’, we have started to figure out other expectations of the new web generation in our work; on one hand our approach shows the possible convergence of Service-oriented architecture and the Semantic web (as it is discussed with details in [22]), on the hand, our research is trying to conduct developing software for reasoning, based on description logic (and provide the feasibility for intelligent agents in the future) [23]. Meanwhile our approach has started the necessary design and emergence of different technologies [20] such as Web services interoperability in distributed P2P solution and at the same time considering not only architectural aspects of the Semantic web but also a primary attempt to offer the full ability for processing Semantic Web technologies in different levels of virtualization. We are trying to complete this merge and move towards deeper aspects of the Semantic web.

References

[1] A. Razavi, S. Moschoyiannis and P. J. Krause "Preliminary Architecture for Autopoietic P2P Network focusing on Hierarchical Super-Peers, Birth and Growth Models." OPAALS project Deliverable D3.2, 2007 - available at: http://files.opaals.org/OPAALS

[2] Z.B. Daho, Simoni, N. "Towards Dynamic Virtual Private Service Networks: Design and Self-Management", 10th IEEE/IFIP Network Operations and Management Symposium, 2006. NOMS 2006.

[3] A. Razavi, S. Moschoyiannis and P. Krause. "A Coordination Model for Distributed Transactions in Digital Business Ecosystems." In Proc. of IEEE Int'l Conf. on Digital Ecosystems and Technologies (IEEE-DEST 2007), IEEE Computer Society, 2007.

[4] A. Razavi, P. J .Krause and S. K. Moschoyiannis. DBE Project Deliverable D24.5: DBE Distributed Transaction Model, 2006.

[5] B. Martini, F. Baroncelli, P. Castoldi, “A novel service oriented framework for automatically switched transport Network”, Sympo-sium on Integrated Network Management, 2005. IM 2005. 2005 9th IFIP/IEEE International. Pub: 15-19 May 2005, pp(s): 295 - 308.

[6] Kristiansen L. et al. "TINA Service Architecture and Specifications", TINA 1.0 DELIVERABLES AND SPECIFICATIONS (available at: http://www.tinac.com/specifications/specifications.htm)

[7] Sahai A. et al. “Automated SLA monitoring for web services”, Proceedings of the 13th IFIP/IEEE International Workshop on Distrib-uted Systems: Operations and Management: Management Technolo-gies for E-Commerce and E-Business Applications, p.28-41, October 21-23, 2002.

[8] Z. Li, P.Mohapatra, “QRON: QoS-Aware Routing in Overlay Networks”, Service Overlay Networks in the IEEE Journal on Selected Areas in Communications (2004).

[9] B. Y. Beverly, H. Garcia-Molina, " Designing a super-peer network", Proceedings. 19th International Conference on Data Engineering, 2003. Pub: 5-8 March 2003, pp: 49-60

[10] B. Yang, H. Garcia-Molina, "Improving search in peer-to-peer networks", Proceedings. 22nd International Distributed Computing Systems, 2002. pp: 5- 14

[11] Yang, Jian; Papazoglou, M P. and Heuvel, W-J van den (2002), “Tackling the Challenges of Service Composition in E-Marketplaces”, Twelfth International Workshop on Research Issues in Data Engineering: Engineering E-Commerce/E-Business Systems, 2002. RIDE-2EC 2002 (IEEE Computer society), pp:125-133

[12] Papazoglou M. P., Traverso P., Dustdar S., Leymann F. and Kramer B.

J. Service-Oriented Computing Research Roadmap. In Dagstuhl Seminar Proceedings 05462, Service-Oriented Computing (SOC), pp. 1-29, 2006.

[13] A. Razavi, S. Moschoyiannis and P. Krause. Concurrency Control and Recovery Management for Open e-Business Transactions. In Proc. of Communicating Process Architectures (CPA 2007), 2007.

[14] A. Rzhetsky, S. Gomez, "Birth of scale-free molecular networks and the number of distinct DNA and protein domains per genome", Bio-informatics, 17(10):988-996, 2001.

[15] G. Karev, Y. Wolf, A. Rzhetsky, F. Berezovskaya, E. Koonin "Birth and death of protein domains: A simple model of evolution explains power law behavior", BMC Evolutionary Biology, 2:18, 2002

[16] A. L Barabasi, R. Albert "Emergence of Scaling in Random Net-works". Science 286(5439): 509–512, 1999

[17] Battelle J. and O'Reilly, T. "The State of the Internet Industry", First Web 2.0 Conference, San Francisco, USA, 2007-11-08.

[18] Anderson, C., "The Long Tail", Wired Magazine ,Issue 12.10 - October 2004.

[19] Anderson, C., The Long Tail: Why the Future of Business is Selling Less of More, Hyperion (July 11, 2006)

[20] Spivack, N. "THE THIRD GENERATION WEB IS COMING", LIFEBOAT FOUNDATION SPECIAL REPORT (last access 10/11/07, available at: http://lifeboat.com/ex/web.3.0).

[21] Markoff J., "Entrepreneurs See a Web Guided by Common Sense" , New York Times, November 12, 2006.

[22] Provoost, L. and Bornier, E. "Service-Oriented Architecture and the Semantic Web: A killer combination?", University of Utrecht, February 10, 2006 (avalaible at: http://lee.webcoder.be/papers/sesa.pdf)

[23] Wainewright, P. "What to expect from Web 3.0", ZDNet, November 29, 2005 (last access: 10/11/07 http://blogs.zdnet.com/SAAS/?p=68)

[24] Berners-Lee, Tim and Fischetti, Mark, Weaving the Web. Harper San Francisco, (1999). chapter 12. ISBN 9780062515872

Self-organising map based tag clouds

Creating spatially meaningful representations of tagging data

[OPAALS Conference 2007]

Jaakko Salonen
Hypermedia Laboratory
Tampere University of Technology
Tampere, Finland
jaakko.t.salonen@tut.fi

Abstract—In this article, we present a method and a tool for visualising tagging data with self-organising map (SOM). Tagging as a knowledge management method is described in relation to existing tools and methods. Current approaches of tagging data visualisation are also presented, especially popular "tag cloud" method and its different variations. Finally, our SOM based visualisation method, SOM Cloud, and a proof-of-concept implementation of it is put to test with data from del.icio.us social bookmarking service. As a result of the study, we found out that while the applicability of SOM to "tag cloud" metaphor is limited, we could use it successfully to add spatial encoding to tagging data.

tagging; folksonomies; information retrieval; information visualisation; self-organising maps; knowledge visualisation

I. Introduction

From the new, "web 2.0" breed of online services, a new approach for creating metadta, called tagging, has emerged. Especially in social bookmarking services, users may add tags – i.e. arbitrary keywords associated with resources – to describe and organise resources. Tagging features have also spread to other types of applications as well. Even offline applications such as music players and photo organisers have started supporting them.

Tagging is technically a trivial feature, but it is one of those rare features that give users the ability to create their own structures for knowledge organisation. Tagging data that creates associations between tags and different resources, can be used to various personal navigation structures, including navigating resources by tags, navigating tags by related tags and navigating related resources by their tags.

Motivation to our study of tagging systems lie on research objectives of the European Network of Excellence Project OPAALS. In OPAALS, an overall objective is to build sustainable interdisciplinary research community and develop integrated theoretical foundation for digital ecosystems' research. This objective is approached by building an Open Knowledge Space (OKS) for knowledge creation and sharing.

One of the challenges in building the OKS, is the integration of knowledge from different practise communities. While various scientific domains may have very well defined, formal domain ontologies that accurately specify conceptualisations, they offer little help for leveraging cross-domain understanding of concepts and their relationships, as these ontologies do not capture the entire process, but mainly the results of the vocabulary construction work.

As a side effect of a well-planned tagging system, tagging contributes to development of a folksonomy. A folksonomy is a vocabulary of tags emerging from community of users, not a vocabulary defined by a single user nor by an outside party. We see that in this area, tagging and the resulting folksonomies could be used to support this process of leveraging cross-domain knowledge. As users are given the freedom to create arbitrary tags, it is suspected that more of the otherwise hidden aspects of the vocabulary construction could be tracked.

Tagging, however, has several challenges that need to be addressed. Vocabularies based on tagging data 1) may evolve rapidly, 2) may be incomplete or inconsistent, 3) may not have clear, unambiguous interpretations for specific tags and 4) may not be readily organised into tight hierarchical or taxonomic structures. As such, it is not all obvious, what is the most effective way to display tagging based data. Fixed presentations typical to most knowledge organisation systems (like hierarchies and taxonomies) may not be readily used. Associative navigations based on binary relationships between tags and resources, on the other hand, can be used, but do not scale well to large numbers of tags and resources.

In our approach, we use a combination of data mining and information visualisation to generate more flexible presentations of the tagging data. We see that the freedom in visualisation for choosing the visual properties (shape, colour and position) of the object, offer an opportunity for recreating new and more efficient abstractions. Ideally, this would mean that not only current relationships could be better understood, but also discovery of hidden properties of the data could be enabled.

Our study to tagging data visualisation is organised to in this paper as follows: section 2 defines basic concepts used throughout the rest of the work. In section 3, we explore current approaches used in tagging systems for information retrieval and visualisation. In section 4 we present our approach to tagging data visualisation, based on an implementation of self-organising maps algorithm and a visualisation client in Java. Section 5 concludes and discusses our work.

II. Tagging and folksonomies

Before beginning, let us clarify some of the more generic terminology we use. By *objects* we refer to any individual abstract information entities. *Resources* are digital information objects such as web pages, photos or video clips. In our definition, that has an identity can be tagged is called a resource. Terms resource and object are used throughout this paper interchangeably as differences are somewhat subtle.

Tagging is defined as the process of attaching tags to resources. In the process of tagging, user selects one or several tags. The user performing the tagging is *called the tagger*. *A tag* is a user-defined string, usually a single keyword that is associated to resources in the act of tagging. Note that, in comparison to traditional keyword metadata, tags are not chosen from controlled, third party defined. While old tags may be re-used, tagger may create new ones on the fly. Recommendations based on both user's own and other community members' tagging habits should be made available To support the reuse of existing tags.

We see that the usual motivation of tagging is personal information retrieval. Examples include tagging posts in weblogs for categorisation, tagging songs or videos for playlist generation and tagging pictures for tag-based navigation. This kind of tagging practise much resembles the use of directories for organising file-based resources, with the distinction that a resource may have several all no tags at all, regardless of physical location.

It would make sense that tagging schemes of personal information retrieval would consist of tens, rarely hundreds of unique tags. However, as social bookmarking tools have demonstrated (fc. [1]), is higher number of unique tags surprisingly common. As such, there is a clear difference between the ways how people use tags in the contexts.

The difference on the way of using labels in social software tools from traditional software tools has spurred a lot of discussion. To distinct the traditional use of labelling from socially influenced labelling, Thomas Vander Wal has coined the concept of folksonomy, defined as ”the result of personal free tagging of information and objects (anything with a URL[1]) for one's own retrieval” [2]. The concept is underlain by the idea that tagging does not always lead to creation of a folksonomy. For folksonomy creation, it is required that tagging is 1) personal, motivated for one's own information retrieval 2) done in a social environment and 3) done by the person consuming the information [2]. As according to Vader Wal's point of view, we see that tags in a folksonomy should meet with the following three criteria:

1) *Tags should be personal*. Users may or may not share same keywords for the same resources. Folksonomic tags are – in fact - cumulative, resulting in social indexing of resources in which everyone gets a vote.
2) *Tagging habits should have an influence on the outcome.* The tagging habits of the tagger himself and other community members should have an influence on the outcome tagging.
3) *Tags should not be added automatically.* An implicit assumption in Vander Wal's definition is that the person consuming the information knowingly adds it to his or her personal collection of resources for later reuse.

To distinct different ways of applying tags in social environment, Vander Wal has made a distinction between broad and narrow folksonomies. *Narrow folksonomy* ”provides benefit in tagging objects that are not easily searchable or have no other means of using text to describe or find the object”, very much resembling the way how media organising software tools employ tagging. *Broad folksonomy*, on the other hand, incorporates many people tagging same objects while everyone may choose to apply their own tags. This is also the situation in social bookmarking services, where most of the tagged bookmarks are public and shared. [3].

Let us next consider the formalisation of tagging. While the intuitive interpretation of a tag is a two-place relation between resources[2] and tags, tags are better understood of votes from corresponding taggers. Tom Gruber has formalised tagging as a three-place relation [4]:

```
Tagging(object, tag, tagger)
```

While this relation separates use of tags by different taggers, it still does not commit on the *in situ* nature of tagging. Gruber sees that by adding the source of the tagging data as a fourth property to the relation, may help to understand in which social context the tagging was done [4]. To put it formally, we can define that:

```
Tagging(object, tag, tagger, source)
```

Gruber remarks that while source is easily interpreted as the community in which the act of tagging was done, it may be understood more generically as an explicit notion of source in scope of namespaces or “universe of quantification” for these objects. Whatever the interpretation, it is important that some formalisation of the tagging context is available: tagging objects in a photography service has distinct implications from tagging objects in a scholar reference service [5].

III. Visualising tagging data and folksonomies

A. Current approaches

Perhaps the most common way for creating visually oriented representations of tagging data is the use of tags as search facet. In this way, users may browse resources according to their associated tags. We split these representations into two types: personal and shared tag indices.

Personal tag index consists of only tagging data entered by one person. These indices are seen to contain usually tens, rarely more than hundreds of different tags. For this reason, it is easy to represent personal tag index as an alphabetically ordered list or as another ordered structure.

Shared tag index consists of an aggregation of tagging

1 Universal Resource Locator

2 Or objects, as according to Gruber

data, entered by several users. Aggregations of tags may be used to explore any resources that anyone has tagged with corresponding keywords. In broad folksonomies, indices may be contextual: by defining context as set of users, a contextual set of folksonomic tags may be displayed.

An index of tags may, however, grow larger than what can be feasibly displayed with simple lists. A commonly used approach for dealing with the increasing amount of information, is the use of different kinds of tag clouds. *A tag cloud* is a two dimensional presentation of tags that makes it easier to perceive a large number of tags. Usually in tag clouds, tags are visually weighted according to their use frequency.

To succeed in creation of visualisations, we see that it is especially important to create abstractions that possess meaningful visual interpretation for users. This is not always the case in tag clouds, as figure 1 demonstrates: tags 'java' and 'howto' appear spatial nearby, but the close positioning is more likely based on alphabetical than on semantic proximity.

The lack of meaningful spatial interpretations in tag clouds has already been address for instance by Hassan-Montero and Herrero-Solana. Their solution is based on use of algorithm that organises similar tags close to each other [5]. An alternative solution, as proposed by Bassett is to implement interactivity to tag clouds. In such tag clouds (or "focus clouds"), focusing over a tag will highlight similar tags from the rest of the cloud to support the understanding of associations between different tags [6].

B. *Self-organising maps in tagging data visualisation*

Our approach to visualising tagging data is based on use of self-organising maps (SOMs). SOM (also known as Kohonen networks) is discussed in literature very extensively, and therefore interested readers are encouraged to take a look at the available literature (see especially [8], [9], [10]).

A self-organising map consists of an arbitrary number of neurons. *A neuron* is m dimensional real number vector. These neurons are associated to higher or equal dimensional model vectors, resulting in bijective, one-to-one mapping from neurons to models. In practise this means that map nodes have both [geometric] target space positions and model vectors for source space alignment.

The bearing idea behind SOM is that it can be used to create easily perceivable visual presentations of high-dimensional data sets. Thus, self-organising maps can be used to convert the high dimensions of statistical relationships between tags and resources (or tags and tags) into simple geometric relationships that can be represented efficiently as low, usually two, dimensional maps. (cf. [8])

As an algorithm, SOM is a form of unsupervised learning, based on the neural networks. The algorithm can be roughly split into two phases: 1) Training of the neural network in which initial input data is used to span the map and 2) The use of the trained map either for drawing or query of the map.

SOM is commonly applied for creation of similarity maps from given multidimensional input data [8]. Neurons are associated with selected points of the target space, so that the region is most sensitive to input vectors near that area. (see figure 1). In this way, while the mapping itself is discrete, any input vector can be mapped with SOM.

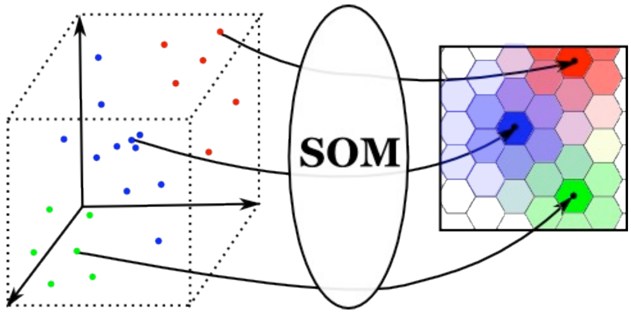


Figure 1: A schematic view of SOM algorithm: neural network is used to assign input vectors to locations in target space

Before the actual learning algorithm takes space, the neurons and their model vectors must be formatted with initial values. For model vectors, these initial values can be randomly selected or algorithmically derived from the input data values.

IV. Experimenting with del.icio.us

A. *Method and data description*

The idea behind our del.icio.us experiment was to compare our SOM based visualisation method with the del.icio.us's integrated tag clouds. Our SOM visualisations were generated using SOM Cloud application: a proof-of-concept Java application created especially for tagging data visualisation. Taking input from an XML[3] file, the application can be used to create self-organising map from virtually any set of tagged resources.

In order to span the self-organising map, a set of n dimensional input vectors was required. In our approach, we simply reserved one input vector component per unique tag. Accordingly, introducing new tagged resource add one new input vector, while every new keyword will add one component to each input vector. Thus, the input data for SOM algorithm consists of data in nxm matrix.

For the study, we collected tagging data from del.icio.us social bookmarking service. The data collections extracted from del.icio.us represent a set of tagging data from a small research community.

The data from del.icio.us included descriptions of both resources and tags associated with them. We experimented with total of eight datasets, from which statistics are compiled into table 1. Seven out of eight datasets were collected from distinct users. The last one, however, was generated from dataset A by removing all tags from first 100 resources. This was done in order to inspect differences in resulting visualisations between tagged and untagged data.

3 eXtensible Markup Language

dataset	distinct resources	distinct tags	tags after dimensionality reduction
A	327	512	188
B	423	582	220
C	402	616	264
D	35	118	21
E	94	84	47
F	49	97	46
G	142	308	52
A'	327	512	188

Table 1: Description of data in del.icio.us experiment

While SOM can be effectively used to reduce dimensions, linear growth in number of dimensions makes map creation exponentially slower. While testing the SOM Cloud application, we found out that when the number of dimensions was around 1000 tags, the visualisation was still usable. Greater number of tags resulted in fast degration of performance. On basis of this experience, it made sense to limit maximum number of tags sent to SOM client. We restricted number of unique tags by excluding all tags that occurred less than thrice. While this approach would not scale to all possible datasets, it was sufficiently effective for our immediate need.

A. Results and discussion

Visualisations of all datasets were successfully generated using both del.icio.us's own user interface and our SOM Cloud application. The tag visualisation readily available in del.icio.us was, however, found out not to scale well to these datasets (see figure 2). When number of distinct tags grew to hundreds, the tag cloud did not any more fit to single page. For this reason, we also ran focus cloud to provide one page visualisations[4].

Figure 2: Del.icio.us's tag cloud with dataset A

4 http://foobr.co.uk/focus/

Figure 3 illustrates how focus clouds rendered datasets. In the visualised focus cloud using dataset A, total of 89 tags were displayed with size of the tags reflecting the number of occurrences. User has moved mouse over the keyword “python” which has resulted in highlighting of 14 associated keywords with yellow background.

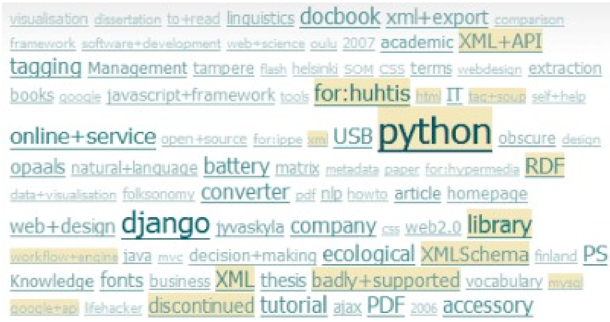


Figure 3: Focus cloud with dataset A

The illustrated focus cloud present a usual problem in tag clouds: as layout of tags is not based on tagging data semantics, there is no clear spatial interpretation. Spatially nearby tags may or may not be related. For instance in figure 3 tags 'framework' and 'software+development' are strongly related, while tags 'USB' and 'python' have very little in common. Such organisation of tags is in conflict with well-established laws of perceptual organisation and therefore may give rise to false interpretations (fc. [11]).

In figure 4, the same dataset (A) is visualised by using SOM Cloud. Tagged resources are organised to neurons, visualised as grey balls. Diameter of the circles are relative to number of resources matching the neurons. While neurons are uniformly distributed over the surface, only the ones with associated resources are displayed. As all tags represent one input dimension in the SOM, tags may be assigned to a resource located anywhere in the map. Thus, tags are rarely assigned only to resources associated to a single neuron. Therefore, tags used as labels, may represent only a fraction of tags applied.

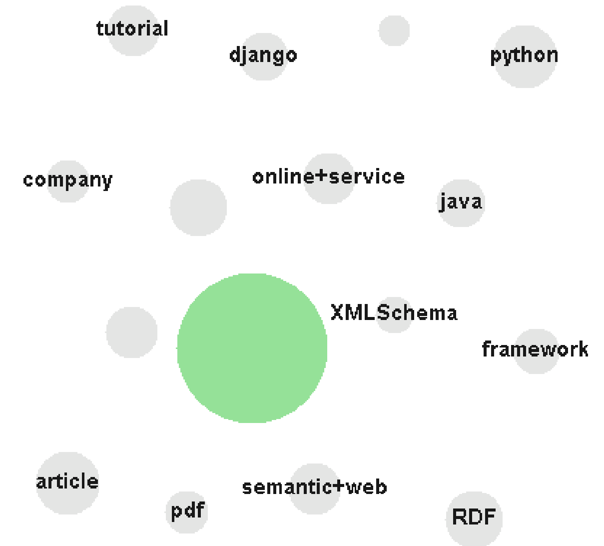


Figure 4: SOM cloud with dataset A

We also implemented a focus feature to SOM cloud. Focus is activated by clicking a neuron, after which focused region is zoomed to fit the screen. As illustrated in figure 5, more details of the neuron are displayed focus mode: tags next in relevance after labelling tag are shown around the circle, along with all resources assigned to the neuron.

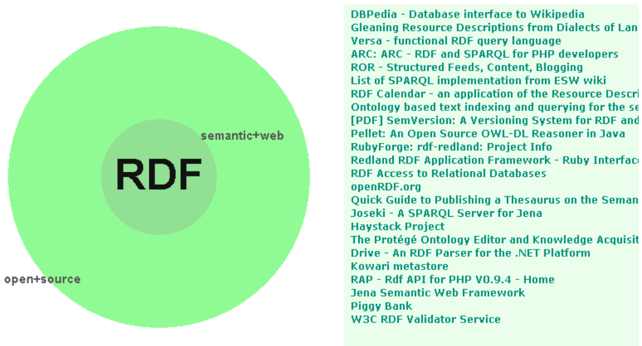


Figure 5: SOM Cloud in focus mode

Our experimenting also pointed out several problematic features in the SOM Cloud. First of all, density of tags in SOM cloud is not always uniform. Labels tend to overlap, making the interpretation of the map more difficult. While most of the overlapping was eliminated by careful text layout, there is no guarantee for overlap not to occur.

Another problematic features that emerged, was the appearance of a large unlabelled neuron, usually located in the centre of the map (also seen in figure 4). The natural interpretation for this feature would be that resources from these region do not contain any tags. This was, however, proven to be false by comparing datasets A and A'.

More likely reason for this feature is that due to the dimensionality reduction, the maps may have less neurons than there are frequently used tags: if two or more sets of completely differently tagged resources fall into same neuron, the neuron may not have a tag that is clearly prevailing. The same applies to the sets of resources that have been assigned multiple, overlapping tags: in such cases, no tag alone best describes these resource sets.

The empty region also reminds us of the fact that there is a fundamental mismatch between SOM and tag cloud topologies. A single neuron in SOM has fuzzy relationships with all tags, while in tag clouds the relationships are exactly one to one. As such, the "SOM cloud" visualisation would better fit the algorithm, if neuron labels could consist of several tags instead of a single one.

V. Conclusions and future directions

In this article, we presented idea behind resource tagging and folksonomies, described current approaches to tagging data and folksonomy use and visualisation and described and experimented with a SOM-based tag cloud visualisation method.

In our study, SOM Cloud visualisations converted surprisingly well the high-dimensional tagging data into easily perceivable tag cloud like visualisations. While we were generally satisfied with the overall quality of the resulting visualisation, the SOM Cloud has features that need to be taken into account. Especially the mismatch between SOM structure and visualised one-to-one associations between tags and neurons should be understood as a potential source of false interpretations.

It should be stressed that the power of SOM does not lie on the algorithm, but rather on how it is used: the selection of correct input components is for crucial in creation of meaningful SOM visualisations.

As for future research, we suggest that usability of the SOM cloud visualisation should be further investigated. Especially, efficiency of information retrieval using SOM Cloud, could further be studied.

Acknowledgment

This work has been supported by the European Comission (IST network of excellence project OPAALS of the sixth framework program, contract number: FP6-034824).

References

1] B. Lund, T. Hammond, M. Flack, T. Hannay. "Social Bookmarking Tools (II) – A Case Study - Connotea", D-Lib Magazine, April 2005, Volume II Number 4. ISSN 1082-9873.

2] T. Vander Wal. "Folksonomy Definition and Wikipedia", vanderwal.net, November 2, 2005, Available at http://www.vanderwal.net/random/entrysel.php?blog=1750 [accessed 13.9.2007].

3] T. Vander Wal. "Explaining and Showing Broad and Narrow Folksonomies", personalinfocloud.com, February 21, 2005, Available at http://www.personalinfocloud.com/2005/02/explaining_and_.html [accessed 14.9.2007].

4] T. Gruber. "Ontology of Folksonomy: A Mash-up of Apples and Oranges", 2005, Available at http://tomgruber.org/writing/ontology-of-folksonomy.htm [accessed 5.9.2006].

5] Y. Hassan-Montero, V. Herrero-Solana. "Improving Tag-Clouds as Visual Information Retrieval Interfaces". I International Conference on Multidisciplinary Information Sciences and Technologies, InSciT2006, 2006. To appear.

6] A. Bassett. "Focus Cloud [concept] – What's my current interest", Foobr weblog, June 23, 2007, Available at http://foobr.co.uk/2007/06/focus_cloud_concept/ [accessed 13.9.2007].

7] C. Ware, "Information Visualization, Second Edition: Perception for Design", Elsevier, 2004.

8] T. Kohonen. "Self-organizing maps", third edition, Springer, 2001. ISBN 3-540-67921-9.

9] J. Feldman. "Neural Networks – A Systematic Introduction", Springer-Verlag, 1996. Berlin, New-York. ISBN 3-540-60505-8.

10] E. Oja, S. Saski. "Kohonen Maps", Elsevier Science, 1999. ISBN 0-444-50270-X.

11] M. Preece, Y. Rogers, H. Sharp, D. Benyon, S. Holland, T. Carey. "Human-Computer Interaction", Addison-Wesley, 1994. ISBN 0-201-62769-8.

Digital Ecosystems – Community Networks or Networked Communities ?

[OPAALS Conference 2007]

Runa Sarkar
Industrial & Management Engineering
Indian Institute of Technology, Kanpur
Kanpur, India
runa@iitk.ac.in

Radhika Rajagopalan
Industrial & Management Engineering
Indian Institute of Technology, Kanpur
Kanpur, India
radhika@iitk.ac.in

***Abstract*— The role of information in promoting development has been well researched and documented, and there has been much interest in harnessing ICT (Information & Communications Technology) for this purpose. At the same time, the need to create projects that are successful as well as sustainable is recognized. We propose that adopting a more community centric design for ICT interventions can improve their sustainability and lead to the evolution of a more complex and multi functional ICT enabled socio-technical system. The digital ecosystem (DE) is one such paradigm. A DE describes an ICT enabled network that displays associative and autopieotic properties capable of self sustenance and of expansion through heightened inclusion and growth. We explore how such a system can be developed, and how this evolution can be understood in terms of whether the emphasis is on increasing individual access to ICT, or on transplanting social relations into the digital space. To illustrate this point, we briefly review a selection of current ICT interventions in India, and contrast it with the DEAL (Digital Ecosystem for Agriculture and Livelihood) project, a rural development initiative in agriculture in Northern India.**

Keywords- Digital ecosystem, Community Networks,Agriculture Social Capital

I. INTRODUCTION

There has been a change in the perceived role of Information communications Technology (ICT), from its purely technical to a human development centric one, and consequently its increasing significance in the service deployment of various development initiatives in the service of human welfare To promote human welfare, an ICT works on two angles, one of economic growth, i.e., the increases in a country's output of goods and services which can take place only when individuals participating in the economy engage in productive activities, and the other is social inclusion ; the positive action taken to include all sectors of society in planning and other decision-making. The importance of inclusion is underscored in development policies. If economic growth alone was targeted, then there is a danger of certain sections of people remaining isolated from the benefits of progress. Most importantly, for these interventions to have a long term impact, they must develop into self sustaining systems that do not collapse when the promoting agencies withdraw. To make an ICT intervention for development financially and socially sustainable, it is necessary to design a model where monitoring and continual support is not required. Instead, these functions are performed by the participants themselves, keeping the system more decentralized and flexible

In this paper, we discuss a possible solution to the sustainable deployment of ICT for development incorporating several such characteristics using the Digital Ecosystem (DE) approach and how the paradigm of increasing community participation in ICT projects to better leverage common resources is reflected in its design. In this context, we examine how a digitally enabled community evolves into a DE, and the conceptual difference between a networked community and a community network. To further illustrate this point, we undertake a brief review of some ICT aided development interventions in India, before presenting the DEAL (Digital Ecosystem for Agriculture and Livelihood) project, a rural development initiative in agriculture in Northern India discussing an evolving DE in an existing community network.

II. THE DIGITAL ECOSYSTEMS APPROACH

A. Definition and Suitability for development projects

Formally defined, Digital ecosystem describes a ICT enabled network that displays associative and autopieotic properties. In other words, not only is a so defined network capable of self sustenance, but also of expansion through heightened inclusion (i.e., increasing heterogeneity in the network composition) and growth (i.e., increase in the size and scope of the network). In simple terms, a DE is a web of interconnected and interdependent ICT enabled users who transact in the digital mode resulting in synergistic benefits for all. The strength of this system is that it enables a resilient, multi- user exchange relationship capable of adjusting to change. The natural system metaphor, employed by several schools (Rothschild , 1990; Moore , 1997; Tapscott, 2000; Power, Jerjian ,2001) to digital ecosystem as a 'digital' environment populated by 'digital species' which could be software components , applications, services, knowledge , and 'agents' or the actors in this ecosystem (individuals, SMEs, Governments). True to it's biological antecedents, the design of a DE aims to mimic a natural ecosystem to encourage autopoiesis , whereby a system produces its own organization

This work has been funded by The European/ Union's 6th Framework Programme of research under the project Open Philosophies for Associative Autopoietic digitaL ecosystemS (OPAALS)

and maintains and constitutes itself in a space

A DE for a social system needs to deal with heterogeneity and greater variations in actor's abilities and resources to participate in the network. As seen in the case of rural ICT deployment, differences in participants are induced by social and economics factors (caste, income group), level of education and exposure and so on. The vision of a DE as network that finally evolves into an "agents-based, loosely coupled, domain-specific and demand driven interactive communities which offer cost-effective digital services and value-creating activities that attract agents to participate and benefit from it", makes it capable of accommodating these variations by encouraging the co-existence of different species. This description also underscores the critical importance of participation to the success of a DE – in terms of growth, sustainability and inclusion. Participation here refers to both content sharing and creation. Sustained growth and heightened inclusion are the keys to successful development .Thus it is essential that rather than make recipients of assistance dependent on the provider, the providers should create the right digital environment where recipients can exercise their choice on the nature and extent of assistance they require. A big asset of a DE, in this context, is that it is intrinsically designed to be self sustaining. A DE functions independent of the entry or exit of individual actors. This is achieved by functioning as a platform fostering various economic (business) and social networks involving a multiplicity of actors engaged in dynamic and amorphous interactions. There is no single entity guiding or directing activities and information flows. Instead, all actors share the responsibility of running the network, by sharing information, resources and interacting with others, making the system robust and less resource intensive in contrast to some of the other ICT implementations discussed earlier

B. *Evolving a DE*

In this context, we briefly examine a possible evolution graph for a social digital ecosystem. The biological entity analogy is invoked once again when discussing the setting up of a digital ecosystem. The system has to evolve gradually from a collection of single cells to a complex organism, which will function even as some cells die out and new ones are created.

The system evolves through the following stages :

Stage 1 : Physical Communities

Members from various offline communities begin adopting basic e-tools like internet, email, online search engines and so on. There is no significant change in the flow of information between members neither of the community nor in the operations

Stage 2: Electronic Communities

Adoption of e-technologies by members forms the next step in transferring the information flows to the digital domain. With more participants using e-tools, online services start evolving which require two way information flow, or at least permit online information access , like in the various e-governance services. In this stage, the network evolution is sustained by a promoting agency, like various Government departments, NGO sponsors etc., which acts as a central authority – financing and monitoring the project

Stage 3: Digital Ecosystem

The evolution of the network breeds a dynamic cooperation between the players which encourages the online communities into greater sharing of knowledge and resources, while providing more opportunities for economic and social development. Local content is created and shared throughout the network. New services evolve online instead of different agencies developing services to meet member needs, increasing the types of players involved. These benefits encourage other members of physical communities to join the network. Most of all, the responsibility of sustaining network is shared by all the participants and the role of the e-champion is reduced. The ultimate aim to create a participative society, which supports economic inclusion, empowers the creativity and participation of all potential action (public and private organizations, communities, individuals) in open socio-economic processes.

Having discussed a possible evolution model for a DE, we ask the question: is gradually networking a community enough to ensure that it eventually becomes a DE? In other words, does providing digital access transplant communities into the virtual domain where their interactions mimic complex real world transactions?

To tackle this question, we discuss a cross section of ICT interventions adopted so far in India. One objective common to all ICT programs is that of attaining self sustenance. We propose to examine in the following programs, where the focus has been on providing increased computing and digital facilities to the community members, and see how successful they have been in becoming independent and self sustainable.

III. EVALUATION OF OTHER ICT ENTRENCHED DEVELOPMENT SOLUTIONS

The earliest appraisal of the potential of computers in rural development in India was done by academics during 1975-80 (Patel, 1979). Initially, computers were seen as effective monitoring tools of existing projects (Bhatnagar and Patel, 1988), but gradually, with declining hardware prices, their use became more widespread and non-specific. Under the patronage of Rajiv Gandhi, the pace of IT use at the district level gained momentum and many programmes , like DISNIC (District Information System of National Informatics Centre) which promoted computerization of all district level offices and CRIS (Computerized Rural Information Systems Project) which developed software for planning and mentoring of IRDP (Integrated Rural Development Programme) , laid the groundwork for future ICT enabled projects. Government interest has however been more marked in supporting e-governance initiatives to make governance more transparent and delivery of certain essential services corruption free, as responsible governance will automatically encourage

economic growth and social development. We now discuss selected ICT enabled development programmes and examine the performance of each in context of the goals they were to set to achieve. We draw on the extensive reports by DIT and Infosys (2005), APARC and NIC (2005) etc, comparing the various ICT projects and evaluating their aggregated performance.

A. Some Selected Schemes

1) Gyandoot

In the Dhar district of Madhya Pradesh, Gyandoot— a government-owned computer network—has been trying since January 2000 to make government services more accessible to villagers. Gyandoot aims to reduce the amount of time and money people spend trying to communicate with public officials and seeks to provide immediate, transparent access to local government data and documentation through a network of 38 telekiosks or soochanalayas.

Gyandoot aims to create a cost-effective, replicable, economically self-reliant and financially viable model for taking the benefits of Information and Communication Technology (ICT) to the rural masses. The purpose of this network is to combat poverty, illiteracy and social backwardness by promoting awareness and participation by citizens/government in community affairs through creative uses of ICT and also encouraging greater inclusion by ensuring equal access to emerging technologies for the oppressed and exploited segments of the society (Rajora, 2002) Information and services offered include online applications for caste, income, and domicile certificates and the access to the list of people living below the poverty line, prices of several agricultural products in various cities beyond the local market. Additionally, a public complaint line for reporting government-related problems, broken irrigation pumps, unfair prices, absentee teachers, and other problems is also available

a) Model of operation

The information collection and dissemination is done though the tele-kiosks. These are different from telecenters in that they are have only a single computer manned by a facilitator or a 'soochak' who helps clients browse or download, while telecenters have personal computers where clients browse on their own (Best and Maclay (2002))

There are two models of ownership of these tele-kiosks- the panchayat (village committee) and the private models. In both models, individuals from the village community are handed the running of the kiosks, thus fostering entrepreneurship.

In the panchayat model the village committee pays for the telekiosk's capital expenditures (space, hardware). Operators have to bear telephone expenses and do not receive a salary, but get to keep 90% of earnings after remitting 10% to the panchayat. For appointing kiosk operators, panchayats select three people who are sent for training, and one is finally selected after interviews and a practical examination. In the private model the soochak is the owner of the telekiosk. The entrepreneur who has the capital, or is able to get a loan, gets the job. An annual payment of Rs.5,000 to the project committee aside, the rest of the earnings accrue to the kiosk operator.

b) Overview of performance

Based on World Bank's Governance Knowledge Sharing Program (GKSP)((supported report by CEG-IIMA (2002)) we make the following observations on Gyandoot :

- High service satisfaction but low usage : Villagers who had used the Gyandoot kiosks were satisfied with the service level offered but they did not advocate it to others. Out of 38 Gyandoot telekiosks, the CEG-IIMA survey found that 10 were not operational. Many telekiosks serve only a handful of people (1–4) each day. The average for 18 telekiosks calculated over a two-year period was 0.62 users per day.
- Despite the publicity initiatives undertaken by the project committee, there was a lack of awareness about the facilities offered by the project. Distance and infrequent power supply were two other reasons for people not using the kiosk.
- Services provided as a response to a need, yet many services underutilized: The Gyandoot project aimed to provide a whole host of services, including access to educational and health facilities by linking up with schools and local hospitals. Survey results indicate that most of the services were not used at all by villagers. Of the 20 services offered , only a handful are really demanded (CEG-IIMA, 2002)
- Low local content creation: lack of involvement and contribution from local players. . Basic nature of project is that of a G2C network , but to achieve the stated aims would require participation of other actors (partnering with private players, NGO's, research institutes) to provide additional services for which there was not sufficient scope.
- Financial feasibility: The cash flows generated by the services provided are not sufficient to help operators sustain. The CEG-IIMA survey found that total revenue from Gyandoot services was approximately Rs. 150 per month per telekiosk over a period of 2 years. The Gyandoot Intranet was set up at a total cost of Rs. 2.5 million. The average cost incurred to establish a telekiosk is Rs. 75,000 and operational costs are estimated at Rs. 1,000 per month (Bhatnagar & Vyas, 2001). Clearly, there is no incentive for more kiosks to be set up or for individuals to continue in the project.
- Issues with existing administrative set-up- cost escalation due to new modes of corruption and bribery: Gyandoot has computerized only the front-end of government services; in most cases, citizens submit applications online and have to go back to the telekiosk for a response. Back-end processes, at government department levels, are not computerized. Printouts of

the applications, requests, and grievances are sent to government departments for further action, except for a couple of departments that are accessible by e-mail. This leaves loop holes for new forms of corruption.

2) IVRP

The Information Village Project in Pondicherry, India started up in 1998. The project was initiated and managed by the Madras based M. S. Swaminathan Research Foundation with the support of the Canadian Government. The objective was to assist sustainable agriculture and rural development by making relevant generic information locale specific and delivering it to the community. The thrust was on involving woman, BPL families and members of the so called back-ward castes . The project implementation was planned in three phases.

a) Model of operations

The project uses a donor-grant based community centric model. Village knowledge centres (VKCs) are set up in order to provide rural communities with access to Internet and training on ICTs. In each case, the participating community provides rent free access space to set up and 2 to 4 volunteers. These volunteers are compensated by the community and care is taken to involve the marginalized sections in volunteering. In turn, the project provides all the needed equipment, training and data. wireless radios were used for data and analog voice transmissions between a semi-urban hub center and eight village centers. Information gathering follows a Hub and spoke model—Hub gathers and distributes information to the Centers. Volunteers are trained in the basic operations of a personal computer: sending messages, composing documents and so on, using local language tools etc. They maintain the VKC and feed relevant content garnered from other sources like newspapers, local news etc to the network. In each phase there is an updation in the ICT tools used and phase III of the project is looking to experiment with the following: wireless fidelity (WiFi); 2.5G mobile technology (includes mobile telephones enabled to transmit data via General Packet Radio Service); Global Positioning System (GPS) for fisher people to improve knowledge of fishing zones and potential ocean hazards etc. VKCs were first created in seven hamlets in rural Pondicherry, and the number has gone to twelve centers, with an average of 25 users daily.

b) Overview of performance

- The success of IVRP has been in terms of local content creation and dissemination. Small groups of volunteers independently picked extra expertise (HTML coding, transmitting voice files) to improve content creation. The information gathered at the telecentres is also feeding more traditional media. A volunteer-run, twice-monthly, community newspaper, Namma Ooru Seithi, was launched in early 2002 to reach those beyond the knowledge centres' ambit. Its articles cover topics of local interest such as agriculture, traditional health care, jobs, coming training programs, recipes, child care tips, and village-specific news
- IVRP illustrated the 'bottom-up' approach to ICT based implementation of development projects. Evolution of the IVRP network infrastructure was gradual, based on demand for services by the community and its willingness to manage and volunteer for the infokiosks.
- Interestingly, the project offers very few e-governance based services. IVRP is dependent solely on its donor institution to be kept running. The project does not seem designed for self sustenance. Most services are information provision based and available free of cost

While the Gyandoot is a government supported ICT initiative meant primarily for delivering e-governance applications and development opportunities, the IVRP network sought to fulfill the same objectives using a more community focused , less e-governance centric service portfolio . We next discuss the case of the Wired Warana project, again a government sponsored initiative, where the core services were more aligned to the operations of the village co-operative.

3) Warana Wired Village

The Warana Village information kiosk was originally conceived as a MIS initiative to assist the Warana Group of Cooperatives (WGC) in make information about the yearly registration for plantations, issuance of harvesting permits and salary payment information available to the co-operative employees via Internet:

Initiated by the Prime Minister's Office Information Technology Task Force, the goal of the Warana "Wired Village" project, in addition to increasing efficiency and productivity of the sugar cane cooperative, is to provide a wide range of information and services, in the local language, to 70 villages around Warana., including crops and agricultural market prices, employment schemes from the government of Maharashtra, and educational opportunities.

a) Model of operations

The different participants involved in the project are: The Warana sugar and Diary co-operatives, Warana Co-operative bank, the Warana engineering college and the Mahatma Gandhi medical trust. The project follows a hub and spoke model. The Sugar Administrative Building (SAB) of the sugar factory and the engineering college for the main hub centre. The business centers and IT centres are the value- adding tier of the structure where data entry takes places. Village level facilitation kiosks provide connectivity down to the village level. The kiosks have a PC with a printer and most are connected to CAB via wireless telephony.

b) Overview of performance and comments

- Implementation of the initiatives having a direct bearing on the operations of WGC proceeded smoother than those intended for community development. The information flows and delivery mechanisms were easier to map for the

- Top down implementation left lesser scope for community participation. Though the union was supportive, the lack of local staff participation in the software development and implementation process, which was done by a central nodal agency in Delhi. This may explain why some applications are not reflective of community information requirements, as in the case of information resources on sugar cane growing and agricultural prices which is lying unutilized and un-updated since 1998.

- Rapid deployment of the different services of the project, without spending some time in training the community in the use of developed software may have been counterproductive. A more phased implementation, as seen in the case of IVRP would have been more appropriate.

- Not too many e-governance services are offered through the project. This is surprising considering the strong Central government involvement (NIC, PMO) in the setting up of the project.

The Wired Warana project is an example of community development schemes integrated into a ICT initiative built for improving business operation's efficiency. A comparable project is that of ITC's eChoupal, the only difference being that the promoter in this case is not a government agency but a private player.

4) ITC eChoupal

Started in 2003 with soya growers in the villages of Madhya Pradesh, eChaupal is a nation-wide e-procurement network set up by ITC Limited. The objective was to cut across layers of intermediaries and deal directly with the farmer, increasing the efficiency of the supply chain and at the same time, ensuring that the farmers received a fair price for their produce. Innovatively combining technology, sociology, and the incentives of the various players involved, the eChoupal provides farmers with effective methods of price discovery, honest trading, and information sharing to the benefit of all in the channel The project presently covers 9 states and 38,000 villages and has enlarged it's scope to cover shrimp farmers and coffee cultivators.

a) Model of operations

E-choupal follows an info-kiosk based franchisee model. The following is brief description of the soy bean procurement system. Selected farmers, called sanchalaks, host the computer in their houses. This computer is linked to the Internet via phone lines or, increasingly, by a VSAT connection. One node serves an average of 600 farmers in 10 surrounding villages within about a five kilometer radius.

The previous day's mandi (local market) closing price is used to determine the benchmark Fair Average Quality (FAQ) price at the e-Choupal. The benchmark price is static for a given day. This information and the previous day mandi prices are communicated to the through the e-Choupal portal. Farmers can also check prices at several nearby mandis and even track global trends. Thus the farmer is empowered to make the critical decision of when and where to sell his crop.

To initiate a sale, the farmer brings a sample of his produce to the e-Choupal. The sanchalak inspects the produce , performs prescribed quality tests and based on his assessment gives the farmer a conditional quote. If the price is agreeable to him, the farmer takes the note from the sanchalak and proceeds with his crop to the nearest ITC procurement hub where further tests are conducted on crop sample. These tests are however conducted after the sale has been finalized and have no bearing on the price. After the inspection and weighing are complete, the farmer then collects his payment in full at the payment counter. All operations at the procurement hub are handled by operators called 'Samyojaks'. The emphasis at the hubs is on professional handling of all transactions, with speed and accuracy.

b) Overview of performance and comments

- E-choupal is primarily a revenue generating initiative. The purpose of investing significant time and resources in this venture was to improve ITC's supply chain management. Like in the case of the Warana project, backbone services are related to the core business and other additional services come next. The benefit of this approach is that it keeps the project sustainable. The focus is however, understandably not centered on development. Thus, while the community benefits from the project through access to better earning opportunities, there are no services to address the needs of the minority sections, women or landless workers. The project unintentionally discriminates among the local population.

- The project while providing many economic benefits to farmers doesn't necessarily change the social framework they operate in. Studies have pointed out that in many cases, sanchalaks are farmers from the forward communities and so members of the backward castes are not allowed to enter the house in which the eChoupal is located.

- E-governance initiatives are underrepresented in the project. Part of the reason is ITC's hesitation to partner with governments given the uncertain quality of response.

- The whole initiative is completely dependent on ITC. All training provided to operators in handling the ICT tools are company specific. This doesn't automatically translate to making users self reliant.

Table 1 provides a ready reckoner for these projects.

TABLE I. SELECT ICT INTERVENTIONS IN RURAL INDIA : AN EVALUATION

Project	Goals	Services	Model of operations	Overview of performance
Gyandoot *Government, Village level committees*	To make government services more accessible to villagers To promote awareness and participation by citizens/government in community affairs through ICT,	- e Governance services : government services (eg : online caste , income and domicile certificates) and official information - Agriculture support information - Communication facilities like e bulletin boards, email etc	Dissemination of information through telekioskes (only one PC used by a *facilitator*) Total of 38 kiosks Two models of ownership - private entrepreneur : owner puts in capital , makes annual payment to village committee - Village committee : committee pays for the telekiosk's capital expenditures (space, hardware). Operators bear telephone expenses and do not receive a salary. Instead they keep 90% of earnings after remitting 10% to the panchayat. For appointing kiosk operators, panchayats select three people who are sent for training, and one is finally selected after interviews and a practical examination.	- High service satisfaction but low usage – CEG-IIMA survey (2002) reported that 10 of 28 kiosks were not operational - Many services under-utilized- users very unaware of many of the service offerings. - Low local content creation – This is indicative that there was no significant engaging of different actors in collaborative efforts. Low word of mouth publicity for the project within service district despite campaigns also surprising. - Low ROI to kiosk owners (private and committee) – not financially tenable - New modes of corruption: computerized only the front-end of government services; Back-end processes, at government department levels, are not computerized. This leaves loop holes for new forms of corruption.
IVRP *MSSRF with support of the Canadian Government. Village level committees*	To assist sustainable agriculture and rural development,involve marginalized sections of the community- To provide rural communities with access to Internet and training on ICTs,		A donor-grant based community centric model. Information dissemination follows Hub and spoke model Village knowledge centres (VKCs) are set up in order to provide rural communities with access to Internet and training . MSSRF provides all the needed equipment, training . The participating community provides rent free access space to set up and 2 to 4 volunteers Volunteers are trained anf maintain the VKC , feed relevant content garnered from other sources like newspapers, local news etc to the network VKCs were first created in seven hamlets in rural Pondicherry, and the number has gone to twelve centers, with an average of 25 users daily.	- Successful local content creation and dissemination. Small groups of volunteers independently picked extra expertise (HTML coding, transmitting voice files) to improve content creation. The information gathered at the telecentres is also feeding more traditional media. A volunteer-run, twice-monthly, community newspaper, Namma Ooru Seithi, was launched in early 2002 covering topics of local interest such as agriculture, traditional health care, jobs, coming training programs, recipes, child care tips, and village-specific news -.'Bottom-up' approach to ICT based implementation of development projects., based on demand for services by the community and its willingness to manage and volunteer for the infokiosks. - Dependent solely on its donor institution to be kept running.. Most services are information provision based and available free of cost
WWI *WGC overnment of Maharshtra*	To assist the Warana Group of Cooperatives providing a large range of information and services,	Information and services, in the local language, to 70 villages around Warana., including crops and agricultural market prices, employment schemes, and educational opportunities.	Hub and spoke model. The Sugar Administrative Building (CAB) of the sugar factory and the engineering college for the main hub centre. The business centers and IT centres are the value- adding tier of the structure where data entry takes places. Village level facilitation kiosks provide connectivity down to the village level. The kiosks have a PC with a printer and most are connected to CAB via wireless telephony.	- Top down implementation left lesser scope for community participation. Though the union was supportive, the lack of local staff participation in the software development and implementation process, which was done by a central nodal agency in Delhi. - Rapid deployment of the different services of the project, without spending some time in training the community counterproductive. some applications are not reflective of community information requirements, as in the case of information resources on sugar cane growing and agricultural prices which is lying unutilized and un-updated since 1998 - Offers few e-Gov services.
eChoupal *ITC Ltd, Farming communities across*	To promote an e-procurement network To empower rural agricultural community, increase modal incomes and service agriculture, healthcare, education, gender and community needs	- Information (weather, prices, news), - Knowledge (farm management, risk management) - sales of Farm Inputs & Consumption goods (screened for quality, price) & alternative output marketing	Info-kiosk based franchisee model Farmers track agricultural prices on the Internet available at a Facilitator's (also a farmer) residence in the village To initiate a sale, the farmer brings a sample of his produce to the e-Choupal. After inspection and quality tests, a conditional quote is generated. If the price is agreeable \the farmer proceeds with the nearest ITC procurement hub After the inspection and weighing are complete, the farmer then collects his payment The emphasis at the hubs is on professional handling of all transactions, with speed and accuracy.	-Major cost savings to company and farmers. Increased incomes, with increasing participation from farmers -The project while providing many economic benefits to farmers doesn't necessarily change the social framework they operate in. -E-governance initiatives are underrepresented in the project -The whole initiative is completely dependent on ITC. All training provided to operators in handling the ICT tools are company specific. This doesn't automatically translate to making users self reliant

B. Overall Observations

In the ICT enabled development schemes discussed above, we note that the solutions were not as effective as intended. All of them required constant supervision and involvement by their project promoters. The schemes required making large capital investments as well as meeting continuous recurring expenses. Each of the schemes are identified with their promoters and the participants engage because of trust in the principle mover, not in the system itself. This has a significant bearing on sustainability, in that , should the promoter withdraw financial or technical support from a project at anytime, other participants may not step in to continue its functioning. Moreover, though all the projects operate in rural, primarily agriculture based communities none of the systems actively robustly involves research institutes or agricultural domain agencies. Except in case of IVRP, there has been emphasis on creating and assimilating locally relevant content. Over all, given the wide variety of facilities, there has been a marked improvement in communication and awareness and in some instances, a positive effect on restrictive social norms. While incomes have improved in some village communities (IVRP, eChoupal) this has been more due to better recognition of market opportunities rather than a direct positive effect on yield. The success or failure of a project aimed at fostering development by increasing the information capital , whether a Government undertaking or otherwise, depends not only the efficiency of the applications provided, but also on involving communities collectively and not just individuals. Successful adoption of new technologies requires collective action and co-operation, which social capital helps secure. Research indicates that in the Indian rural agriculture sector social capital is pivotal in mobilizing resources and bringing about market unsupported outcomes. This phenomenon has been documented for adoption of new seed or production technology, as well as for dissemination of information to other rural communities, in the absence of official channels (Chopde and Parthasarathy, 1999)

There is a need to distinguish between a networked communities and a community network. A networked community is one where all members have access to internet and computers; the emphasis may not be on whether the entire community is interacting online. A community network represents the links between different members and different groups in a community, forged through various media, both digital and in person. In the ICT interventions discussed, the attempts focused on bringing as many community members online as possible, to directly access the information digitally available. Though kiosks and other common access points have been established and staffed to assist farmers to use different tools, the services, like email, online complaint registration etc are aimed at the individual and not the community. The effectiveness of these services is diluted as many farmers using the ICT program may not be capable of navigating through the system, and instead of asking constantly for help, they avoid it altogether. Unless a networked community evolves into an online community, the existing social capital does not get transferred to the new system, and therefore, as soon as the supporting node withdraws, the sustainability of the project is compromised.

When understanding how ICT can be integrated in development projects, it is reasonable to expect that outcomes are dependent on "social contexts of design, implementation and use" (Rosenbaum, 1999). Studies have shown that outcomes of ICT implementation and use in different real world settings, be it in an organization or a educational community, are difficult to predict or replicate, and that the "contextually dependent nature of ICT's suggests that similar ICT's can have different outcomes in different situations" (Kling,et al, 1994)

Along the similar lines, the impact of ICT on a community is also important. Conflicting views abound – whether social capital is harmed or enhanced by information technology. Studies have shown that ICT facilitates in the formation and maintenance of 'weak' ties and may be harmful to strong ties (Sabatini) , or alternatively, ICT helps maintain strong ties, like those with close family , over distances through the use of email, chat etc. Weak ties in the network have been traditionally associated with enhanced economic activity (Granovetter, 1985). This makes ICT interventions very attractive for economic development Another school of thought (Kali, Farmer) is that decreasing social involvement and community participation as detailed by Putnam (2001) is further exacerbated by ICT, as users becomes more disconnected from their immediate community. Given the positive role of weak ties on enhancing economic growth, and the propensity of ICT to promote these ties, the contention is that proliferation of weak ties moves the individual out of the ambit of his immediate social group and alienates him from his surroundings. Also, when dealing with members outside one's community, the basis for trust is reduced, and the effectiveness of social capital does not translate directly online. The evolution of institutions is required to retain coordination. This isolates the economic sector from the social one, and the dynamics are so poised that development in one sector can have a negative impact on the other. Empirical evidence available is not sufficient to conclusively prove either claim, though there is strong evidence of technology promoting some weak ties. One learning, however, is that regardless of whether ICT is viewed as social capital enhancing or depleting, ICT in itself is not context independent and ICT interventions can be suitably designed to use existing social relations and meld them into the digital network, broadening their scope and reach at the same time.

C. The DE Approach in a Community Network: The case of the Indian DEAL project

The DEAL (Digital Ecosystem for Agriculture and Livelihood) project is a step in towards addressing these issues by assembling a technology enhanced agricultural extension intervention in a DE framework. Conceived by IIT K and Media Lab Asia, DEAL is a ICT enhanced network built on an existing framework of tele-centers in rural institutes, village schools, village level agriculture extension centers (KVKs) and other deployment partners. The project aims to create a

digital knowledge base by involving the various actors in the existing system in the content creation process and making this knowledge accessible to farmers and other agricultural practitioners. The entire process of content creation and dissemination is self generating, node independent and self sustaining using an electronic medium. The moderating node in this system is IIT Kanpur- providing the collaboration and collation technology platform, skills and resources to assist knowledge flows through the network. The presence of Government agencies helps build trust in the network. The agricultural experts and educational institutions are responsible for verification of content generated. The field deployment of the project was between December 2006 and June 2007. A study was conducted at 4 KVKs in Sept 2007, to assess the effect DEAL has had on information flows. The following network diagram, prepared in NetDraw[1], represents the ties that were present before the implementation of DEAL.

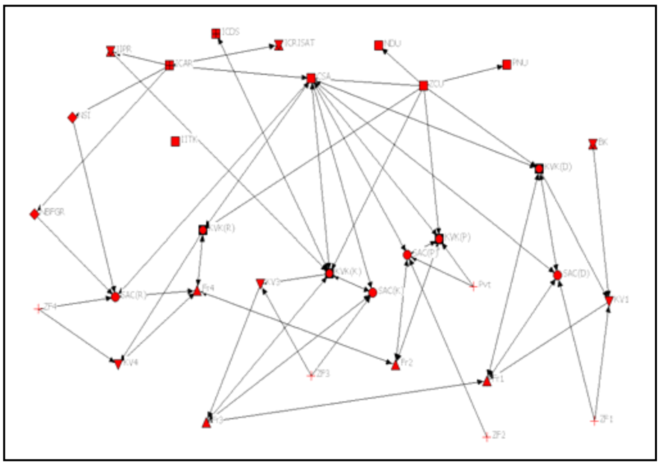


Fig.1: Network Ties before DEAL

The shapes of actors in the network are based on their role, i.e. KVKs, Research Institutes, Government agencies, etc. The thickness of lines between agricultural experts within the same KVK, or between a farmer and his respective KVK are examples of strong ties, while links between KVKs and NGOs are examples of weak ties. The network shows the information flows within and across community. Here, the community is understood in terms of the village unit. So, within community linkages are those between actors in the same village – for eg, between the farmer of a village and the respective village KVK. Across groups links include links between actors from different villages – like the link between farmers of different villages. Another classification of group is in terms of the strata of operations the actor belong to – IITK is a member of the educational institutions group, KVK are part of the villages level functionaries, and the ZCU , ICAR are all implantation and monitoring agencies. In this above network diagram , we have represented the different sources of agricultural information and the interrelations , both formal and informal, between them. Formal links are characteristic of the reporting relationship between actors – for instance, in the case of a KVK and the ZCU (Zonal Co-ordination Unit), and informal links are characteristic of the social relations between actors – like relations between farmers of adjoining villages.

Fig 2 represents the ties after implementation of the DEAL project. IITK is the only completely new actor being introduced into the framework and its integration into network, as well resultant increase in information flows is evident

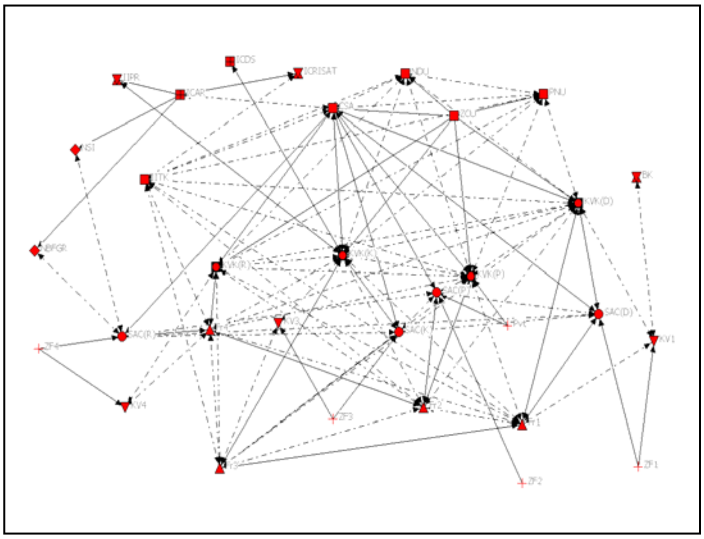


Fig.2: Network Ties post DEAL

The dotted lines represent ties that have been formed due to content co-creation and sharing by partners facilitated by IITK through DEAL, while the solid lines represent the preexisting network ties. Linking together all the actors in dynamic relationships helps retain both strong and weak ties. We present here a few results[2] from the analysis done using Ucinet[3]. The total number of ties increased from 77 to 183, and no old ties were displaced. New ties between existing partners, for instance between the KVKs, has decreased the degrees of separation between them while also increasing the network density from 0.0726 to 0.1615 showing a significant increase in cohesion between members. Another indicator of this increased interaction is the group reciprocity measures increasing from 0.3585 to 0.7745. From accessibility point of view, by using DEAL, farmer of all 4 KVKs are at a maximum 3 degrees of separation from any actor in the network. While these early reports are in some measure indicative of positive network externalities, more field reviews are needed to confirm these results. Since social change is an evolutionary phenomenon requiring time in the order of decades to manifest significantly, this is an area for further investigation.

DEAL is different from a networked community like the Seattle community network (SCN), for example, in that the

[1,3] Software from Analytic Technologies

[2] Details of study available at IITK Deal site

focus in not on promoting individual web presence, but more or extending reach of information flows while reducing the degrees of separation between members. This is done by using both digital and non-digital media. For instance, at the village level, the farmers, lacking access to computers or internet, have been introduced to the DEAL portal by the village KVK scientists. The idea is that using DEAL, they are able to use the online databases to address their problems – for instance, the picture particular kind of pest can be picked out from the common pests' database to search for the right insecticide, or the details of different rice varieties can be to find the appropriate fertilizers. A tool called the 'Kisan Blog' helps access the experiences recorded by members with respect to practical applications – grade of seeds, fertilizer or pesticides used etc. The endeavor is to make information flows and not the online network the focus of the project. This way, participants are 'won over' to use and 'socialise' the use of the system, and as prevalent infrastructure problems are resolved, the community network, with its social capital in place, makes a transition to the digital medium, that is a networked community, which is also a digital ecosystem.

IV. Conclusion

For an ICT intervention to evolve into a DE, there are several necessary and sufficient conditions. This paper has tried to demonstrate that while establishing adequate infrastructure for a networked community is necessary, it is by no means sufficient for a DE. The associative and autopoetic nature of a DE necessitates the need for embeddedness of the intervention in existing social structures, which would grow through weak links to strengthen social capital and create new norms and networks. Thus, a digital ecosystem could be described as a community network of networked communities, which can grow in quality and quantity without any external assistance. The Indian implementation of the DEAL project illustrates the formation of a community network in the agricultural knowledge space. Here, the enabling parameters for developing into a DE are in place and with time (and adequate infrastructure) the transformation is inevitable.

References

[1] Ben-Porath, Y., The F-Connection: Families, Friends, and Firms and the Organization of Exchange. Population and Development Review, (Mar., 1980),. Vol. 6, No. 1.: p. 1-30.

[2] Bhatnagar, S. and N. Vyas. (2001). "Gyandoot: Community-owned rural Intranet kiosks." E-Government Case Studies.

[3] Bhatnagar, S. and R. Chawla (2005). Bhoomi: Online delivery of record of Rights, Tenancy and Cultivation to farmers in Karnataka. Land Reforms in India - Computerization of Land Records. W. Habibullah and M. Ahuja: 47-77.

[4] Bhatnagar, S. and N. Patel (1988). "Decentralized Computer for Rural Development'." OMEGA 16(1).

[5] Borgatti, S.P., Everett, M.G. and Freeman, L.C. 2002. Ucinet for Windows: Software for Social Network Analysis. Harvard, MA: Analytic Technologies.

[6] Buskens, V., The social structure of trust. Social Networks, 1998. 20: p. 265-289

[7] Chopde V.K, D.Parthasarathy , Building Social Capital: Collective Action, Adoption of Agricultural Innovations, and Poverty: Reduction in the Indian Semi-Arid Tropics.

[8] Chatterjee, J. and T. V. Prabhakar (2005). On to action- Building a digital ecosystem for Knowledge difussion in rural india. International conference on Knowledge Management, North Carolina, USA

[9] Cecchini, S. and M. Raina (2004). "Electronic Government and the Rural Poor: The Case of Gyandoot." The Massachusetts Institute of Technology Information Technologies and International Development 2(2): 65–75.

[10] Coleman, J.S., Social Capital in the Creation of Human capital. The American Journal of Sociology, 1988. 94

[11] Dini, P. and F. Nachira. (2005). "The Paradigm of Structural Coupling in Digital Ecosystems."

[12] Dossani, R., D. C. Misra, et al. (2005). Enabling ICT for Rural India, Asia-Pacific Research Center, Stanford University ,National Informatics Centre.

[13] Fahey, L. and L. Prusak (1998). "The eleven deadliest sins of knowledge management." California Management Review 40(3): 265-76.

[14] Granovetter, M., Economic Action and Social Structure: The Problem of Embeddedness. The American Journal of Sociology, Nov., 1985,. Vol. 91, No. 3.: p. 481-510.

[15] Grover, V. and T. H. Davenport (2001). "General perspectives on Knowledge management : Fostering a research agenda." Journal of Management Information Systems 18(1): 5-21.

[16] Keniston, K. and D. Kumar, Eds. (2004). IT Experience in India : Bridging the Digital Divide, Sage.

[17] Koert, R. v. (2000). "Providing Content and Facilitating Social Change: Electronic Media in Rural Development Based on Case Material from Peru " First Monday 5(2).

[18] Krishna, A., Understanding, measuring and utilizing social capital: clarifying concepts and presenting a field application from India January 2003 , CAPRi WORKING PAPER NO. 28

[19] Krishnan, R., M. D. Smith, et al. (2003). "THE ECONOMICS OF PEER-TO-PEER NETWORKS." Journal of Information Technology Theory and Application 5(3): 31-44.

[20] Kwok, J. S. H. and S.Gao (2004). "Knowledge sharing community in P2P network: a study of motivational perspective." Journal of Knowledge Management 8(1): 94-102.

[21] Markus, M. L. (2001). "Toward a Theory of Knowledge Reuse: Types of Knowledge Reuse Situations and Factors in Reuse Success." Journal of Management Information Systems 18(1(Summer)): 57-93.

[22] Owen, A. L. and J. Videras. (2007). "Trust, Cooperation, and Implementation of Sustainability Programs: The Case of Local Agenda 21."

[23] Rajora, R. (2002). Bridging the Digital Divide, Tata- McGraw Hill.

[24] Sen, A. (1997)."Maximization and the Act of Choice." Econometrica , Vol. 65(No. 4.): 745-779.

Community Currencies

A Critical Approach to Community Currency Systems

[OPAALS Conference 2007]

Panayiota Tsatsou
Department of Media and Communications
London School of Economics and Political Science
London, UK
p.tsatsou@lse.ac.uk

Abstract: **This paper discusses community currencies and their importance, positioning trust at the epicentre of the interest. The paper draws on ongoing work in the Opaals project and the task T6.3 on Community Currencies for Digital Ecosystems, discussing the ground of community currencies and the benefits deriving from their employment. After some introductory remarks on money and community in section I, section II presents a critical approach to community currencies and their benefits. More specifically, the history and ground principles, as well as the rules of community currencies are discussed, providing a clear overview of community currencies' characteristics. Then, the rationale for the employment of community currencies is briefly presented, whereas the element of trust is further discussed both as a requirement and a benefit in a community currency system. Finally, possible risks and challenges resulted by the employment of community currencies are presented, while the paper concludes in section III by summarising and stressing the importance of looking at community currencies in a Digital Ecosystem environment.**

Keywords: money; community; community currencies; trust; Digital Ecosystem

I. Introduction: money, community & community currencies

> Money is like an iron ring we've put through our noses. We've forgotten that we designed it, and it's now leading us around. I think it's time to figure out where we want to go – in my opinion toward sustainability and community – and then design a money system that gets us there [13]

The quotation above refers to the role of money in community-building and the importance of putting money in a community context. Money can be defined as goods or tokens used in trade as a medium of exchange, store of value, and unit of account. Usually, money refers to currency, particularly currencies with legal tender status conferred by a national state.

Money was first in the format of commodities where individuals, communities and social systems were exchanging various kinds of commodities to which different values were attributed. In the Medieval era, the barter economy model and the community exchange of goods and services in parallel to a broader national economy system were significantly facilitated. In this barter model, fairly horizontal structure, reciprocity, distribution and trust were established as ground rules, whereas the use of multiple commodity currencies constituted a prototype of today's community currencies.

All commodities used as exchange systems had an intrinsic value beyond their use in exhcanges, whereas nowadays paper notes are only worth as much as the monetary value assigned to them. Thus, fiat money issued by a central authority (government) is a new money object imposed as a medium of exchange by law, having no intrinsic value and being used as:

- Medium of exchange - an intermediary used in trade;
- Unit of account - a numerical unit of measurement of the market value of goods and services;
- Store of value - to be saved, stored and retrieved.

The fundamental swift from commodities to fiat money highlights the social constructivist character of money, as money is assigned value on the basis of how it is understood and used within society and in the same way in which ancient exchange systems used to assign value to goods: 'money has become a form of social information, something of absolutely no value in itself that we use to measure how we value other things, including our time' [30].

On the other hand, community is a social reality itself which can take various formats in different times and spaces. The word *community* comes from the Latin *communis,* meaning "common, public, shared by all or many' [11]. The Latin term "communitatus", from which the English word community comes, consists of three elements: 'Com-' - a Latin prefix meaning with or together, '-Munis-' - meaning the changes or exchanges that link (both municipal and monetary take their meaning here) and '-tatus' - a Latin suffix suggesting diminutive, small, intimate or local [27].

Human communities in particular are distinguished by a common sense of intent, belief, resources, preferences, needs and risks which shape the community identity and the degree of participants' adhesion to the community. Thus human communities are marked by three key characteristics: sense of connectedness and espousal of common values on the basis of freedom and security, monetary exchange and local scale [27]. Especially the principles of tolerance, reciprocity and trust are particularly important for a community to be sustainable in the long term: 'feelings and ideas are renewed, the heart enlarged, and the understanding developed, only by the reciprocal action of men one upon another' [8, p.515].

With regard to communities and monetary exchange, some communities have developed their own local currencies, such as the "Local Exchange Trading Systems" (LETS)[1] and the Ithaca Hours[2], to provide alternative economic growth and a stronger sense of community.

II. Community Currencies: a critical approach.

At this point the discussion moves on to community currencies and it proposes community currencies as an alternative to economic risk on the ground of the social constructivist nature of money and the role that money plays in community life.

A. Community currencies: history, principles & ground rules

1) History

Community appeared in written English in 1283 and it is referred to the material organisation of a self-sufficient economic entity [16, p.10]. On the other hand, local currency is a currency not backed by a national government and not necessarily by a legal entity, which is intended to support trade in small areas or communities. Local currency can take various forms, physical and financial, and it is associated with particular needs and purposes of the area or community where it is issued and used [28].

Community currencies can be looked at in parallel to the notion of 'commons' [3], whereas their economic advantages rely on the sufficiency of currency supply, on use of currency within the community and on the issuance of currency by the community members [18]. In that way, the community members produce and trade their products or services within the community, decreasing the degree of the community's dependence on external resources.

The prompter of the contemporary community currencies can be traced back in the 19th ctn. in Silvio Gesell's vision of money as a service to the community and the members of that community through releasing money from interest [26].

Free banking was the economic prototype of local currencies, whereas nowadays company scrip[3] to pay workers and tokens issued by business to encourage consumer loyalty are the most common forms of local currencies. Some of the oldest known currencies to be in continuous use are the WIR in Switzerland, and the Labor Banks in Japan. The reality shows that community currencies become increasingly popular with over 2500 community currencies in 2000 [17, p.159], whereas there are, broadly speaking, three archetypes of community currencies: the Backed Currency, the Fiat Currency (e.g. Ithaca-hours system), and the Mutual Credit System Currency (e.g. LETS) [20, pp.3-8].

Although community currencies have tight bonds with the old, community-based, tradition of the barter system [7, p.10] where two parties exchange goods or services on the basis of reciprocity and proximity, community currencies attempt to go beyond developing exchange of 'alternative currencies' (alternative notes, tokens, vouchers, etc), goods or services and skills among multiple parties (individuals usually) participating. Thus community currencies broaden the scope of exchange both in terms of parties involved and goods exchanged.

Overall, phases of crisis and economic depression lead to the issuance and use of emergency currencies, giving thus the opportunity for countering the general phenomenon of money scarcity. Local currencies achieved the initial boom during the economic depression of the 1930s, indicating thus their importance for economic sustainability within localities particularly in periods of money scarcity and poverty. For instance, the LETS system appeared and spread in different localities of the globe in the financially difficult years of the 1980s and early 1990s. Also, during the Argentine economic crisis of 2002, small-denomination interest-free provincial bond IOUs were issued by local governments successfully.

[1] Local Exchange Trading Systems were first developed by Michael Linton, in Courtenay, BC [15].

[2] The Ithaca Hours system, developed by Paul Glover, is outlined in 'Creating Community Economics with Local Currency' [10].

[3] 'Scrip' in this context means something like 'emergency currency', namely currency/notes that companies create when they face severe scarcity of regular currencies.

Historical examples of local currencies over the last two decades are the above mentioned LETS, set up in Canada by Michael Linton in 1983, as well as the 'Ithaca Hours' paper currency initiated by Paul Glover in Ithaca, New York in the 1990s[4]. Although local currency systems arrived in Europe quite later, currently there are more than 124 systems in France, 100 in Italy, 65 in the Netherlands and many other in other European countries and regions. Within the European context, though, the UK and the Republic of Ireland have been the two regions where local currencies have met the greatest development with more than 450 groups being established so far [7, pp.14-5].

2) Principles and ground rules

The following principles and rules constitute key characteristics of community currencies, marking thus community currencies' appearance, growth and sustainability.

- **Accountability and trust. Regulation of accountability and trust?**

Trust and accountability (transparency) are fundamental pre-requisites for community currencies' sustainability and empowerment. Usually, no strict regulation and only a members' agreement emerge in a local currency context. People base thus communication and trading on mutuality, trust and embracement of the same values, with no need for formal regulation and mandatory rules.

More specifically, trust and accountability in community currencies are founded and operated on the basis of the following principles [15]:

- Co-operation and mutual responsibility for the integrity and stability of the network, as none owns the network;
- Self-regulation with the user defining the community rules and regulations;
- Empowerment, as all users are entitled to issuing the currency;
- Money, as a means of exchange that can take various formats and can be used for satisfying various needs.

Hence and in order for community currencies to be sustainable and to operate in tune with the above principles, the following ground rules are required [15]:

- Consent among participants;
- No interest charged on account balances;
- Common ownership;
- Disclosure of information, so that informed action is taken;
- Money issued and spent within the community;
- Money with no intrinsic value or material identity; value is attributed only when money is spent;
- All money in the community equals zero and it is issued only when it is needed - neither shortage nor surplus of money occur.

However, there are instances where the absence of trust among participants and under the absence of appropriate regulation has forced local communities to come to an end. Indicative is the case of the Belfast community in Northern Ireland that survived for only three years, as conflicts and lack of trust within the community comprised a huge barrier to sustainability and further establishment [7, p.43]. This raises in turn the issue of regulation, since in offline community currencies only flexible and non-institutionalised frameworks of rules in the format of agreement among community members apply.

- **Local scale, diverse and small-sized membership**

Local scope is another parameter of development for community currencies. The whole idea of community currencies aims to enhance the sense of proximity and the activities within local settings of life, so that local economy and community bonds are empowered. Also, on the grounds of locality, community currencies aim to benefit people who are socially disadvantaged through creating a sound and solid community framework. This is why communities that have attempted to expand beyond local boundaries have been facing a lot of difficulties in terms of administration and transportation, such as communities in Northern Ireland [7, p.45].

In terms of scale and membership size, community currencies are diverse, whereas small scale and membership size are considered the most common type, constituting both a strength and weakness of what community currency offers and how it can be sustained and expanded. There are certain criteria influencing the characteristic of scale, such as: 'diversity, relations of proximity, and relations of affinity' [6]. These three principles point to the parameters influencing the workability and viability of a community currency, determining at the same time its scale and overall character. In that sense, diversity of skills and capabilities, proximity of collaboration and personal contact, as well as affinity of interests, identity and reciprocity show in the direction of trust and accountability in the trading behaviour within the community. From that perspective, trust, accountability, as well as

[4] Since 1991, 2000 people are involved and $2 million has been traded.

predictability constitute key principles in every community currency and as it is briefly discussed below.

Thus on the basis of the above criteria and in order for proximity and affinity to be maintained, community currencies are of a limited geographical scale but with diverse small-sized membership. This in turn poses questions about prospects of expansion, speed of development and sustainability, as there are other views that consider local scope to be a limitation that makes community currencies operate in a mostly complementary way, with no large variety of goods and services being traded. According to those views, expansion beyond local scope and inter-trading 'is an obvious way of providing access to a wider range of foods and services' [7, p.55].

- **Alternative life style, ideology and trading**

Overall, people's traditions, life-style and politics have proved to be of particular importance when a local currency system is to be established. By drawing on the case of Stroud, which has been the first local currency in the UK established in the 1990,we can see that it has had a long tradition of green politics and artistic activity [7, p.18], facilitating thus the employment of an alternative economic system.

By drawing in general on case studies of how LETS system applied to the UK[5], the local currencies introduced initially a different life style, as:

> Many of these groups were started by people active in environmental politics, the green movement or the Green Party. This was certainly the case in Norwich and in West Wiltshire. Others, notably in Stroud and Totnes, were already involved in an 'alternative' lifestyle, and were therefore perhaps more receptive to an idea such as LETS that stood outside the mainstream economy [7, p.16].

On the basis of these principles and rules, one can realise that community currencies are socially and economically constructed with money being used as a means of strengthening the sense of community while empowering the individual user. Besides, the fact that they are found in developing regions in Asia, Africa and Latin America can fuel discussions about that role. However, the local economies can operate in parallel to the national economic system, as their role is usually complementary to that of the national currency systems.

B. Why community currencies? Community currencies socially constructed & socio-economically significant

In principle, community currencies can make up for the money scarcity in regular currencies, while providing underdeveloped regions with important financial opportunities. Users/businesses of community currencies can benefit as they have the potential to attract costumers, increase case income[6], encourage costumer loyalty, ease cash flow and gain a higher profile within the community [14].

Especially nowadays, the overwhelming presence of globalisation and the elimination of geographic and time distances in every domain of social activity and in economic exchanges in particular have contributed to the establishment of trans-national networks, as well as to the rapid weakening of the traditional community format and spirit. These trends have stimulated the emergence of a new movement of community currencies in the 1980s, which attempted to revive the Middle Ages and the primitive communities of that time. The three key examples of community currencies which have inspired the creation of many more instances of communities are the Local Exchange and Trading System (LETS) in Canada, ITHACA HOURS in the USA, and the service-credit exchange system Time Dollars by Edgar Cahn [6].

These and all other community currencies aim at fulfilling the diverse social needs along with the empowerment of local businesses. An indicative example is the Toronto Dollar project [24] launched in 1999 in Canada, with its organizers promising to strengthen the neighbourhood's identity and autonomy, to promote local trade, and to provide locally-generated funds for charities, thus fighting against poverty and homelessness. These societal goals were achieved through a small difference between the price of selling and buying any product in the community, half of which went to local social service groups. LETS has also developed lately a similar model called Community Way. An offline community currency example that contributed significantly to the development of public works can be found in Curitiba, in Brazil. In the 1980s that poor city's mayor, Jaime Lerner, decided to introduce tokens through which citizens were rewarded for cleaning the city garbage, whereas the purchase of tokens could be afforded through selling the recyclable garbage to recycling companies.

However, what are the comparative advantages of local or complementary community currencies in relation to national regular currencies? While the capitalist mode of production and consumption

[5] The following places where LETS has come into existence in the UK and in Northern Ireland can be mentioned: Stroud, London borough Hounslow, Banbridge in Northern Ireland.

[6] Case income is 'income per business case'.

favours the concentration of economic power in only a few hands[7], the ecological destruction and the weakening of communities, community currencies come to resist on the basis of empowering interdependent communities, facilitating cooperation and establishing equality and democracy among participants. Local community currencies make a significant difference, as they aim not only to retrieve economic equality but also to restore human and social capital on the ground of morality 'by showing how money is backed by our beliefs systems' [4].

The moral aspect of community currencies much relies on the principle of reciprocity and in accordance with the community's needs and desires. Reciprocity and the satisfaction of all participants' needs through interaction, cooperation and real work relieve participants from the stress and risk they undergo in regular currencies where competition and scarcity of resources prevail. Even beyond these effects, some consider that money is not itself the major medium of exchange but rather an economic system of affection and reciprocity [14, p.2]. Thus the principles of mutuality, reciprocity, trust[8] and common interest are strengthened in the interactions developed among community members, communicating the value of currencies on the ground of morality and ethos.

Lietaer [16] proposes community currencies as the tool for resolving three issues of economic, political and societal interest: unemployment, environmental destruction, and community breakdown. What is argued is that community currencies enable people to work on 'alternative' task through exchanging services and locally produced goods, while encouraging recycling. Hence, they happen to be particularly popular among many green and sustainable living groups such as the UK Green party [28].

Lietaer [16] thus approaches community currencies from a social constructivist approach, arguing that economic motivation and activity go hand-in-hand with desirable and ethically correct behaviour. Lietaer [16] maintains that the functionality of community currencies is based, from a moral perspective, on morality and human behaviour and, from a functional point of view, on the money attributes of 'standard of measure' and of 'medium of exchange', whereas other money characteristics dominating national currencies, such as money as 'store of value', 'tool for speculative profit' and 'tool of empire', do not apply in community currencies.

On the other hand, Lietaer [16, p.11] makes clear that community currencies are not typical welfare systems, as they don't transfer any resources but they rather create the resources on their own through work and exchange among participants. Also, they do not create inflation, as they link unused resources to unmet needs and only to such needs. Therefore, co-existence and harmonic operation of national and community currencies are both possible and needed.

According to the following graph, complementary local currencies promote longer-term planning and encourage trade and cooperation, as their money is in sufficient supply and no interest applies. The fact that no interest incurs means that the economy of debt and transfer of wealth to only a few hands is countered and that more equal distribution of wealth is achieved. They are also backed by an external reference – commodity or service – rather than a central authority, being thus less subject to inflation. In that sense, community currencies seem to benefit through fighting against money scarcity, inequality, unemployment and debt:

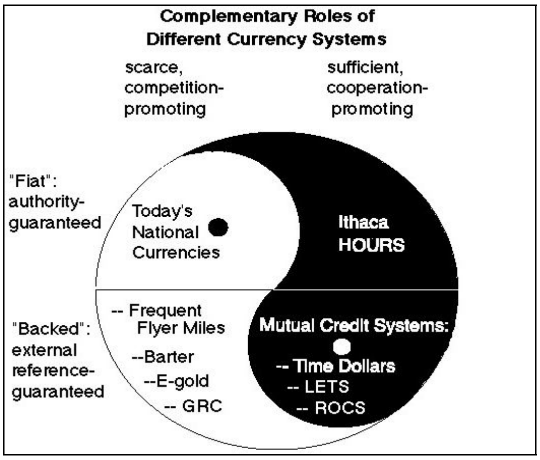


Figure 1. Complementary Community Currency Systems and Local Exchange Networks [25]

What is thus to be proposed is the promotion of local community currencies 'at the grassroots level, along with a communications network' [5].

C. Trust in community currencies: benefit & presupposition

To the question why there is a need for community currencies and what this alternative 'money' has to offer in juxtaposition to regular currencies, trust comes to the fore as a strong answer, as well as a prerequisite for community currencies to be established and maintained.

[7] In the article 'Money Versus Wealth', Korten remarks that '…450 billionaires have combined financial assets greater than the combined annual incomes of half of humanity' [14, p.1].

[8] The importance of trust in general and whether the creation of open commons with access granted to everyone in particular can put trust at risk are to be further discussed later on.

1) The ground for trust in community currencies

The following argument encapsulates only some of the key comparative advantages, with 'trust' being placed at the centre:

> Some communities are coming up with an abundance of creative alternatives to conventional money, in the form of local money systems and complementary currencies, based on **trust, time, and the wealth of energy and skills in communities.** This is a form of 'money' that makes a change, and that offers new opportunities for sustainable development, North and South (emphasis added) [21 p.60].

The fact that community currencies are not used for storage or profit but only as a means of exchange and a standard of value, communicates a currency value that discourages dishonest behaviour between participants and underpins trust through morality. In community currencies, it is not money that values people, but people value each other by using money that is not scarce [15].

More specifically, there are a series of forces facilitating trust in community currencies, such as scarcity, accountability, scale and structure:

- Scarcity: regular money is a scarce currency and the aim to create money that you can trust in completely is a very, very expensive tool to use if you already trust the people you trade with. Trust and scarcity intersect, and local currency retains value from day to day while not being scarce. However, scarcity will always be relative, since the value of money always adjusts to the relationship between money supply and available products and services [29].
- Accountability: trust is a term that becomes entangled with accountability. Accountability can be more easily ensured at local scale, for example in a community where the chain of responsibility is more transparent. So, if the community enjoys some level of prosperity, the participants are inclined to trust that products are not flowing out of sweatshops [29].
- Scale: trust also depends on scale and the community currency system in Argentina is an example. The community currency system in Argentina initially was an answer to the scarcity of global bucks and gained trust because the big money interests were not trusted. However, when the system became so large that the RT turned in a national currency, then accountability was lost and the system began to collapse due to the decreasing amount of trust in the system [29].
- Structure: The structure of community currencies allows for more trust to be established. Community currencies are usually issued by a non-profit organisation whose members are chosen from the community. In addition, there is no interest paid to those people, so the community will not be indebted to them, whereas the real labour of the community members who are willing to accept the currency empowers the community and the trust among members [19].

2) Trust in community currencies: requirement & benefit

With respect to the argued benefits that community currencies bring up, those currencies contribute to: '…local regeneration, build social capital and foster sustainable development' [22], p.56]. In order for local currencies to provide those benefits, trust comes to the fore as a requirement [12], as well as one of the most significant benefits when using local money. Community currencies contribute significantly to the establishment of trust among participants exchanging goods or services through cooperation, reciprocity and satisfaction of needs and desires [14. p.2]. Thus community currencies have through trust the potential to retrieve the social, human, natural, institutional and business capital being lost in today's world economy [14, pp.3-4].

Community currencies are considered almost 'anti-money' [2]. This money is regarded as a practical source of trust, with trust relationships being restored and validated in transactions. Arguably, this currency value is what creates communities, and communities are what establish and empower trust in transactions and beyond transactions and through alternative money [2].

The alternative nature of community currencies is additionally argued by critics of regular currencies who emphasize the importance of trust, as well as the crises that government-issued money are subject to, making thus people loose their trust [23].

Through bringing to the fore more specific examples, Paul Glover talks about the way in which trust and human contact has made the Ithaca's Hours one of the currently most successful community currencies:

> We printed our own money because we watched Federal dollars come to town, shake a few hands, then leave to buy rainforest lumber and fight wars. Ithaca's HOURS, by contrast, stay in our region to help us hire each other…HOURS reinforce community trading and expand commerce which is more accountable to our concerns for ecology and social justice…Residents are proud of income gained by doing work they enjoy. We encounter each other as fellow Ithacans, rather than as winners and losers scrambling for dollars. The Success Stories of 300 participants published so far testify to the acts of generosity and community that our system prompts. We're making a community while making a living…Yet our greater self-reliance, rather than isolating Ithaca, gives us more potential to reach outward with ecological export industry…Ithaca's money honors local features we respect, like native flowers, powerful waterfalls, crafts, farms and our children…Local currency is also lots of work and responsibility [9].

On the basis of the above statement, social and environmental responsibility, community trading

and commerce, marketplace time and skills not employed by the conventional market, income gained by doing work that people enjoy, generosity and community building, self-reliance, ecological industry, money that honors local features and it is fun and legal are only some of the characteristics and principles that make community currencies an important social and economic addition, establishing trust and morality among participants.

Another example of community currencies where trust is guaranteed is the WAT system used in Japan. The WAT system allows for participants to issue their own tickets (IOUs), which are used as a complementary currency in the community on the basis of self-guaranteed trust. Trust in a community currency is based on the community itself and self-regulation has made the WAT community one of the most trustful and successful among small, well-established businesses in Japan [1].

D. Risks and challenges: community currencies and sustainability

However, community currencies are encountered with a range of internal and external risks and challenges that dispute the language of trust and reciprocity that community currencies communicate, deserving thus further attention.

More specifically, funding and organisation issues set serious challenges and risks against community currencies' viability and long-term sustainability. With respect to funding, the necessity for sponsorship constitutes a key issue indicating the economic fragility of community currencies:

> The burden is increased by lack of reliable funds to pay operating costs, let alone staff salaries. Some systems have paid organizers with their own currency, but typically this is a short-term strategy, as spending money freely into circulation can be inflationary. Many turn to grant funding for support, but this requires tax-exempt status and specialized talent, and, at best, usually brings small returns. Donations, another source of funds, must be solicited. Dues are a more reliable wellspring, however, they must be kept modest enough to allow for broader initial appeal and to include lower-income members of the community. The regular collection of dues is costly and time-intensive and a renewal rate of more than 50% is uncommon [6].

At the same time, the division of labour and the overall organisation of community currencies are most of the times a struggle, as well as a pre-supposition for the existence and sustainability of those communities:

> Organizer burnout is a risk because, unlike traditional issue-oriented organizing, a wide range of skills is necessary and a long learning curve is required. Creating and growing a grassroots economy from scratch is an enormous commitment that does not always lend itself to simple delegation and job-sharing. Work at the start-up stage tends to be distributed more evenly among members, but as time passes, becomes more the labor of a few individuals [6].

In addition, transaction costs and the 'risk inherent in exchange rate fluctuations between the currencies' [20, p.24] raise the question of why people or businesses should engage with alternative money when they are simultaneously involved in regular currencies. A fairly realistic response to that challenge could rely on the fact that regardless of the additional transaction costs and the risks deriving from fluctuations between the currencies:

> Considering the benefits associated with accepting both currencies in terms of the extra business or the ability to market excess capacity will be willing to accept a well-managed complementary community currency. Moreover, once implemented, such a [currency] is robust to fluctuations of the regional economy, given sufficient system credibility and a capacity to manage the money supply…a boom merely reduces trade volumes rather than causing the system to collapse [20, p.31].

On the other hand, common criticisms regarding black economy and tax evasion through transactions with alternative currencies shape the debate between opponents and proponents of community currencies [20, pp.1-2].

Finally, another challenge relates to incentives and the loose rules characterising a community currency. This criticism especially regards the provision rules and who is going to provide, maintain and restore the community's resource system, as well as the appropriation rules and the aim of optimum appropriation, exchange and 'withdrawal' [20, p.36]. Although both categories of rules are set within and from the community as a whole, it has to be acknowledged that their workability depends on the preconditions of reciprocity, responsibility and trust, as community currencies' economic bonds are fundamentally based on social norms, voluntary conformity, moral affinities and democratic mediation.

III. Concluding remarks and community currencies in digital ecosystems

In concluding, recursive, reflexive and self-organising community building through the symbolic meanings, power relations and benefits that community currencies bring about are at the epicentre of the research on the employment of community currencies.

Social life is based on the principles of morality and ethos, and community currencies have a role to play, so as responsibility, responsiveness and sustainability to be achieved. For the purpose of the paper, trust is considered to be of extreme importance, being positioned at the top of the pyramid of morals and principles that community currencies enable and in order for social life to be sustainable in the long-term.

However, at stake there are semantic, pragmatic and formal complexities and the research has to

deal with them if the aim is to model community currencies in digital ecosystem context. Thus parameters that challenge the idea of community currencies in digital ecosystems, such as trust, reciprocity and power relations in online business environments, are deeply involved in the alternative of community currencies in a digital ecosystem.

IV. Bibliography

[1] Andron, M. and B. Lietaer (2006), Complementary Currency Innovation: Self-guarantee in peer-to-peer currencies' in *International Journal of Community Currency Research*. 10: 1-7

[2] Bakshi, R. (2001) 'Community currencies', *The Hindu. Online editions of India's National Newspaper*. 30 September 2001. Available at: http://www.hinduonnet.com/2001/09/30/stories/1330046b.html [Accessed 30/12/2006]

[3] Benkler, Y. (2006). *The Wealth of Networks: How Social Production Transforms Markets and Freedom*. New Haven, CT: Yale University Press.

[4] Brouillet, C. *The Difference Between Money and Real Wealth.* Available at: www.communitycurrency.org/ [Accessed 30/11/2006]

[5] Brouillet, C. *Reinventing Money, Restoring the Earth, Reweaving the Web of Life.* Available at: http://www.communitycurrency.org/reweaveWeb.html [Accessed 30/11/2006]

[6] Cohen-Mitchell, T. 'Community Currencies at a Crossroads: New Ways Forward'. *Community Scale Economics*, Issue 2. Available at: http://www.newvillage.net/Journal/Issue2/2commcurrencies.html [Accessed 30/11/2006]

[7] Croall, J. (1997). *Lets act locally: the growth of local exchange trading systems.* London: Calouste Gulbenkian Foundation

[8] De Tocqueville, A. (1994) *Democracy in America*, London: Fontana Press.

[9] Glover, P. (1998) 'Local Currency: In Each Other We Trust Creating community economics with local currency', *Whole Earth Spring 1998.* Available at: http://wholeearth.com/ArticleBin/87.html [Accessed 18/11/2006]

[10] Glover, P. *Creating Community Economics with Local Currency.* Available at: http://www.ithacahours.com/intro.html [Accessed 18/11/2006]

[11] Harper, D. (2001) *Online Etymology Dictionary.* Available at: http://www.etymonline.com/index.php?term=community [Accessed 15/06/2007]

[12] Henderson, H. (1994). Information: The world's new currency isn't scarce. World Business Academy Perspectives, Fall.

[13] Interview with Bernard Lietaer, *Beyond Greed & Scarcity*. Available at http://www.ratical.org/many_worlds/cc/Lietaer.html

[14] Korten, D.C. (1997) 'Money Versus Wealth' in *YES Magazine*. Summer 1997 Issue: Money: Print your Own! Available at: http://www.yesmagazine.org/article.asp?ID=885 [Accessed 15/05/2007]

[15] LETSystems – new money. Available at: http://www.gmlets.u-net.com/ [Accessed 15/05/2007]

[16] Lietaer, A.B. (1994) *Community Currencies: A New Tool for the 21st Century.* Available at: http://www.ratical.org/many_worlds/cc/CC.html [Accessed 30/11/2006]

[17] Lietaer, A.B. (2001) *The Future of Money: Creating New Wealth, Work and a Wiser World,* Century: The Random House Group Limited, London

[18] Linton, M and Yacub, E. (2006) *Open Money.* Available at: http://subsol.c3.hu/subsol_2/contributors0/lintontext.html [Accessed 30/11/2006]

[19] Powell, J. and Salverda, M. (1998) *Community Currencies—An Innovative Tool to Promote Economic Self-Reliance.* Available at: http://www.appropriate-economics.org/asia/thailand/tdsc.html [Accessed 13/11/2006]

[20] Schraven, J. (2001) *The Economics of Community Currencies: a Theoretical Perspective*, Unpublished Honours Thesis, Oxford University

[21] Seyfang, G (2001) 'Money that makes a change: community currencies, North and South' *Gender and Development,* 9 (1): 60-69.

[22] Seyfang, G. & Pearson, R. (2000) 'Time for Change: International Experience in Community Currencies' *Development*, 43: 56-60

[23] Tangwisutijit, N. and Rojanaphruk, P. (2000) 'Community currency system poses ''no threat to baht'', *The Nation.* 04 June 2000. Available at: http://www.appropriate-

economics.org/asia/thailand/nationapril26.html [Accessed 15/01/2007]

[24] Toronto Dollar project. Available at: www.web.net/~tordoll/ [Accessed 15/11/2006]

[25] Transaction.Net, *Complementary Community Currency Systems and Local Exchange Networks.* Available at: http://www.transaction.net/money/community/ [Accessed 15/11/2006]

[26] Van Arkel, H. and G. Peterse (1998). *Money and its alternatives: How money controls the world and the alternatives to change it.* Utrecht: Stichting Aktie Strohalm.

[27] Wikipedia, *Community*. Available at: http://en.wikipedia.org/wiki/Community [Accessed 15/11/2006]

[28] Wikipedia, *Local currency*. Available at: http://www.wikipedia.org/wiki/Local_currency [Accessed 15/11/2006]

[29] http://burlingtoncurrency.blogspot.com/2006/04/ripple-pay.html [Accessed 30/11/2006]

[30] http://www.openmoney.org [Accessed 30/11/2006]

A Framework for Digital Ecosystems:

The 'First' of Theoretical Interpolations

Frauke Zeller
Institute of Media and Communication Studies
Ilmenau University of Technology
Ilmenau, Germany
frauke.zeller@tu-ilmenau.de

Paolo Dini
Department of Media and Communications
London School of Economics and Political Science
United Kingdom
p.dini@lse.ac.uk

Abstract

In order to arrive at a solid foundation for a Digital Ecosystem (DE) vision that could provide the necessary research scope as well as political credentials, the need for a meaningful and useful theoretical framework has grown since the first introduction of the DE concept and gained importance ever since. A seemingly fitting tool for such a theoretical framework is represented by the Networks of Excellence (NoE), a specific project type funded by the European Commission which focuses on research outcomes that feed into industrial development rather than merely concentrating on the 'manufacturing of knowledge' (i.e. industrial development). In this article we will embark on this challenge by means of a two-fold approach: at first, we will discuss why communication theories represent an approach that carries (historically and paradigmatically) the potential of depicting DE's main concepts in a unifying framework that connects to all disciplines involved in DE research; secondly, we will depict our on-going work on the theoretical framework of DEs in the NoE project OPAALS.

Digital Ecosystems; Language; Autopoiesis

I. Introduction

The need for a more systematic approach to information society development to support economic development on a European scale has led (among other factors, see Dini et al., 2005, Nachira et al., 2002) to the Digital Ecosystem concept/vision (for a more exhaustive discussion and description of the DE concept see Dini and Nachira, 2007 and Nachira et al., 2007). The research efforts in this area were initially triggered by the initiative Go Digital, aimed at boosting ICT adoption by European SMEs.

The DE concept strives to support sustainable regional socio-economic development and therefore has to integrate economic, political, and socio-cultural factors that correspond to specific regional demands of its stakeholders as well as integrate themselves symbiotically into the broader DE. Consequently, is was soon discovered that in order to "bring into existence information and communication technologies (ICTs) that help in the achievement of the challenges identified by the objectives of the Council of Lisbon (higher growth, more and better jobs, and greater social inclusion [COM 2004]) we needed to widen our horizons with a more holistic and systemic approach. In addition to ICT, this new approach should consider socio-economic aspects and the human perception, communication and representation dimensions in one single research domain" (ibid).

II. The Digital Construction of Reality

The DE concept carries the notion of an evolutionary digital reality, which means that it does not only strive to be adaptable for all stakeholders but also dynamically developed by its stakeholders through a socio-constructivist process. Moreover, 'digital' contains another conundrum as it refers to the aim of boosting ICT adoption and development on a regional scale and, at the same time, attempts to reach this aim 'by means of' a digital environment. Hence, we are dealing with a complex system that is capable of switching back and forth between the individual stakeholders' reality on a regional scale and the overall digital reality of the DE, which is being shaped by all stakeholders. This means that we cannot focus on the development of a DE that is merely based on a reductionist binary system, but a *digital* ecosystem that integrates a concept of evolution that is grounded in socio-cultural factors which are based on human perception, communication, and representation dimensions.

In social science studies of technology the dichotomy between socio-centric and techno-centric approaches is still existent. Whereas in the socio-centric approach the social context and environment of the different actors plays an important role, the techno-centric approach tends to diminish this part to a merely computable, binary, and logical phenomenon by focusing on development of an intelligent and adaptable information system apparatus.

In digital ecosystems research, likewise, we might be tempted to assume that a similar dichotomy applies between the reality constructed at the regional scale through social processes and the general "digital reality" of the digital ecosystem. The architectural principles of the DE, however,

aim to redress this tension. In technical terminology, the use of the term "evolutionary" is analogous to "stateless": the structure and behaviour of the digital infrastructure and of the services and digital entities that populate it are not predetermined (deterministic, or stateful), but respond and adapt to the local context dynamically. From a social science point of view this sounds like a relatively small improvement, basically we are moving from a static to a dynamic model. If we however stress the fact that the dynamic architecture of a DE is also *open*, then the context-dependence becomes more believable. As a consequence, digital ecosystems are not expected to be all alike for any geographical region, industry sector, or virtual community. Although they will all share the same architectural principles, the individual realisations will be very much dependent on the context. We can therefore see how the DE has the potential to achieve all the properties summarised in the first paragraph of this section, which aims to integrate the well known mutual dependence between technology and its users with a systemic and biologically-inspired perspective. As the former comes from a social science tradition of studies of technology, whereas the latter is traceable from the joining of computer science and artificial intelligence with biology and physics, we can see that DE research has the potential to construct compelling disciplinary bridges at the theoretical level and useful apparatus at the applied level. In essence, DE research aims to advance the state of the art on a long-standing problem in technology adoption, and that is the formalisation of knowledge. By adopting a systemic viewpoint, it aims to integrate all the aspects of a digital ecosystem into the single metaphor/model of an associative autopoietic digital ecosystem. The following paragraphs begin to outline how this might be achieved through a language and communications perspective.

The OPAALS project aims to apply the conceptual apparatus of autopoiesis for an underlying theoretical DE framework as it corresponds to the aims of a self-sustaining, evolutionary digital ecosystem. Autopoiesis is grounded in biology, but its appeal reaches well beyond biology. If we imagine autopoiesis to embody exclusively principles of biological organisation, then its application to social systems seems undesirably reductionist. It would exclude, besides the analysis of socio-cultural and socio-economic factors which are elementary to a successful DE concept, the individual stakeholders' level regarding their perception of reality and representation dimensions. Hence, for any serious attempt of a theoretical framework for DEs we need to integrate the communicative paradigm as the redeeming feature of this challenge.

The communicative paradigm depicts a substantial shift in social theory and sociological methodology. According to Luckmann (2005), "social reality is constructed in communicative interaction, and if it is pervasive in social life, our most reliable knowledge of that reality will come from reconstructions of these processes". The communicative paradigm therefore provides us with the possibility of amplifying the conceptual apparatus of autopoiesis. That is to say, by reconstructing the processes of communicative interaction we understand the construction processes of social reality on a regional and digital scale. Obviously, the digital scale does not only consist of network architectures (from a computer science perspective), but it is shaped by the joint usage of individual stakeholders, which is again grounded in communication.

III. Autopoiesis and Semiotics

We already stated above that the conceptual apparatus of autopoiesis reaches well beyond biology and argued that an amplified concept is needed in order to arrive at a meaningful and useful underlying theoretical framework for DEs. The acknowledgement of the communicative paradigm then leads us to the field of semiotics, the theory of signs and meaning.

As in almost all scientific fields that are shaped by dialogue rather than canonical thinking, there are different assumptions in the linguistic domain regarding meaning, representation, and reality. For example, the assumption that linguistic signs represent (denote) different aspects of reality significantly shapes the conceptual framework of traditional linguistics. In this current of thought, representation (denotation) is regarded as the basis for understanding in human communication insofar as language expresses messages, which are "mental representations in the form of conceptual structure," (Pinker and Jackendoff, 2005, cited in Kravchenko, 2007). However, Kravchenko highlights the contradiction that "the representational theory of mind built largely on this notion and implying a kind of non-arbitrary (i.e., computationally definable) relationship between the sign and what it stands for has been unable to facilitate advances in areas where applications of the coded equivalence principle should do the job, such as machine translation or Artificial Intelligence". Therefore, for want of a more satisfying theory of mind and human communication, researchers then turned to a bio-cognitive and/or biosemiotic approach that integrates Maturana's notion of linguistic behaviour: "[D]enotation is not a primitive operation, it requires agreement consensus for the specification of the denotant and the denoted. If denotation is not a primitive operation, it cannot be a primitive linguistic operation, either. Language must arise as a result of something else that does not require denotation for its establishment, but that gives rise to language with all its implications as a trivial necessary result. This fundamental process is ontogenetic structural coupling which results in the establishment of a consensual domain. [...] Linguistic behavior is behavior in a consensual domain" (Maturana, 1978).

Biosemiotics is the scientific study of signs and semiosis in living systems (Uexküll, 1957/1934; Seboek, 1976). It regards life and semiosis as coexisting and overcomes on the one hand the "pure chemical description of life in molecular biology and, on the other hand the traditional idea that semiotics is only the study of signs in the language and culture [of] human beings" (Brier, 2006). Biosemiotics focuses specifically on the Peircian semiotic paradigm (Peirce, 1931-1966; Peirce, 1992), which is considered as a transdisciplinary paradigm (Deely, 1990) as it aims at a unified theory of nature, cognition, and mind. The Peircian semiotic paradigm is also used in computer semiotics, which deals with the application of the sign system to an algorithmic sign concept and therefore tries to provide a

theoretical framework for natural and formal sign processes (see Anderson, 1997).

In addition to the semiotic tool-set, biosemiotics incorporates the concept of autopoiesis and also deals with evolutionary systems theory. It therefore represents a promising and useful point of departure for the development of a theoretical framework of DEs. Emmeche (2000) states four different integrative underlying concepts regarding evolutionary systems theory:

(a) self-organisation (or emergence, autopoiesis, autocatalysis)
(b) evolution (or development)
(c) communication (or semiosis, information processing)
(d) living (or feeling, acting, learning)

Emmeche (2000) combines Peirce's notion of a sign with Bateson's notion of information in order to stress the relational character of a biosemiotic process: "Life entails semiosis as the action of signs, where a sign is a **first** [emphasis in the original], i.e., a representamen that stands (by a code or a habit) in such a relation to a **second**, its object, so as to determine a **third**, its interpretant, to take the same relation to that object (that the representamen takes) and thereby effecting that interpretant so that this effect is significant (potentially or actual) to that interpretant's interpreter organism, in the sense that it is a difference that makes a difference to the interpreter".

According to this definition, however, an interpreter must be an organism or part of an organism. This hints towards the notion that semiosis (as well as autopoiesis) is only possible in living systems and to the unavoidable question of what machines, like computers, are processing when no humans are interpreting it: signs or signals? According to Brier (2006) we codify signals in order to carry meaning in a specific context and therefore become signs to human perception. Nevertheless, a DE would rely to a great extent on a sound technological backbone, that is to say a machine. This means that, strictly speaking, the development of a self-evolving, self-sustaining, autopoietic system based on algorithms and circuits would be a contradiction in itself.

Again, the Peircian concept of semiotics allows us to extend the range of objects that are capable of semiosis and autopoiesis by means of the concept of the quasi-sign: "The term quasi-sign suggests an answer to the question whether there can be semiosis in a machine of the kind which Peirce knew. A quasi-sign is only in certain respects like a sign, but it does not fulfill all criteria of semiosis. While some criteria of semiosis may be present in machines, others are missing," Nöth (2002). In short, machines are indeed capable of symbol processing, however they lack the 'window to the world', which means that they cannot relate the sign to an object of experience. Nöth (2002) summarises two kinds of messages produced by a computer in the interface between humans and machines: (a) messages conveyed by a human sender and mediated by the computer, and (b) quasi-signs resulting from an automatic and deterministic extension of human semiosis (see also Krämer, 1988).

IV.A Roadmap

In the search for a meaningful and useful theoretical framework for digital ecosystems, a framework that will help us answer questions about what characteristics a digital ecosystem should have in order to support and catalyse socio-economic development, we encountered the concept of autopoiesis.

So far we have argued the need for an extended concept of autopoiesis that integrates language/communications as a reality-determining concept. We have also discussed different approaches to (sign) systems from a linguistic stance and come to the concept of biosemiotics, which is being amended by Brier (2006), for example, into the concept of cybersemiotics that attempts to place the mechanical, living, and conscious system in relation to each other and how to fit this into an evolutionary framework of science. These attempts certainly promise further theoretical underpinnings for our underlying DE framework.

This short article can only represent *first*, and according to Peirce's notion of 'First',[1] preliminary theoretical interpolations of not yet fully connected ideas and concepts into the direction of the theoretical DE framework. In our ongoing work in OPAALS we will then pursue the aforementioned and additional fields according to the following 'roadmap':

- We will argue that autopoiesis is the beginning of a theory that itself uses biology as metaphor and that is concerned with much more abstract properties of systems. In light of the strongly relativist epistemology of autopoiesis, if we accept that a sociology of regulation grounded in the systemic point of view can only account for a subset of social, economic, and political phenomena, we might consider the possibility of a genuine overlap between associative and autopoietic systems. We will argue that this overlap is best explained through a language lens (i.e., referring to the communicative paradigm). Furthermore, we will also show how discourse organisation can lead to conclusions about the structure and dynamics of social systems that resonate strongly with autopoietic organisation.

- After an initial summary of the main points of the literature on second-order cybernetics and systems theory applied to different disciplinary domains, we will explore the topology of the relationship between associative and autopoietic systems from a communication theory viewpoint, and will show how it can be seen as a symptom of a wider epistemological tension between the naturalist and hermeneutic philosophical traditions. Our excursion will enable us to appreciate better the subtlety of the postmodern claim that 'everything is text', where by 'text' we mean a purely relativist social construction. Enlightened by our greater awareness

[1] The subtitle of this paper "The 'First' of Theoretical Interpolations also refers to Peirce's categories Firstness, Secondness, Thirdness.

of the universality of language, we will argue how the most naturalist philosophy can hope for is a 50% share of Truth in the case of mathematical structures, since they still require to be expressed and communicated through language. But that this share of Truth is indeed a great deal more than we dared hope for.

• Armed with a renewed philosophical optimism for a constructive interaction between structure and context, we will attempt to trace a path that from social theory and through linguistics arrives at a loose specification of formal language systems with heightened self-organising properties. We will attempt to further prove how language is intrinsically autopoietic because we wish to extend its properties of organisational closure to language and knowledge communities mediated by digital ecosystems, in order to arrive at an autopoietic formal language architecture within an associative natural language context.

• Finally, as a recurrent loop, we will address the more open-ended questions of whether the algebraic structures of cell biology embody the blueprint for autopoietic behaviour and whether the isomorphisms between algebra and logic can mediate the transfer of such behaviour to formal language systems in the form of constraints, rules, and 'laws' of digital ecosystems animated by their structural coupling with the language and knowledge communities of their users.

References

P B Anderson (1997). A theory of computer semiotics. Cambridge University Press.

S Brier (2006). "The cybersemiotic model of communication: An evolutionary model of the threshold between semiosis and informational exchange," in Semiotica, 158-1/4, 255-296.

J Deely (1990). Basics of semiotics. Bloomington: Indiana University Press.

P Dini et al. (2005). "The Digital Ecosystem Research Vision: 2010 and beyond," position paper following the cycle of workshops 2005 and the on-line debate among the digital ecosystem community, http:www.digital-ecosystems.org/events/2005.05/de_position_paper_vf.pdf [15.10.2007].

P Dini, and F. Nachira (2007). "The paradigm of structural coupling in digital ecosystems," in The Digital Business Ecosystem, A. Corallo, G. Passiante and A. Prencipe, Eds. Edward Elgar Publishing.

C Emmeche (2000). "Closure, function, emergence, semiosis and life: The same idea? Reflections on the concrete and the abstract in theoretical biology," in Closure: Emergent Organizations and Their Dynamics. J. L. R. Chandler and G. Van de Vijver, Eds.

S Krämer (1988). Symbolische Maschinen. Darmstadt: Wiss. Buchgesellschaft.

A V Kravchenko (2007). "Essential properties of language, or, why language is not a code," in Language Sciences, 29/5, 650-671.

T Luckmann (2005). "On the communicative construction of reality," lecture to the LSE Department of Information Systems 2nd February 2005.

H R Maturana, (1978). "Biology of language: the epistemology of reality," in Psychology and Biology of Language and Thought, G. Miller and E. Lenneberg, Eds. NY: Academic Press.

F Nachira et al. (2002). "Toward a network of digital business ecosystems fostering the local development," http://ec.europa.eu/information_society/topics/ebusiness/godigital/sme_research/index_en.htm [15.10.2007].

F Nachira, A Nicolai, P Dini, L R Lèon and M Le Louarn, Eds. (2007). "Digital Business Ecosystems: The results and the perspectives of the Digital Business Ecosystem research and development activities in FP6".

W Nöth (2002). "Semiotic machines," in Cybernetics and Human Knowing, 9/1, 5-21.

C S Peirce (1931-1966). The collected papers of Charles S. Peirce. 8 vols., C. Hartshorne, P. Weiss and A. W. Burks, Eds. Cambridge: Harvard University Press.

C S Peirce (1992). Essential Peirce: Selected philosophical writings, vol. 1. N. Hauser and C. Kloesel, Eds. Bloomington: Indiana University Press.

S Pinker and R Jackendoff (2005). "The faculty of language: what's special about it?" in Cognition, 95, 201-236.

T Seboek (1976). Constributions in the Doctrine of Signs. Bloomington: Indiana University Press.

J Uexküll (1957/1934). "A stroll thorugh the worls of animals and men. A picture book of invisible worlds," in Instinctive Behavior. The development of a modern concept, C. H. Schiller, Ed. New York: International Universities Press.