OPAALS PROJECT

Contract n° IST-034824

WP3: Autopoietic P2P Networks

Del3.6 – Consensus detailed architecture of the OPAALS DE

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

Contract Number: IST-034824

Project Acronym: OPAALS

Deliverable N°: D3.6

Due date: 31/05/2008

Delivery Date: May 2008

Short Description:

This document is the first in a series of deliverables (D1.2, D12.1, D12.2) that aim to frame the interdisciplinary understanding and consensus in Digital Ecosystems research. In OPAALS the technological and social concerns in providing the necessary digital infrastructure are not treated as distinct but as part of the same continuum. This deliverable sets out the key concepts and characteristics of interest from a computer science and social science point of view. From the social science viewpoint it highlights the issues regarding the policy domain in DEs and suggests areas of theoretical debate that open up. The key aspects of the core DE architecture, in terms of P2P and transaction support, as well as identity and trust, are then described from a computer science viewpoint.

The process of putting the design models and the social theories and arguments in a dialogue is an iterative one and requires an open debate that gradually includes more and more stakeholders. In this deliverable we show how this process has been set up and demonstrate the consensus reached on key aspects of the core DE architecture.

Author: UniS, LSE, WIT, CAM, T6

Partners contributed: ITA, UniKassel, NUIM, CN

Made available to: IITK, UL, BCU

VERSIONING		
VERSION	DATE	NAME, ORGANIZATION
0.1	**6/2008**	**S. MOSCHOYIANNIS (UNIS), M. L. DARKING (LSE), J. STANLEY (CAM), A. RAZAVI (UNIS)**
0.2	**6/2008**	**S. MOSCHOYIANNIS (UNIS), M. L. DARKING (LSE), L. RIVERA LEON (T6), A. PASSANI (T6), J. VAL (ITA), P. MALONE (WIT), M. MCLAUGHLIN (WIT), P. TSATSOU (LSE), A. RAZAVI (UNIS), P. KRAUSE (UNIS)**
0.3	**7/2008**	**S. MOSCHOYIANNIS (UNIS), M. L. DARKING (LSE), P. MALONE (WIT), M. MCLAUGHLIN (WIT), P. TSATSOU (LSE), J. STANLEY (CAM), M. IQANI (LSE) , P. KRAUSE (UNIS)**
0.4	**7/2008**	**S. MOSCHOYIANNIS (UNIS), M. L. DARKING (LSE)**

Quality check

Internal Reviewers: Dr Frauke Zeller (UniKassel), Mr Juanjo Aparicio (TI)

Dependencies:

Work Packages	WP12: Task 12.8 – OS principles of communication and collaboration, Task 12.9, 12.10 – Business models in DEs WP11: Task 11.1 – collaboration and innovation in the Knowledge Economy, Task 11.4 – social innovation networks WP10: Task 10.10 – visualisation of P2P infrastructure WP5: Task 5.6 – adding e-business support in current DBE, Task 5.7 – integration of P2P network services into the DE
Partners	TI, BCU, UniKassel, IPTI, NUIM, UL, IITK, TUT, UNIVDUN
Domains	Computer Science domain: P2P networks, interactions, long-running transactions, distributed systems and networks, redundancy, diversity, consistency, concurrency, design for failure, formal semantics, behaviour patterns, lock mechanisms, formal analysis, distributed identity, distributed trust. Social science domain: participation, power, control, technology infrastructure for competitive advantage, monopoly, lock-in, proprietary software / platforms, knowledge, openness, reciprocity, context-dependent trust, identity. Natural Science domain: simple reference to key analogies with ecosystems in nature and their key features found mostly in studies of biodiversity.
Targets	Computer, social and natural science researchers, SMEs, business analysts. Computer science communities: database, transactions, P2P networking and applications, formal methods. Social science: language, socio-economics, governance, power and control, identity and trust.

Table of Contents

Executive Summary

The purpose of this deliverable is to provide a detailed outline of key aspects of the digital ecosystem core architecture, namely: the distributed transaction model, self-organising P2P network, and trust, accountability and identity models. The aim is to foster an environment for open collaborations which is not dependent on a central point of control and does not suffer from a single point of failure. The proposed dynamic architecture capitalises on the increased diversity inherent in the digital ecosystem in coping with failure and distributes the responsibility of maintenance operations and coordinated interactions between the participating entities.

Due to the distinctive mode of systems development that OPAALS aims to enact - in which technological and social concerns are not treated as distinct, but as part of the same continuum - this deliverable not only considers technological arguments supporting core architecture design decisions it also considers social science dimensions to these arguments. Using this approach, this deliverable reports on a 'consensus view' of what the core architecture aims to achieve and why that is gradually developing via interactions between OPAALS' social and computer scientists.

We start with an introductory chapter that gives the context of this research work and the main objectives we set out to achieve. We also describe what it is we understand by consensus in the context of this report and the process through which computer scientists and social scientists have worked together to generate a consensus view.

In Chapter 2 we describe the key design principles that have driven development of the core architecture, firstly in computer science terms and secondly in social science terms. For the computer scientists this involves a discussion of the 4 key design principles that have informed their development process. These are: no single point of failure; diversity; reliability and distribution. For the social scientists this involves expanding discussion of these principles using social science theory and concepts alongside consideration of empirical data relating to: the current socio-technical environment in which small organisations interact with one another; and potential digital ecosystem users. At the end of this section key questions raised by social science researchers regarding specific operational aspects of the core architecture that have emerged from the consensus building process are noted. These questions will be addressed (be returned to) later in the deliverable, and specifically at the end of each chapter that describes the key elements of the core DE architecture.

From Chapter 3 onwards our focus turns to a detailed outline of the core architecture. We present the model the distributed coordination of long-running transactions and highlight how the design allows for local coordination of the service involved. This ensures that local autonomy of participants is preserved and is achieved by using lightweight log structures given by directed graphs, from which a formal description of the behaviour and service executions patterns should follow in order to guarantee a successful outcome. A component-based design of the local agent to be implemented on each participant is also given. We describe how the required orderings of service invocations between the local coordinators' components of the local agents can be captured in *transaction scripts* and how the corresponding compensating sequences are determined. We also outline an extended lock scheme for handling data dependencies during the execution of long-running transactions and their recovery whenever some failure makes this is necessary.

In Chapter 4 we present the key aspects of the design of the P2P network connecting the participating entities. Our focus is on providing support for transactions realising business activities and our design is targeted at achieving this without the need for intermediation by a (centralised) network operator. We show how the temporary networks formed by long-running transactions between existing partners can be used to boost critical characteristics of the P2P network topology such as connectivity and reliability. The proposed P2P architecture is based around the notion of the *Dynamic Virtual Super Peers* (DVSPs), which are aggregations of nodes performing network operations and whose formation evolves over time to adapt to the usage of the network and reflects the dynamicity of the DE environment.

In Chapter 5 we turn our attention to the issues of identity, accountability and trust. We focus on how entities are identified in the DE and how trust is established between entities prior to transactions. We also explore the links between accountability and trust and the basis for underpinning trust with accountability data. In this chapter we present models for identity, accountability and trust and examine an integrated model that combines all three into an evolutionary trust framework.

Some concluding remarks and directions in which the work and collaborations described in this deliverable can be taken forward are given in Chapter 6. Appendix A includes the full version of the documents that have been used in the empirical social science research drawing data from SMEs. Appendix B includes the full version of a joint article by a number of computer and social scientists in OPAALS which examines the socio-economic perspectives of the technology in digital ecosystems and sets the overall aims and intentions of the DE core architecture.

1. Introduction

The purpose of this deliverable is to provide a detailed outline of key aspects of the digital ecosystem core architecture, namely: the distributed transaction model, self-organising P2P network, and trust, accountability and identity models. The aim is to foster an environment for open collaborations which is not dependent on a central point of control and does not suffer from a single point of failure. The proposed dynamic architecture capitalises on the increased diversity inherent in the digital ecosystem in coping with failure and distributes the responsibility of maintenance operations and coordinated interactions between the participating entities. The key aspects of the core architecture are designed to facilitate cooperation on business and knowledge services and improve connectivity between small-and-medium enterprises (SMEs) in the digital ecosystem.

Due to the distinctive mode of systems development that OPAALS aims to enact - in which technological and social concerns are not treated as distinct, but as part of the same continuum - this deliverable not only considers technological arguments supporting core architecture design decisions it also considers social science dimensions to these arguments. Using this approach, this deliverable reports on a 'consensus view' of what the core architecture aims to achieve and why that is gradually developing via interactions between OPAALS' social and computer scientists. This viewpoint is not simply a synthesis of social and computer science arguments. It constitutes a critical process that involves questioning our respective fundamental assumptions about technology and its ability to configure social relations. By developing an interdisciplinary understanding of what the digital ecosystems architecture aims to achieve we are able to use this as a basis from which to scrutinise component parts of the architecture ensuring that key social science concerns are considered alongside technological decision-making.

Whilst 'consensus view' implies a process where by complete agreement between social and computer scientists was sought, we prefer to speak in terms of an ongoing process of critical dialogue in which informed opinions can and do diverge but within which significant concerns and disagreements are resolved. In this sense, our approach could be viewed as an example of the type of consensus building processes that a governance framework of model would need to encompass. As is the case with multi-stakeholder governance discussions, this intensely interdisciplinary, inter-organisational, inter-professional mode of working is particularly demanding and the efforts required to develop the approach described in this deliverable should not be underestimated. However, such an approach corresponds to the principle of intellectual plurality that underpins the OPAALS academic research community and such challenges are therefore viewed as worthwhile.

In practical terms this viewpoint has been generated via discussions between social and computer scientists. These have taken place via project mailing lists, face-to-face at individual and project meetings, and also through the process of writing this and other project reports. As well as our own research agendas, our discussions have been informed by comments made by the project officer and the review panel from the end of the 1st year review and also by the project management board. The most significant meeting that has taken place with respect to our consensus building process happened in May 2008 at project meeting designed specifically to being social and computer scientists together to discuss the core architecture. At this meeting and in the discussions that followed it several specific questions were asked by the social scientists regarding operational implications of the core architecture. These questions will be noted at the end of each chapter along with references to the particular design aspects of the core architecture where responses to them are provided.

2. Digital Ecosystems: concepts and characteristics

In order for interdisciplinary understanding and consensus to be reached it is important for researchers to first understand the key terms and concepts that each uses in their specialist field of knowledge. In this section we set out the key computer science and social science concepts discussed in this deliverable. We describe the consensus building process as it took place and present some empirical social science research findings that help contribute to an understanding of digital ecosystems requirements from a user point of view.

2.1 The DE initiative from a computer science point of view

In this section, we argue that pursuing the analogy from natural ecosystems to digital ecosystems where the environment is a socio-economic context supported by a suitable digital infrastructure is worthwhile. This line of thinking leads to the identification of the basic or primary characteristics the corresponding digital infrastructure should have.

These pose concrete challenges for computer science, in particular with respect to providing robust support for open and trusted collaborations in distributed transactions and designing the dynamic topology of the underlying P2P network to supports this transactional environment. In what follows we outline the primary characteristics that the digital infrastructure of a DE should have and highlight how these have driven the core aspects of the OPAALS digital ecosystem architecture. The way these concepts have fed in the design models of the core architecture will be described in subsequent chapters.

Ecosystems in nature have always excited interest and been studied for a long time. The British ecologist Arthur Tansley (1871-1955) was among the first people to describe functioning organisms and their physical environment as the "basic units of nature on the face of the Earth" and referred to them by the term "ecosystem". In his 1935 article [Tan35] he refers to an ecosystem as an interactive system that is established between living creatures and the environment in which they live. Members of an ecosystem benefit from each other's participation, even if the benefits are not so obvious in the first instance. Studies of biodiversity indicate that the respective populations in a predator-prey relationship tend towards a stable attractor. Among the main characteristics of natural ecosystems is the absence of a central point of command and control, the increased diversity and the dynamic interrelationship between participants and the environment (of which they are also part of).

In talking about Digital Ecosystems we need to be mindful that the term can mean different things to different people [Din07]. In this chapter we take the view of computer and natural scientists, i.e. make an analogy to an environment defined by a socio-economic context whose members are software artefacts, be they components, applications, information sources or businesses. In light of the ever-increasing complexity of modern software which is often expected to perform previously unrelated functions, and the need for software applications that are highly concurrent and distributed, it seems useful to pursue this analogy further. Some of the characteristics found in natural ecosystems are relevant to the software world and some of the inherent properties exhibited by the respective populations are desirable in various settings.

In a business setting, sustainable economic growth (or survival) is a common denominator for small, medium and large organisations. In a certain important sense, the interest in Europe and the OPAALS project lies with fostering an environment that facilitates cooperation and improves connectivity between small-to-medium enterprises (SMEs), in way that allows them to be (collectively) competitive in an

environment largely dominated by large enterprises. It turns out that some of the characteristics of natural ecosystems that we are looking for in Digital Ecosystems go some way towards fostering an environment where businesses can cooperate and benefit from each other's participation.

In order to achieve sustainable digital business ecosystems, an appropriate software infrastructure for e-business transactions is required, together with new formal and semi-formal languages that enable open and trusted collaborations between small-and-medium enterprises to ensure their sustainability in an (ultimately global) constellation of Digital Ecosystem. This entails a move away from 'traditional' centralised solutions for transaction modelling and support, and towards fully distributed solutions. In distributed solutions the participating (and interacting) entities share the responsibility of command and control in coordinating the interactions in performing a specific task. In other words, all participating entities are equal partners in the logic of the application rather than simply responsive machines or *clients* to the requests of a central authority. This is in contrast to centralised solutions where there is an entity which acts as the central point of command and control (often referred to as the *server*) for all other participating entities (often referred to as *clients*).

Distribution is again an overarching concern when it comes to the architecture of the network that is required to connect the participating entities in a DE. The network itself is considered as an overlay of the Internet, so the basic networking layers of the Internet (such as TCP, HTTP, etc.) shall be used, but in a specific design needed to exhibit certain characteristics for supporting distributed interactions in terms of business (long-running transactions) in the first instance but also knowledge services across the DE network. We have opted for a purely distributed P2P architecture whose design is injected with specific constructs such as the *Dynamic Virtual Super Peers* (DVSPs) so that the topology of the network is dynamic, highly resistant to failure, and continuously adapts to its usage by the participating entities.

2.1.1 Distribution

The requirement for distributed solutions is reinforced by the need for preserving the *local autonomy* of the participants, especially when it comes to SMEs or even very small businesses (VSBs) for whom managing external uncertainty while maintaining their independence is a a major concern. As will be discussed in the sequel (Section 2.2) from a socio-economic point of view, SMEs are often willing to avoid all together business activities that put their independence and autonomy at risk. In short, local autonomy here refers to the prerogative that each organisation participates in activities of interest by only revealing what it wants to reveal. In software infrastructure terms, this means that organisations want to make their services available (and use services from other organisations) without necessarily providing access to the requester to the actual realisation level of the provided service.

In other words, each participant wants to keep the local design of its platform, including data and implementation, hidden form the outside world and communicate with the rest of the participating entities through calls and responses to its services through well-defined interfaces. These typically only require a service description information which can be made available in some standard language such as the *Service Description Language* or the more recent *Web Services Description Language* (WSDL) or some extension of these considered for example in the DBE project[1] [DBE].

This way of deploying services in developing distributed applications is referred to as *loose coupling* and is the basic premise of the Service-Oriented Computing (SOC) paradigm and its prevalent architectural style called Service-Oriented Architecture (SOA) [Pap03]. Centralised solutions do not lend themselves to designing loosely-coupled services since the centralised coordinator employed typically requires detailed

[1] www.digital-ecosystem.org

information about the local state of execution of each participant in order to coordinate the execution of the rest of the services in the interaction scenario.

In the Oxford Dictionary for Computing in English [ODC] *state* is defined as follows:

> "The condition of all registers, switches, and memory locations of a device, computer or system at a certain time."

It can be seen that the notion relates to execution of a system / process / program code and access to the state of execution of a service implies that the local data and implementation are visible to the outside world, i.e. to the platform requesting to use the service.

A *stateless server* is defined in wikipedia[2] as follows:

> "A stateless server is a server that treats each request as an independent transaction that is unrelated to any previous request."

An example of a stateless server is a World-Wide Web (WWW) server.

An architecture that does not require access to the local state of execution of the parts (subsystems, components, servers) of the system is referred to as a *stateless* architecture while an architecture where communications between parts of the system necessarily go through the local state of execution are often referred to as *stateful*. Considering that SMEs' business model is often in their local design and implementation, they come with a demand for local autonomy, something that is expressed explicitly in the requirements of SMEs in the regions as we will see in Section 2.1.5, and hence the DE architecture must refrain from requiring access to the local data and implementation of the participating entities. What's more, it should be *technology-agnostic* in that it does not require or prescribe a specific technology (data model, database system, programming language, specific vendors' hardware or software) of the users.

In terms of providing support for long-running transactions in DEs the issue of preserving the local autonomy of the participants, including the Initiator of the transaction, is a major challenge. This is partly because in transactions the coordination of the underlying service executions does not only concern the forward actions involved (series of service interactions that need to take place when a transaction is being executed successfully, so while no failure is encountered) but also compensating actions that need to take place (typically in reverse order to that of forward actions) when some service invocation or execution fails in which case the previously successful parts of the transaction must be effectively 'undone'. Even when considering a central point of command and control the issue of coordinating forward and compensating actions between different organisations (service providers) is far from trivial. In OPAALS we are set to do this in a distributed fashion while ensuring it is done in a principled, stateless and tractable manner.

2.1.2 No single point of control or failure

In addition to the issue of preserving the local autonomy of the participants in an interaction, distributed solutions have the additional benefit that they potentially make the overall application more reliable. It is not hard to see that when using a central point of command and control the system is highly dependent on that point being alive and providing the necessary support without which the interaction (or transaction or processing) is no longer possible. This means that if for some reason the central point experiences a failure then the rest of the interaction scenario is abruptly terminated since the participants are agnostic of each

[2] http://en.wikipedia.org/wiki/Stateless_server

other and rely on the central node coordinating the interaction scenario. In modern applications, a failure may occur at various levels; the network connection at a node is lost, a particular service is unavailable, a problem with the logic at a participant, e.g. deadlock, data inconsistency, or at the interaction level, e.g. race conditions. Hence, failure at some stage in a highly dynamic environment is not unlikely.

In ecosystems found in nature there is no single point of control or failure. Species may vary in size and capacity but no single instance dominates the setting to the degree that the whole ecosystem is dependent on its survival. This also relates to the inherent diversity in a natural ecosystems and it should be noted that this would not be the case with a mono-culture (which in the B2B world would translate in market monopoly), e.g. the prairie fields of one type of cereal. It transpires that this principle is useful in designing both transactions themselves and the underlying P2P network to support them. The absence of a critical point of failure (e.g. a central server) at the network level has obvious advantages in terms of reliability of the network. The idea is that while a node goes down, i.e., encounters a failure, the rest of the nodes that are still alive can keep the network going until the failed node revives and joins the network again. This is not possible in centralised P2P networks when the central node fails.

As we will see in the following chapters of this report where the core aspects of the DE architecture in OPAALS are discussed, we have taken particular care to ensure that our design models have no dependency on a single point of control or and do not suffer from a single point of failure.

2.1.3 Reliability

We have briefly mentioned the various types of failure that may occur in a networked environment. Naturally, reliability is one of the major concerns in the DE as it is essentially an open collaborative environment for consuming services, whether these correspond to conducting business activities as such or more general services. In terms of long-running transactions reliability comes in the guise of recoverability, i.e. the provision for recovering a transaction when it has to be aborted due to some failure. In short, this entails the ability to compensate for previous completed actions if a failure of an action later on in the execution of the transaction makes this necessary. There is potential to exploit the interplay between aspects of the transaction model and the characteristics of the underlying P2P network to make for a more sophisticated recovery mechanism in that costly chains of compensations or re-starting the whole transaction can be avoided.

In the Digital Economy connectivity is a major concern in a transactional environment for networked organisations. The network must be up and running at all times and participating entities should be afforded the connections / links to partner organisations in order to carry out transactions corresponding to business activities. Large scale networks are susceptible to *fragmentation*; that is, the network gets divided into smaller networks that are not connected, also called islands, and although there exist techniques for de-fragmentation it is generally a situation from which it is difficult to recover. This implies that the P2P network must be highly resilient to failure and in case failure occurs there must be mechanisms in place that minimise the impact across the network. Since we do not consider a central point of control (central *hub* in networking terminology) and there is no explicit hierarchy or structure, a distributed P2P network has to have good connectivity (often measured in terms of parameters such as distribution degree or cluster coefficient).

It appears that fragmentation is more likely in a network with a static topology. For example, if a few highly connected nodes experience failure (say, as a result of a denial-of-service attack) then the rest of the nodes, which were not directly linked to each other precisely because of the reliance on a few static nodes, cannot get connected to each other. This inevitably leads to the formation of various small islands, some of which may even be comprised of a single node. This risk can be averted by considering a more dynamic

architecture which does not rely on pre-selected static nodes to provide connectivity. To this end, our efforts have been directed at exploiting the high degree of dynamicity and the increased diversity in a DE.

2.1.4 Diversity

In a DE that facilitates cooperation on business activities we expect to find various SMEs including very small businesses (VSBs) that belong to the same or similar domain. By this is meant that some participants will already be doing business together forming a cluster. Every time a long-running transaction takes place among them a temporary private network (again as an overlay over the Internet) is created. Transactions and business domains are not mutually exclusive. Instead, more often than not, overlaps exist in that one SME involved in a particular transaction may also be providing a (small) service to a business activity of a seemingly different domain.

Moreover, SMEs are typically small businesses and one of their main characteristics is that they are flexible and can change and adapt their business model in response to market or technology-driven demands (e.g. see [MacG04] for a discussion on the unique features of SMEs) and as a result enter in transactions with businesses from another domain. This increased likelihood of overlaps can be factored in the P2P network architecture as it may provide a boost for the measurement of reliability for the network. In addition, it is often the case that different participating SMEs may be in different time zones or do business at different times. For instance, consider an online taxi service which is available 24 hours a day and a small business selling shoes whose website may be accessible 24 hours a day but only offers its services between 9am and 5pm. This is another interesting parameter that can be used to effect connectivity and ensure that the overall P2P architecture is highly resilient to failure.

2.1.5 Regional SMEs view of the digital infrastructure

As mentioned before, a digital ecosystem can be seen as a social-economic environment facilitated by a digital infrastructure. From a computer science point viewpoint our aims are directed at providing a sustainable infrastructure for open and trusted collaborations between participating entities. Experience from the DBE project [DBE] has shown that in order to be sustainable the infrastructure should support performant B2B business transactions.

For this purpose *Instituto Tecnologico de Aragon* (ITA) has been working with SMEs in the Aragon region in Spain to gather data that can be used to analyse the requirements of regional SMEs. One of the main information sources for eliciting SMEs requirements for a B2B integration infrastructure and the re-engineering of the Servent has been the data collected by the survey of SMEs conducted by ITA. The detailed requirements for the Servent and support for a B2B Integration among with the full details and methodology used in the survey, as well as other information sources used, is reported in Deliverable D5.3 of OPAALS .

In this report we are concerned with a ‘consensus view’ of what the core DE architecture aims to achieve. The regional SMEs requirements, however, are important from a user perspective and apart from the two aspects mentioned earlier they have also been used to inform key aspects of the core DE architecture. Therefore, in addition to providing a social dimension to the arguments supporting design

decisions we will augment these with requirements coming from the regional SMEs. In what follows, we list a number of requirements expressed by SMEs in ITA's survey which relate to the core DE architecture.

R1. The infrastructure must be service-oriented. Here, a *service* is understood as a mechanism to enable access to one or more capabilities, where the access is provided using a prescribed interface and is exercised consistent with constraints and policies as specified by the service description. This is in fact the definition of a service as given by the Organisation for the Advancement of Structured Information Standards (OASIS) [OASIS]. Usage of a service must only require access to a prescribed interface and not to the data it uses and its implementation details.

R2. The infrastructure must manage service deployment, i.e. the service life-cycle from starting to stopping. It must also manage service searching for the services employed into it. It must also manage the remote execution of services deployed into it, given a known or standard interface.

R3. The infrastructure must allow for service composition between the services deployed into it. When read in conjunction with requirement R1, the infrastructure must support interaction-based model of service composition, i.e. through calls to prescribed interfaces. The survey also showed that dynamic service composition is desirable but can only be applied in restricted areas and dynamic business transactions require a legal or regulatory framework.

R4. The infrastructure must manage transactions in service composition. In the composition of services a transaction manager is needed to coordinate the underlying services deployment of distributed transactions.. The transaction manager must also coordinate the rollback mechanisms that can be executed in case of failure. The transaction manager must manage service compositions and rollback mechanisms without requiring access to the actual state of service execution. A short discussion on *stateless* as opposed to *statefull* architectures was given in Section 2.1.1 and the issue will be revisited in the sequel.

R5. The infrastructure must give support for scalability and performance. Even though there is no agreement on figures to determine performance or scalability, it is under common sense that infrastructure must be scalable and offer good performance. Further interviews with SMEs showed that performance is often represented by response time when searching or executing. Scalability is represented by the challenge of adding new users or services without a high performance reduction.

R6. The infrastructure must be based on standards and give support for semantic management.

R7. The infrastructure must run on different software and hardware platforms.

R8. The infrastructure must manage identity. Any user entity on the infrastructure must be given an identity so that all peers can be identified when engaging in interactions (e.g. business transactions, knowledge services) so that it can be accountable for its actions.

R9. The infrastructure must have support for asynchronous communication.

R10. The infrastructure must implement some self-*properties. In view of its use for connecting SMEs performing business transactions, the aim should be for a reliable platform that can handle failures which means a platform that is capable of self-management, self-organising and other self-properties such as:

- self-healing; automatic discovery and correction of faults
- self-optimisation; automatic monitoring and control of resources to ensure optimal functioning with respect to defined requirements
- self-protection; proactive identification and protection from arbitrary attacks

R11. The infrastructure must use P2P network architecture. This is necessary so that every peer can operate under the same conditions and no SME can exploit the basic infrastructure used by all to gain a dominant position in the market.

In the remaining chapters of this report where we describe the key aspects of the core architecture, in terms of the P2P network (Chapter 4), transaction support (Chapter 3) and trust and identity (Chapter 5) we will make reference to the particular requirements supported by specific aspects of the design models, aiming to show how the core DE architecture can continue to support the Servent and its upgraded version that can be used by regions of SMEs both within and outside the DBE [DBE] Test case.

2.2 Social science perspectives on DEs

There are three main areas of social science concepts and debate that have been employed in this deliverable. These correspond to the research coordinates of language, socio-economics and governance described in Deliverable D1.2 which currently guides the development of the OPAALS' social science agenda. In the context of this deliverable these coordinates manifest themselves as:

- Arguments concerning the significance of autonomy and self-maintenance in digital ecosystems
- Analysis of the broader and locally specific social, economic and technological environment in which smaller organisations interact as both businesses and non-business entities
- Social and political science arguments concerning power, participation and technology implementation

From a social science standpoint, critique of the core architecture has occurred at both a conceptual and an applied level. Social science researchers have challenged the design rationale behind component parts of the core architecture in terms of the assumptions they imply about the broader socio-economic environment and in terms of specific operational implications for potential digital ecosystem users. In-depth discussion of the key theoretical concepts employed in this deliverable will be provided in deliverables D1.2, D12.1 and D12.2, and hence these concepts are only outlined here.

In the context of this deliverable and the critical dialogue that underpins it, the axes around which each of these theoretical discussions revolves reflects the accepted need to reintroduce questions of context back into modelling and design activities. The iterative relationship between models and their real-life contexts is one which inter-disciplinary research dialogues bring to light particularly well. To a greater or lesser extent, research methodologies constitute processes through which individual subject disciplines generate focus on a particular phenomenon. This necessitates that some processes and phenomena are placed in the foreground of analysis, whilst some are temporarily pushed into the background and others placed completely beyond the analytical frame (for example, 'externalities' in economics). In order to concentrate their analytical and developmental efforts, computer scientists have to generate models and rationales that necessarily limit the implications that local or specific social contexts might bring to bear on the knowledge and technologies they produce. During the development phase decisions are therefore primarily driven by pragmatic and technological rationales. Whilst certain key principles regarding the intended implementation context are kept in mind, it is inevitable that a detailed view of both the broader environment and local circumstance are lost.

This explains the significance of interdisciplinary or multi-stakeholder dialogue in technology development contexts. Purposeful critical debate between researchers - or between researchers and users - allows these lost elements that surround expert areas of knowledge to be recognised and addressed. In the case of technology development, by reintroducing the complexity inherent in specific and local contexts, social scientists aim to ensure that 'value-free logic' (i.e. logic that is entirely self-referential or premised purely on the internal pragmatics of a system or model potentially to the detriment of social equalities), are countered through a continuous process of re-contextualisation.

With respect to digital ecosystems, this requires looking carefully at assumptions concerning the extent to which the core architecture is capable of reconfiguring specific socio-technical relations. To facilitate this process social scientists are looking closely at:

- data that reflects the needs and requirements of potential digital ecosystem users
- economic-related analysis regarding infrastructure and the disruption of monopolistic practice
- specific socio-economic understanding of the business-to-business environment of SMEs including the significance of trust, accountability and identity models
- autonomous and self-maintaining characteristics of SMEs
- the significance of social phenomena in driving innovation and new modes of interaction
- concepts of governance relating to common infrastructure and participatory processes

A number of these perspectives have been written about in a paper by Dini et al. which will be published in a special issue of the International Journal of Technological Learning and Development on global value chains and innovation networks: prospects for industrial upgrading in developing countries. Key arguments from that paper will be cited in this section and the paper in its entirety is appended to this report.

The sub-section that follows provides specific details of how the interdisciplinary process of consensus building has taken place. It provides an overview of the critical questions that social scientists raised with respect to the core architecture. These questions are returned to and addressed later in the deliverable. The questions themselves were informed by specific social science theories and debates that researchers drew upon during the course of dialogue. A brief overview of the theoretical concepts and arguments social science researchers have employed with respect to their appraisal and critique of the core architecture is provided in sub-section 2.2.2. The final section provides two examples of research that provide insights into the broader socio-technical environment of smaller organisations and specific, local concerns of potential digital ecosystem users.

2.2.1 Critical interdisciplinary dialogue as a means to consensus building

Understanding the relationship between technology and the social contexts in which it is designed, developed and used is recognised as being of critical importance to its operational, economic and social viability. The consequences of large-scale technology use and its role in reconfiguring social relations – in the workplace, between governments and their citizens, and in terms of individual capacity for global communication – represent key areas of concern. However, so too are questions of information technology failure and the 'productivity paradox' where technologies do not produce the economic benefits that they were designed to achieve [BrH03].

Social scientists have argued that technologies do not realise intended results because insufficient synergies are developed between technology design, development and implementation processes, and the social contexts within which these activities take place. Ignoring social context in favour of technologically-driven rationales marginalises the significance of political, social and economic (or business) concerns when, in reality, these aspects of a situation could prove to be of central importance to realising a technology's potential. The significant role that knowledge sharing between designers - and between designers and local users - plays in successful technology implementation has been widely recognised [Fle94]. The need to foster conditions whereby knowledge sharing is supported and encouraged has been identified as a key component of many diverse technology development and implementation environments.

It is for these reasons that the OPAALS project aims to create an interdisciplinary foundation for digital ecosystems that involves intensive dialogues between social and computer scientists. Through our research

conversations we aim to generate synergies between: our different areas of specialist knowledge; between the technologies developed and their broader social context; and between those technologies and their users.

In practice, a number of different activities have helped us to achieve this end. Firstly, in order to conduct our conversations we have had to develop 'common' understanding of technical terms and concepts. Within our own research communities terms carrying specific meanings become naturalised. Therefore, our process had to begin with a level of reflexivity which required us to identify when we were using technical terms. We then had to be prepared to 'unpack' the meaning of those terms for each other. Having established a shared understanding we could then relate questions and issues regarding the topics under discussion to our respective bodies of knowledge.

As an example of this process, below are 2 descriptive pieces of text: one written in computer science terms and the other in social science terms. The discipline-specific terminology used within each text is underlined. Both pieces of text point towards an explanation of why the digital ecosystem core architecture has been designed in the way it has.

> **Computer science**
> In digital ecosystem terms, long-lived transactions require a stable, de-centralised, peer-to-peer architecture that can protect against fragmentation.
>
> **Social science**
> Underlying the transaction coordination model design is an assumption that SMEs currently operate within an IT environment that potentially inhibits their capacity to innovate and/or adopt technology due to specific barriers to participation.

In order to demonstrate the process of 'unpacking' these terms and descriptions a synthesised version that offers researchers a common understanding of these two sentences is provided here.

> Small organisations have to make imortant choices concerning the technologies they use to carry out their work. The options they have to consider are often complex and the degree of choice they actually have is frequently limited. Whilst the complex nature of information technology is difficult to overcome, the limitations smaller organisations experience could be alleviated. In the past it has been argued that the only way to ensure that complex technological processes have time to complete is to ensure that they take place within a technology environment that is centrally owned, overseen and managed. However, recent technological development carried out by OPAALS computer science researchers indicate that a centralised environment is not always necessary and that an integrated framework for interaction can be created between computers that are physically and organisationally separated.

Through developing a common understanding of terms in this way, OPAALS social scientists have been able to question design and operational aspects of the core architecture. This involves a process of applying relevant social science theories to illuminate areas of critical analysis enabling assumptions underlying key design characteristics and decisions of the core architecture to be explored in relation to empirical contexts with which social science researchers are engaged. This is another example of the inter-epistemological dialogue our project is engaged in that is discussed in D1.2.

Many of the critical questions asked by the social science researchers focused around the peer-to-peer network design. The assumption behind this aspect of the core architecture is that its construction would prevent and potentially disrupt the emergence of monopolies or monopolistic practices. Essentially these questions focused on concepts and theories of power and the way power manifests itself in technological and economic environments.

Of particular concern to the social science researchers were the 'in practice' operational consequences of having particularly powerful nodes within the network – so called 'super peers'. A number of very specific questions were raised on this issue where different potential contexts and scenarios were explored. These revolved around the potential for peers to exploit their position in the network and the capacity of smaller companies to fully participate in the network created.

The culmination of this critical dialogue was a request to change the terminology used by the computer scientists to refer to network 'super peers'. It was generally felt that the term 'virtual super peer' carried a suggestion of privilege or superiority. In addition, it did not adequately capture the dynamic and continuous selection between candidate nodes in the formation of the 'virtual super peers'.

From a social science point of view, it was important to stress that the network topology formed by the digital ecosystem core architecture components is dynamic. As a form of policy intervention, the underlying aim of the digital ecosystem architecture is to provide an alternative to the static and inflexible character of current technology infrastructure. A network topology that is dependent on having a few, powerful nodes or peers allows larger organisations to dominate operations and therefore affords them greater potential to exploit technology-related business opportunities. Smaller organisations are structurally excluded by this kind of design - from business opportunities, from knowledge sharing, from opportunities to innovate – and lack either the influence or the computing capacity to alter the situation. Therefore, it was important to the social scientists to draw a clear distinction between previous notions of 'super peers' as powerful, fixed points in a network and the fluid, dynamic nature of the virtual super peer concept. It is the fact that powerful nodes in the network are dynamic that gives the architecture its unique capacity to prevent domination.

As a result, the name 'dynamic virtual super peer' was adopted. Whilst this change did not affect the core architecture design in any substantive way it nonetheless carried implications for how a key characteristic of the architecture would be communicated to users.

The critical modes of analysis used by social scientists to pursue these and other questions were founded on a range of social science theories. Of particular significance were social science theories related to: knowledge; power; regulation; trust; identity and accountability; and autonomy and self-maintenance. Due to the significance of these theories to the process of critique the core architecture has undergone, a brief overview of each is provided in the following section.

2.2.2 Key social science theories used in the critical analysis of the DE architecture

This sub-section will offer a necessarily brief summary of the central theoretical concepts that have been used to frame research into digital ecosystems in general and more specifically, interdisciplinary debates focussed around the core architecture. More lengthy discussion of these concepts appears in other deliverables. For example, in Deliverable D1.2, the first part of a working theoretical framework for digital ecosystems is proposed including a high level discussion of the social science agenda and the 3 main 'research coordinates' of language, socio-economics and governance. This report also includes a review of the influence autopoiesis has had on thinking about social systems, as well as criticisms of the concept from sociological perspectives. These arguments will be developed in more detail in Deliverable D12.1 which will present the second part of a theoretical framework for digital ecosystems focussing on "associative" as well as autopoietic viewpoints on digital ecosystems as knowledge communities.

One of the main theoretical challenges OPAALS faces is that of carrying out interdisciplinary research into knowledge and the role of knowledge in building social and economic (i.e. business) communities.

Design and development of the core architecture will play a vital part in advancing our understanding of this challenge. There are two main ways in which this will occur. Firstly, the 'open' mode of infrastructure development and governance constitutes an effort to share not only the technological outputs of publicly funded innovation but also the 'know-how' involved in producing those components and the decision-making powers associated with their ongoing development. By making documentation and technological components of the architecture openly available via licensing and publication agreements, knowledge of how to produce and develop the infrastructure is made publicly available to potential users. The open source licensing used ensures that this knowledge *remains* in circulation. Secondly, the core architecture will hold implications for the types of knowledge sharing and communication that the digital ecosystems infrastructure can ultimately support. In subtle and not-so-subtle ways, technological infrastructures shape the way communication and knowledge sharing takes place. With regard to its integration with knowledge sharing applications and environments, such as the OPAALS open knowledge space, potential ramifications of the core architecture will be difficult to assess until technological implementation has actually occurred. These integration activities have been delayed but are scheduled to take place over the course of the next few months after which point a further set of interdisciplinary consensus building activities will take place. Nonetheless, the concepts of knowledge and 'openness' remain central to the social science agenda and as such are discussed here, as are the key concepts of: power, regulation, trust, accountability and identity, autonomy and self-maintenance.

Open

The animating philosophy of OPAALS is aptly summarised in the first word making up the project's acronym: "Open". The simplicity of this word belies deeper and more complex connotations that include a commitment to communication and reflexivity from both theoretical and methodological perspectives. In terms of communication, the core principles animating the work of OPAALS are the ideas of transparency and accountability, echoing ideas of a democratic, collaborative process of community building. Communication is understood not only in the sense of dialogue, deliberation, debate and exchange but also in terms of the technological structures that make these possible..

As well as these, openness refers to a normative preference for open systems of community building and governance (including, of course, the technology chosen to materially manifest that community into a software architecture). This implies a commitment to moral equality in the context of diverse disciplinary approaches to research and exchange (plurality), as well as to non-proprietary models of making available the outputs of this research (transparency and accountability). In this sense, openness involves a commitment to a multi-faceted and pluralistic view of scientific research, which echoes democratic collaborative processes. As summarised in the introduction to D1.2. the main topics of OPAALS research are "language, knowledge production/management, community building …[as well as] methodological, epistemological, and ontological questions underpinning research in democratic principles and processes".

Knowledge

Deliverable 10.5 suggests that the core concepts framing OPAALS social science research can be summarised by the phrase "open knowledge". It is doubtless that the concept of knowledge is theoretically central to OPAALS. This is not to discount the acknowledged complexities and problems of defining "knowledge" within an interdisciplinary research community. Nevertheless, a number of social science partners' work centres around concepts of knowledge and its role in: reflexive, interdisciplinary, research; socio-economic development on a regional scale; and empowerment. The ultimate aim of OPAALS is to contribute substantively (in terms of technology, social analysis, empirical data and the establishment of an open knowledge community and space) to regional socio-economic development. The culture of research and communication developed around the digital ecosystems infrastructure is vital to ensuring that an open, participatory environment for knowledge sharing exists in which all stakeholders can engage. Among these key stakeholders are the regional and European policy makers whose need for high quality information is particularly acute and WP11 aims to formulate a basis for meeting this knowledge requirement. Given the

significance of cultivating an open and accessible knowledge base around the digital ecosystems infrastructure, the organisation, management and licensing of this 'open knowledge space' – i.e. its governance – is an important consideration of for OPAALS researchers.

Since the concept of language is so central to the social science research agenda it is important to acknowledge, as Foucault does, that

> "Knowledge and language are rigorously interwoven. They share, in representation, the same origin and the same functional principle; they support one another, complement one another, and criticize one another incessantly" (Foucault, 1966: 95, see [Fou66]).

It is for these reasons that so much of the work of OPAALS focuses on language, both natural and, more importantly in the context of this deliverable, formal language. Conceptualising knowledge through the prism of language, the social scientist asks, 'how does the formal language that encodes the technological architecture of OPAALS facilitate the complex task of reflexively and transparently creating both knowledge and a knowledge community?' Furthermore, assuming that knowledge is embedded in human practice and understanding, 'what are the implications of our technology infrastructure for facilitating and extending knowledge sharing and exchange'?

Power

Conceptualisations of power and the means through which it is exercised are central to social science research. "Open knowledge", considered as a set of values that inspire and shape the work of OPAALS, implies a commitment to empowerment and the distribution of power beyond existing, unsatisfactory and unjust, power structures. This kind of conceptualisation of power might lead, for example, to a socio-economic discussion of defensive or monopolistic practices on the part of large technology companies or a socio-technical analysis of the licensing regimes they operate. In the context of a distributed, peer-to-peer architecture power is consciously diffused through processes and set of relationships rather than mediated through hierarchical, top-down structures. Foucault's conceptualisations of power-relations, explained by, Gilles Deleuze provide an interesting frame of reference here.

> Power relations are simultaneously local, unstable[3] and diffuse, do not emanate from a central point or unique locus of sovereignty, but at each moment move 'from one point to another' in a field of forces, marking inflections, resistances, twists and turns, when one changes direction, or retraces one's steps ([Del06]).

As social scientists interested in the possibility of creating more distributed and egalitarian relationships, assessment of OPAALS' technology architecture therefore requires an informed critique that draws on a variety of empirical and theoretical research and encompasses a broad range of social science disciplines. For example, political science is an important resource for considering the implications of governance regimes and their capacity to foster participation and power sharing (. In addition, there is a body of research focussing on the sociology of technology where specific insights regarding technology and its capacity to influence human behaviour have been developed.

Regulation

A range of regulatory domains in e-business activities have been identified for the purposes of collaboration of SMEs in digital ecosystems. Regulation can be defined as (i) the presentation of rules and their subsequent enforcement usually by the state, (ii) any form of state intervention in the economic activity of social actors, or (iii) any form of social control whether initiated by a central actor such as the

[3] By "unstable" we should take Deleuze to mean "unfixed" rather than "unreliable".

state or not and including all acts whether they are intended to be regulatory or not. Especially the last is often used as equivalent to governance [BSH98], [BaC99].

An extensive review of literature from US, EU, and international organizations led Berkey [Ber02] to identify three main categories of international regulatory issues related to e-business:

- Privacy and consumer protection. This refers to processing, control and distribution of personal and consumer data over electronic formats on the ground of the individual rights and freedoms of e-business users.
- E-signatures and security when sharing information over digital media. This determines significantly the system's autonomy and cross-border interoperability through authentication, integrity and non-repudiation.
- Jurisdiction and consumer protection. This results from the cross-border nature of e-business services, and the associated challenges in contractual relationships between service providers and customers.

Regulation in e-business follows the principle of requisite variety established in systems theory by W. Ross Ashby: 'the larger the variety of actions available to a control system, the larger the variety of perturbations it is able to compensate' [Ash56]. The ability to manage complexity in a self-organising and evolving system is in direct relation to the capability of the system to represent diversity through combinations of less complex regulations. Hence, in order to manage the complexity of the regulatory domain in digital ecosystems, it is helpful to build from general constraints towards the nuances of local- and user-specific implementations. This will allow systemic requirements and user needs to collaboratively construct a complex and continuously evolving regulatory framework for digital collaboration and business.

Trust

Trust is a concept defined and used variously in a wide range of social science disciplines, such as sociology, psychology and media studies. A sociological approach to trust that has been recognised broadly in social science disciplines is Coleman's sociological definition of trust and his theory of prisoner's dilemma [Col94]:

1. Placement of trust allows actions that otherwise are not possible (e.g. when there is incomplete information).
2. If the trustee is trustworthy, the trustor is better off than if he or she had not trusted. Conversely, if the trustee is not trustworthy, the trustor is worse off than if he or she had not trusted.
3. Trust involves a voluntary transfer of resources from the trustor to the trustee with no real commitment from the trustee.
4. A time lag exists between the extension of trust and the result of the trusting behaviour.

From a management perspective, these four elements of trust are considered constituents of a trust relationship, which in turn can be defined as:

> The willingness of a party to be vulnerable to the actions of another party based on the expectations that the other party will perform a particular action important to the trustee, irrespective of the ability to monitor or control that other party. ([MDS95])

In this sense, trust enables action by establishing confidence among interested parties in the expected outcomes of current or future transactions [Cla02], [DuS05]. One important prerequisite to confidence is 'certainty'. In the e-business context, certainty is related to trust in each of the three facets that Nachira [Nac02] identified as necessary to a digital ecosystem:

- Trust in services and technological solutions
- Trust in business activities

• Trust in knowledge

Literature and empirical research in the field acknowledge that trust is a critical enabler of e-business ([Cla02], [Kee00], [SUG+02], [SwR04]) and the foundation of the digital economy [Pav02], [SUS02]. In the context of online commerce, electronic networks provide increased possibilities for opportunistic behaviour compared to offline settings where face-to-face communication exists ([Cla02] [Pav02]), and, therefore, trust relationships constitute an enabler for companies to engaging in e-business activities [SUS02].

From a social science perspective, trust in digital business ecosystems can be approached on the basis of a three-dimensional model that Meents, Tan and Verhagen [MTV03] developed for B2B virtual marketplaces. The authors suggest three main types of trust relationships in e-business:

- Trust X: participants joining the system must trust that the system will provide secure services over proven technology, and that it is capable of facilitating trust relationships in business activities.
- Trust Y: established participants must trust that other participants joining the system will not behave opportunistically, perhaps through asymmetrical access to information.
- Trust Z: the system must ensure that trust relationships can be established between participants themselves.

The expectation is that the system's technical architecture will provide all the requirements for carrying out transactions in such a way that compliance with established rules and norms will be ensured. In order to establish good trust relationships between participants and trust in the system overall, the participants are expected to comply with established laws and norms, while the system will provide all requirements for the applicability and enforcement of those laws and norms.

Trust is important in distributed software environments where decentralised, resource sharing and scalable P2P networks exist. Interactions (exchanges) in such environments require trust (trust relationships) not only between businesses or other actors participating in the system but also trust in the system as a whole. Trust is critical for the sustainability and growth of society in general and for the establishment of business networks in particular. In DEs and in technology-mediated communications and transactions, trust becomes even more critical because of lacking physical interaction between communicators. Digital environments pose extra risks with regard to security, privacy and the ways in which control and regulation should and could be implemented. These issues raise the importance of trust in the system and the ways in which trust relationships among participants can mitigate some of the online risks, while enabling more efficient and balanced regulations to apply.

Identity and accountability

Trust is arguably interwoven with accountability [DFM00] and identity [Dou02]. Especially the notions of identity and reputation constitute key elements of trust both in social science and computer science. From a social science perspective, reputation based on the identity of agents enables trust relationships between agents, encouraging interaction and exchange. From a processual perspective, identity is to be understood as highly dependent on the context and trust relationships required for communication and elaboration between two or more participants in the system. In other words, depending on the trust criteria that someone uses to evaluate his or her collaborator, the identity elements assigned to the latter vary.

Identity in social science is often understood as the attributes of 'self' and is variously defined in disciplines such as psychology, sociology and social anthropology. Typical example is Erik Erikson's psychological framework of identity and the distinction between the ego identity (the psychological sense of self), the personal identity (the personal traits that separate one person from another) and the social or cultural identity (social roles that a person might play). For Erikson, the development of a strong ego identity and the proper integration into a society and culture may lead the person to a stronger sense of identity. Accordingly, a deficiency in any of these three parameters may lead to an identity crisis.

In electronic commerce, authentication is the process whereby an entity (person, computer or organization) establishes that another entity is who it claims to be. Thus, authentication involves issues such as membership, access rights, and public certification of identity. Authentication can facilitate trust relationships since it provides the means for identifying any malpractice of the system participants. Digital signatures are one of the most widespread methods of proving identity to those who do business online.

The importance of trust and identity is critical for the existence and sustainability of human communities. Indicative is how important they are for the functionality of 'money' that is used within and by a community, namely of community currencies. More specifically, trust and identity in CCs are founded and operationalised on the basis of the following principles (LETSystems – new money. Available at: http://www.gmlets.u-net.com/):

- Co-operation and mutual responsibility for the integrity and stability of the network, as no-one owns the network;
- Self-regulation, with the users defining the community rules and regulations;
- Empowerment, as all users are entitled to issue the currency;
- Money as a means of exchange that can take various formats, while being used for satisfying various community-based needs.

Also, one could refer to the following foundations of trust in CCs, such as scarcity, accountability, scale and structure, while trust is a prerequisite in order community members to drop regular currencies and do business with 'money' that is only community-based legitimized.

Autonomy and self-maintenance

From an organisational point of view, SMEs are distinguished from their large counterparts on the ground of size, numbers, organisational structure and interactions with the broader socio-economic environment. From an information systems and management perspective, MacGregor [MacG04] provides a very interesting discussion of the unique features of SMEs. These features are summarised in the following table:

ID	Features unique to SMEs	Reported by
	Features related to management, decision making and planning processes	
INT 1	SMEs have small and centralised management with a short-range perspective	Markland (1974); Reynolds *et al.* (1994); Bunker and MacGregor (2000); Welsh and White (1981)
INT 2	SMEs have poor management skills	Blili and Raymond (1993)
INT 3	SMEs exhibit a strong desire for independence and avoid business ventures which impinge on their independence	Dennis (2000); Reynolds *et al.* (1994)
INT 4	SME owners often withhold information from colleagues	Dennis (2000)
INT 5	The decision-making process in SMEs is intuitive, rather than based on detailed planning and exhaustive study	Reynolds *et al.* (1994); Bunker and MacGregor (2000)
INT 6	The SME owner(s) has/have a strong influence in the decision-making process	Reynolds *et al.* (1994); Murphy (1996); Bunker and MacGregor (2000)
INT 7	Intrusion of family values and concerns in decision-making processes	Dennis (2000); Bunker and MacGregor (2000); Reynolds *et al.* (1994)
INT 8	SMEs have informal and inadequate planning and record-keeping processes	Reynolds *et al.* (1994); Tetteh and Burn (2001); Miller and Besser (2000); Markland (1974); Rotch (1987)
INT 9	SMEs are more intent on improving day-to-day procedures	MacGregor *et al.* (1998)
	Features related to resource acquisition	
INT 10	SMEs face difficulties in obtaining finance and other resources, and as a result have fewer resources	Cragg and King (1993); Welsh and White (1981); Gaskill and Gibbs (1994); Reynolds *et al.* (1994); Blili and Raymond (1993)
INT 11	SMEs are more reluctant to spend on information technology and therefore have limited use of technology	Walczuch *et al.* (2000); Dennis (2000); MacGregor and Bunker (1996); Poon and Swatman (1997); Abell and Limm (1996); Brigham and Smith (1967)
INT 12	SMEs have a lack of technical knowledge and specialist staff and provide little IT training for staff	Martin and Matlay (2001); Cragg and King (1993); Bunker and MacGregor (2000); Reynolds *et al.* (1994); Blili and Raymond (1993)
	Features related to products/services and markets	
EXT 1	SMEs have a narrow product/service range	Bunker and MacGregor (2000); Reynolds *et al.* (1994)
EXT 2	SMEs have a limited share of the market (often confined to a niche market) and therefore rely heavily on few customers	Hadjimonolis (1999); Lawrence (1997); Quayle (2002); Reynolds *et al.* (1994)
EXT 3	SMEs are product-oriented, while large businesses are more customer-oriented	Reynolds *et al.* (1994); Bunker and MacGregor (2000); MacGregor *et al.* (1998)
EXT 4	SMEs are not interested in large shares of the market	Reynolds *et al.* (1994); MacGregor *et al.* (1998)
EXT 5	SMEs are unable to compete with their larger counterparts	Lawrence (1997)
	Features related to risk taking and dealing with uncertainty	
EXT 6	SMEs have lower control over their external environment than larger businesses, and therefore face more uncertainty	Westhead and Storey (1996); Hill and Stewart (2000)
EXT 7	SMEs face more risks than large businesses because the failure rates of SMEs are higher	Brigham and Smith (1967); DeLone (1988); Cochran (1981)
EXT 8	SMEs are more reluctant to take risks	Walczuch *et al.* (2000); Dennis (2000)

Table 1: Unique features of SMEs (Mac Gregor, 2004: 3)

As shown in the table above, the features of SMEs are classified as being internal or external to the business. Internal features are those relating to management, decision-making and planning, as well as to acquisition of resources, while external features relate to the market and the broader external environment (risk taking and uncertainty). The literature in the field has argued that small firms are more risky, more vulnerable to failure and less efficient in record keeping than large firms. In the last few years, where technology has become major enabler of business activities, the literature has focused more on the ways in which SMEs relate to new technological artefacts. Thus, management and information systems studies argue that SMEs lag behind large companies in technical and technological expertise due to insufficient capital, inadequate organisational planning, and the smaller scale of products and/or services available to customers. From a management perspective, it has been argued that SMEs have a small management team, they usually adopt a top-down management strategy and they aim to remain independent. Also, uncertainty is a key difference between small and large businesses, with internal uncertainty being more a characteristic of large business and with external uncertainty characterising smaller organisations.

From the above internal and external features of SMEs what deserves further attention is the trait of external uncertainty and independence. On the one hand, small firms aim to remain independent and autonomous, as they avoid business activities which put their independence and autonomy at risk (Dennis, 2000; Drakopolou-Dodd et al., 2002). On the other hand, small firms are marked by external uncertainty that derives from their low capability to control the external socio-economic environment and conditions, being thus more likely to evolve and change over time than larger organisations [Sto94].

Literature on the Information Society has argued that independence and ability to change are traits that make SMEs more flexible and self-sustainable, as well as more likely to adapt to and benefit from technology-mediated and internet-based business [AuG97]. However, administrative and economic constraints often constitute barriers to the potential of SMEs to adopt new technologies. Nevertheless, technological networks appear to provide SMEs with necessary interorganisational links and adaptability to external changes. Networks are appropriate for SME based organisational structure, as they are more adaptable and flexible due to loose coupling and openness to information [AcK99].

The complexity of our society and economy, and the rapidly changing contextual conditions within which human activity takes place pose challenges to SMEs. Small firms are in need to respond to contextual particularities and changes, adopting relevant mechanisms and espousing flexibility and self-maintenance as requirements for long-term sustainability. The question of interest is how technological systems and digital system architecture can respond to such needs and to the particularities of the socio-economic environments we live in and construct. What the EC suggests is that we 'extend systems engineering methods to deal with open-ended and frequently changing real-world environments', as what it is needed is 'new systems design and engineering principles and implementations for machines, robots and other devices which are robust and scaleable enough to deal with the real world and to behave in a user-friendly and intuitive way with people in everyday situations' (EC, 2007: 1, see [EC07]).

More specifically, management and information systems literature has argued that P2P systems are particularly appropriate for SMEs because the infrastructure for enabling such systems is easy to establish (e.g. see Sweeney et al, 2001; [Bon01]). Also, in business models suggested for SMEs knowledge sharing often crosses different organizations [Spa01] and P2P technology enables cross-organisational collaboration, information sharing and workflow (for example, see the model proposed by Rehfeldt and Turowski in [ReT00]).

However, not only autonomous, scalable and self-maintaining technology systems are needed for SMEs. Small and medium enteriprises are essential for innovation building, as 'they play vital roles in the development and nurturing of new visions in ICT and their applications and in transforming them into business assets' (EC, 2007: 7, see [EC07]).

2.2.3 Empirical examples to the social science analysis of the DE core architecture

In this section, two examples of empirical social science research are summarised. Full versions of the documents upon which these summaries are based are provided in the appendix to this report (see Appendix A). The first example offers a broad analysis of the technological and socio-economic environment within which small and medium-sized enterprises (SMEs) currently operate. This article makes macro level, policy recommendations relating to the need to increase competition within the technology environment for small businesses. The second example provides a micro level analysis of SMEs themselves and draws on

data that describes their day-to-day use of technology. From this data the researchers draws important conclusions regarding the SME experience of technology procurement, use and maintenance.

The overall aims and intentions of the digital ecosystems core architecture are expressed very clearly in an article written by Dini et al. which can also be found in the appendix to this report (see Appendix B). Written from a socio-economics perspective, the article explores potential reasons to explain divergent trends within the current electronic business environment. According to the authors, in business-to-customer (B2C) interactions, new technologies have facilitated significant innovation and the development of radically new business models. However, the same trend is not observable in business-to-business (B2B) electronic interactions. They claim that it is the tightly-coupled and inflexible organisation of technologies underlying B2B interactions that are responsible for this trend. They identify 'lock-in' - where a lack of alternatives means technology users are contractually tied into using a particular set of technologies - as a key characteristic that limits the extent to which new technologies can be explored and combined.

The significance of the core architecture for digital ecosystems is highlighted by this analysis. Many of the rationales for maintaining a centralised B2B infrastructure, owned and maintained by a limited number of large technology companies, are technological. However, the core architecture offers a viable alternative to centralised arrangements and as such, it has the potential to open up new possibilities for SME technology use.

This article frames issues associated with digital ecosystems in a particular way taking a primarily socio-economic perspective and focussing in on potential macro level, policy actions. However, from a social science point of view it is also critical to take into account micro level and holistic or 'whole system' evaluations of these phenomena. In addition, there are other related social science theories and perspectives that are also important to understanding the significance of the digital ecosystems core architecture. The following example provides a ground-up view of the macro phenomena described in the article above.

The second article focuses on a study of the computing experiences of very small businesses (VSBs). These organisations are frequently subject to resource constraints and lacking in computer skills. For the purposes of the study, size of firm was defined by number of workstations, rather than by the conventional, number of employees. The data is cross-sectoral, based on feedback from 30 questionnaires, 8 in-depth interviews and around 30, short 'doorstep' interviews in which companies were asked the neutral question, 'how are you getting on with business computing?' The work was conducted using grounded theory method (GTM). The dominant category of concern amongst the firms emerged to be **interoperability**, with ***minimal use*** or ***avoidance*** of computerised solutions as a further prominent, repeating factor.

For these businesses, the acceleration of the software upgrade cycle was found to pose a significant threat. Core business activities were frequently stalled whilst employee efforts were focused on updating software versions. Firms involved in the study perceived software upgrade as a potential point of failure. Some of the reasons given for this were: past experience of adverse impact on other application and system software; loss of interoperability with the new version; and the necessity to upgrade hardware to cope with amplified software functionality (much of which the firms neither desired nor required). Writers (technical and editorial) occupy professions especially afflicted by backward and lateral incompatibility problems between both successive application versions, and versions of the same application written for other proprietors' hardware and operating systems. One interviewee reported a 20% reduction in productivity through this cause.

The study found it fruitful to explore the core category of *interoperability*, with *minimal use or avoidance* of computerised solutions in terms of the participant entities in interoperability relationships: interoperability amongst knowledge domains; media; humans; humans and machines; and machines. Across these associations lie the cardinalities of the relationships: one-to-many, many-to-many, one-to-one, between the VSBs and their customers, suppliers and other contacts.

A major conclusion of the paper was that the firms preferred software upgrade to be business-led, and not coerced by a software publisher. The software 'migration funnel' currently drives the user forward to the latest upgrades, frequently leaving no return path of backward compatibility.

An ancillary conclusion concerns the threat posed to digital holdings, which are becoming a substantial part of the modern knowledge-base held in the world's libraries. The lack of long-term version stability heightens the probability that electronic readers for these documents will disappear from the library community, and material will be irretrievably lost.

2.3 Summarising notes and outline

This chapter has described the key computer science and social science concepts that underlie the digital ecosystems core architecture design. Achieving the level of integration required to allow the discussion of topics that rely on disciplinary specific terms and concepts is a difficult feat. In the OPAALS network this feat has been achieved through intensive critical dialogue. In practical terms, this has resulted in a process of questioning each other in order to ensure we understand each others terms and assumptions.

Within this deliverable, the social scientists have highlighted the issues that strike them when they consider the policy domain of digital ecosystems. They then suggest some areas of theoretical debate that can open up these issues for further consideration. Likewise, the computer scientists have looked closely at technological configurations that have led to the current state of imbalance in the electronic business-to-business landscape. They have conceptualized key components that contribute to the problems of lock-in and dependency that exist in this area and developed solutions that will – in a technological sense – undo the structural dependencies that these forms of design encourage. Synthesising how these two standpoints can be used to express key issues within digital ecosystems is one of the main, ongoing tasks of the OPAALS network. It is hoped that by achieving such synthesis the perceived drawbacks of computer and social science – such as, for example, techno-centrism or lack of practical, solution outcomes - will be overcome.

As one of the main aims of this deliverable is to provide a detailed, technical description of the digital ecosystems architecture, the following chapters focus in on the core components of that architecture and are written from a computer science point of view. However, at the end of each chapter are a set of questions that have come out of the process of critical dialogue with OPAALS social science researchers, along with the answers that the computer scientists consequently provided. It is this process of critical dialogue and questioning that runs through the core of this design process and that therefore runs through the core of this deliverable.

In effect, in terms of understanding the structure of the deliverable as a whole, this chapter acts as a guide. The detailed technical description of core architectural components provided in the next 3 chapters is constructed around the key concepts outlined in Section 2.1. The social science questions that come at the end of each of these chapters are informed by the process of critical dialogue that has taken place and the theoretical positions described in Section 2.2 of this chapter.

3 The OPAALS Model for Long-running Transactions

In this chapter we outline our framework for coordinating distributed transactions in digital ecosystems. The objective is to support distributed long-running transactions which are essential in performing long-lived business activities in open communities of small and medium sized enterprises (SMEs). We highlight the basic characteristics of long-running transactions in digital ecosystems for business and then proceed to outline the key aspects of the transaction support in the core DE architecture considered in OPAALS.

3.1 Long-running transactions in DEs

The conventional definition of a transaction within the database community [Dat96] is based on ACID properties (Atomicity, Consistency, Isolation, Durability). In advanced distributed applications however, these properties often present unacceptable limitations and reduce performance dramatically. In a business environment, 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 or hours or even days.

A *long-running* transaction between SMEs in a digital ecosystem for business can be either a simple usage of a web service (rarely in B2B relationships) or a mixture of different levels of composition of several services from various service providers. This makes the adoption of the *Service-Oriented Computing* (SOC) paradigm perhaps more relevant than ever. The basic premise of a service-oriented architecture (SOA) is that applications from different providers are offered as a service that can be used, composed and coordinated in a loosely-coupled manner. In short, the deployment or usage of a service does not require knowledge of the actual realisation or implementation of the service. This allows enterprises to mix and match services to perform business transactions within a digital ecosystem with minimal programming effort.

Long running multi-service transactions typically involve interactions and coordination between multiple partners that engage in long-lived business activities. It is often the case that internal activities (or subtransactions) need to share results before the termination of the transaction (transaction commit). Further, many business scenarios in a digital ecosystem require that a transaction releases some results to another transaction, before it commits (partial results). More generally, dependencies exist both within and across transactions.

It transpires that when designing transactions in a purely distributed and highly dynamic environment such as that of a digital ecosystem it is impractical, and in fact undesirable, to maintain full ACID properties. In particular, Atomicity and Isolation are questionable in this highly versatile business setting where transactions are performed for a range of B2B scenarios. The dynamicity of the environment also increases the likelihood that some transaction will fail. The standard practice in the event of failure is to trigger compensating actions that will effectively ‘undo’ the effects of the transaction up to the moment of failure. And this must be done in a way that takes into account the dependencies both inside and outside of a transaction.

Further, the recovery and compensation mechanism must respect the loosely-coupled nature of the underlying service executions in order not to violate the local autonomy of the participants, which poses further challenges for recoverability in the face of maintaining consistency.

The abortion of a transaction, even if it is successfully recovered and compensated for, can be very costly in a business environment. It is thus important to build into the system the capability to deal with failure. In other words, it is key to *design for failure* in a transaction setting where more advanced concepts

in recovery management come into play – how to add diversity into the system and the provision for preserving as much progress-to-date as possible (omitted results).

Current transaction models such as *Web Services Transactions* (WS-Tx) [CCJ+04],[CCC+03] and *Business Transaction Protocol* (BTP) [FDF+04] are based on the conventional *Sagas* [G-MS87] long-lived transactions and seem to be geared towards centralised control and their coordination mechanisms require access to the local state of services, at least in recovery management. In addition, there is little, if any, support for partial results, forward recovery and omitted results. 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. A study of the existing transaction models has been given in Deliverable D24.28 of the DBE project [RKM06] and a detailed analysis can be found in D3.2 [RMK07b] of OPAALS. The problems with the current setting may not be as visible as they should be and this is in part because the current transaction models targeted for web services are advocated by a handful of large enterprises for which distributed coordination and local autonomy of the participants are not major issues of concern. On the contrary, central points of control and failure offer openings for violating the local autonomy of service providers, which large players are keen to exploit in continuing to dominate the transactional setting in Europe and elsewhere.

These are issues that need to be addressed explicitly in providing support for open e-business transactions in DEs. Harnessing the complexity of the underlying service compositions and capturing the call interplay between services, is a first step for increasing the confidence in a successful outcome in a transactional environment.

In what follows we describe the key transactional aspects in the DE architecture. We start by showing how distributed long-running transaction involving a number of underlying service executions can be coordinated at the design level. Then, we give a component-based design of the local agent at each participating platform, focusing on the structure of the local coordinators. We briefly outline how the complex interplay between service invocations between local coordinators is formally analysed to identify the behaviour patterns the service compositions inside a transaction should follow, in order to increase confidence in a successful outcome prior to deployment. The formal model used can express true-concurrency and the compensations of concurrent actions are also themselves performed concurrently. Finally, we are concerned with handling data dependencies and describe an extended lock scheme that ensures consistency, drives the compensations routines in case of failure, and maximises concurrency in a transactional setting.

3.2 Designing transactions

In this section we are concerned with the distributed coordination of long-running transactions in terms of the underlying interaction-based service compositions. This will be covered at the design modelling level using tree structures and directed graphs for keeping track of dependencies that arise during execution. Dependencies may arise due to the ordering of execution (e.g. book a flight before booking a hotel at the destination) or data dependencies (e.g. the choice of transport depends on the remaining budget after booking flights and accommodation).

We have seen in the previous section that a business transaction between SMEs in a Digital Ecosystem is rarely a simple usage of a web service but rather involves a mixture of different levels of composition of several services from various service providers. This means that all participants should behave in a coordinated manner in order to execute a transaction effectively.

In our model for long-running transactions in digital ecosystems, described in detail in Deliverables D3.1 [RMK07a] and D3.2 [RMK07b], 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 which will be described in more detail in Section 3.3.

We have seen that one of the basic requirements of SMEs as attested by ITA's survey and experience in the Aragon region of SMEs in Spain is the provision for an infrastructure that supports service-oriented computing (see R1 on p.11). This comes as no surprise as it is not difficult nowadays to conceptualise businesses offering their products as a service that can be consumed or used by other companies either in isolation or as part of a chain of services in a larger business activity. The mix and match of services however does imply that these can be put together or composed in various modes depending on the specific business scenario being executed. For instance, a travel agency may want to query a number of preferred hotels in sequence and then proceed to book the one that best matches the customer's criteria. Another agency, or the same in other scenarios, may want to query a number of preferred hotels in parallel and book the first to respond that offers a close match to the customer's criteria. In other words, services from different service providers need to be composed in various modes if a range of B2B scenarios are to be covered.

Based on the latest work on an extended service-oriented architecture for a business environment [YPH02], [Pap03], [PTD+06], five different types of coordinators have been considered in our model for interaction-based service composition, 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. These are included in order to incorporate a number of different service composition types, most of which have already been proposed as part of the inception of Service-Oriented Computing, which are necessary if we are to cover a range of B2B scenarios - something that is necessary as far as requirement R3 for service composition by regional SMEs is concerned.

Sequential: Where the execution of a service is dependent on the previous one. This coordinator can handle sequential process-oriented service composition with provision for both Sequential with Commit Dependency (SCD) and Sequential with Data Dependency (SDD).

Parallel: This coordinator handles parallel process-oriented service composition covering Parallel with Commit Dependency (PCD), Parallel with Data Dependency (PDD) and Parallel without Dependency (PND).

Sequential Alternative: the services will be attempted in succession until one produces the desired outcome, as specified by some criterion (e.g. cost, time, etc). This coordinator is particularly useful for forward recovery as will be discussed in Section IV.

Parallel alternative: alternative services are executed in parallel and once a service produces the desired outcome, the rest are aborted.

Data-oriented: this coordinator handles data-oriented service composition and specifically deals with released data items within a transaction (between its sub-transactions) or partial results released between different transactions.

Delegation: this coordinator allows the whole transaction or a sub-transaction to be delegated to another platform, e.g. as a means of overcoming traffic bottlenecks or low bandwidth connections.

Note the introduction of the *data-oriented* coordinator is targeted specifically at incorporating the ability for realising *partial results* across transactions.

Figure 3.1 shows a transaction tree with four basic services whose order of execution is determined by the five coordinator types employed. We have adopted the notation of [PDH+96] extended with a symbol for the data-oriented coordinator (labelled by d1 in the figure).

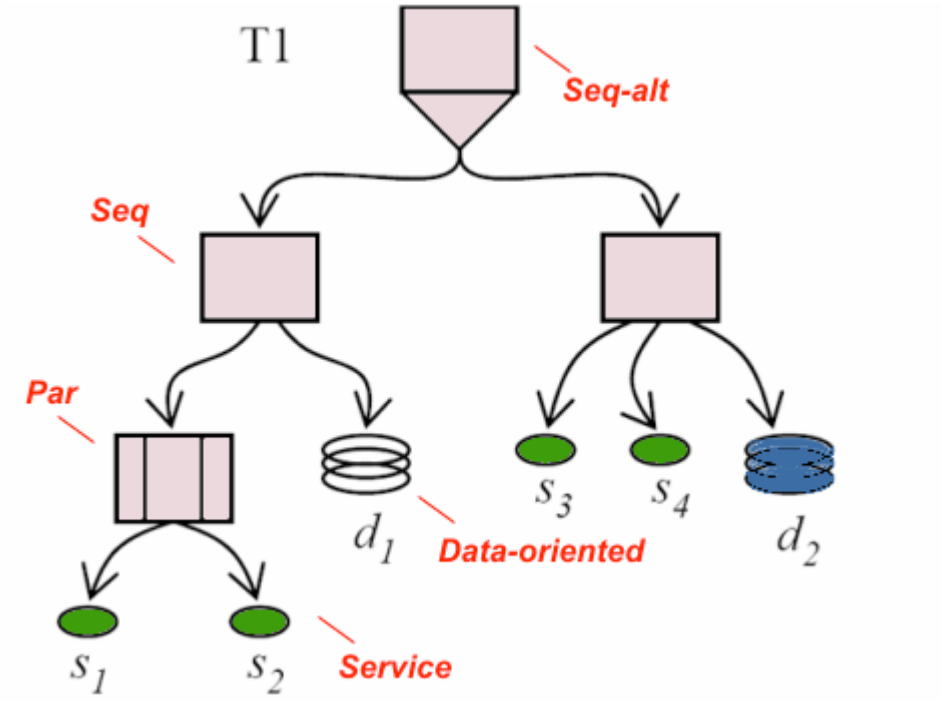


Figure 3.1 Transaction in a tree structure

The scenario described in Fig. 3.1 has also been used in deliverable D3.1 [RMK07] and is rather simple, nevertheless the transaction in question is complicated enough to illustrate the key ideas of our formal approach (and allows the reader to follow on from the description of the transaction model in D3.1). In the transaction tree of Figure 3.1 we have five coordinator types that say in what order the corresponding services are to be executed. The basic services (e.g. s1, s2, s3) appear on the leaves of the tree and get executed in the order dictated by their parent coordinator. It can be seen that a coordinator can also be a child of another coordinator, for example the parallel coordinator of s1 and s2 (marked by "par" in the figure) is also a child of the sequential coordinator (marked by "seq"). This shows that there are different levels of composition and the model we will be describing is in fact an open nested transaction model in which service composition can be nested inside other composition as shown in the transaction tree of Figure 3.1. This is an important distinction from other traditional models for long-lived transactions which are linear and sequential in execution, such as the well-known *Sagas* [G-MS87].

The services s3 and s4 in our example are children of a sequential coordinator and hence the service s4 can only be executed after s3. In other words the execution of s4 is dependent on the (successful) execution of s3. We note that these services may be provided by the same participant, e.g., they are services of a single SME, or by different participants. The latter case means that this way of setting up transactions also covers the case that a participant takes part on a transaction with more than one service. In either case, there is a sequential dependency between the two services and this may be due to the order of execution (e.g. book hotel after booking the flight) or due to a data dependency between s3 and s4, i.e. s4 has to use the results released by s3

The transaction tree is set up by the participating entity that wishes to execute a business activity involving services from different service providers. To establish some terminology, this entity in a transactional environment is called the Initiator while the rest of the service providers and/or consumers are called Participants. The rationale is that the Initiator of a transaction sets up the corresponding tree as it knows which partners offer which services in a given domain, and which of those services it needs to use to perform a specific business activity. In other words, they are companies that are already doing business together.

In the example transaction tree shown in Figure 3.1 the Initiator is setting up a transaction in which the services s1 and s2 need to be executed in parallel – if they belong to the same Participant then the Initiator requests that they are executed in parallel; if they belong to different participants then the Initiator issues requests to both concurrently. The execution of these two services is in sequence with the data-oriented coordinator d1. This means that the result of the parallel execution of s1 and s2 is passed on to d1 which can release it to another transaction. Notice it does so before this transaction has committed (or, terminated its own execution – this is a partial result). Now if the result does not satisfy a pre-specified criterion or some failure occurs half way through the execution up to the release by d1, the sequential alternative coordinator at the root of the tree (marked by "seq-alt" in the figure) says that execution will continue with the branch on its right. So if the left branch fails for whatever reason (service unavailable, results does not meet a pre-set criterion) then execution of the transaction continues with the right branch. In our example, the right branch includes a sequential coordinator and this means that execution continues with service s3, then s4 and then the result is released outside the current transaction by the data-oriented coordinator d2.

We have seen that the services s3 and s4 in our example are children of a sequential coordinator and hence the service s4 can only be executed after s3. In other words the execution of s4 is dependent on the (successful) execution of s3. Such dependencies are also important for the recovery mechanism since they determine what actions need to be taken and in what order during compensation of an aborted transaction. In our example, the sequential dependency between s3 and s4 has as a consequence that if s3 is aborted, then s4 must also be aborted. Therefore, it is important to have a way of keeping track of such dependencies that may arise during execution of a transaction. In deliverable D3.1 we introduced the *Internal Dependency Graph* (IDG) for representing such dependencies. The dependency between s3 and s4 is shown in the IDG of Figure 3.2.

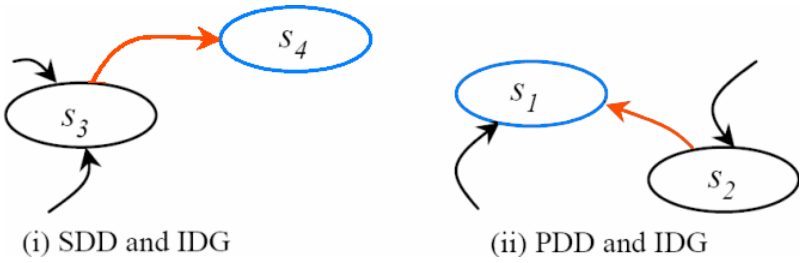


Figure 3.2 Internal Dependency Graph

In a highly dynamic and purely distributed environment such as a digital ecosystem for business, it is often the case that a subtransaction (an internal part of a transaction) requires access to a data item released (possibly as a *partial result*, which means before the releasing transaction has finished its execution or *committed*) by a subtransaction belonging to a different transaction. In other words, dependencies may exist not only within a transaction but also between transactions (which may take place on different platforms). For example, consider the case of (compensatory) subtransactions that release partial results in a conditional commit state [PDH+96].

To capture such dependencies we introduced the *External Dependency Graph* (EDG) in deliverable D3.1 [RMK07a]. This directed graph keeps track of dependencies between (services or coordinators of) different transactions. The log structure it provides can be used in the recovery routines for running a compensating procedure. This is necessary in a transactional setting and has been identified as a key requirement of regional SMEs (see requirement R4, p.11). Figure 3.3 shows part of the EDG for the transaction trees T1 (of Figure 3.1) and part of a new transaction T2. In this case, the data-oriented coordinators d1 and d2 of T1 release partial results that are required by d3 of transaction T2.

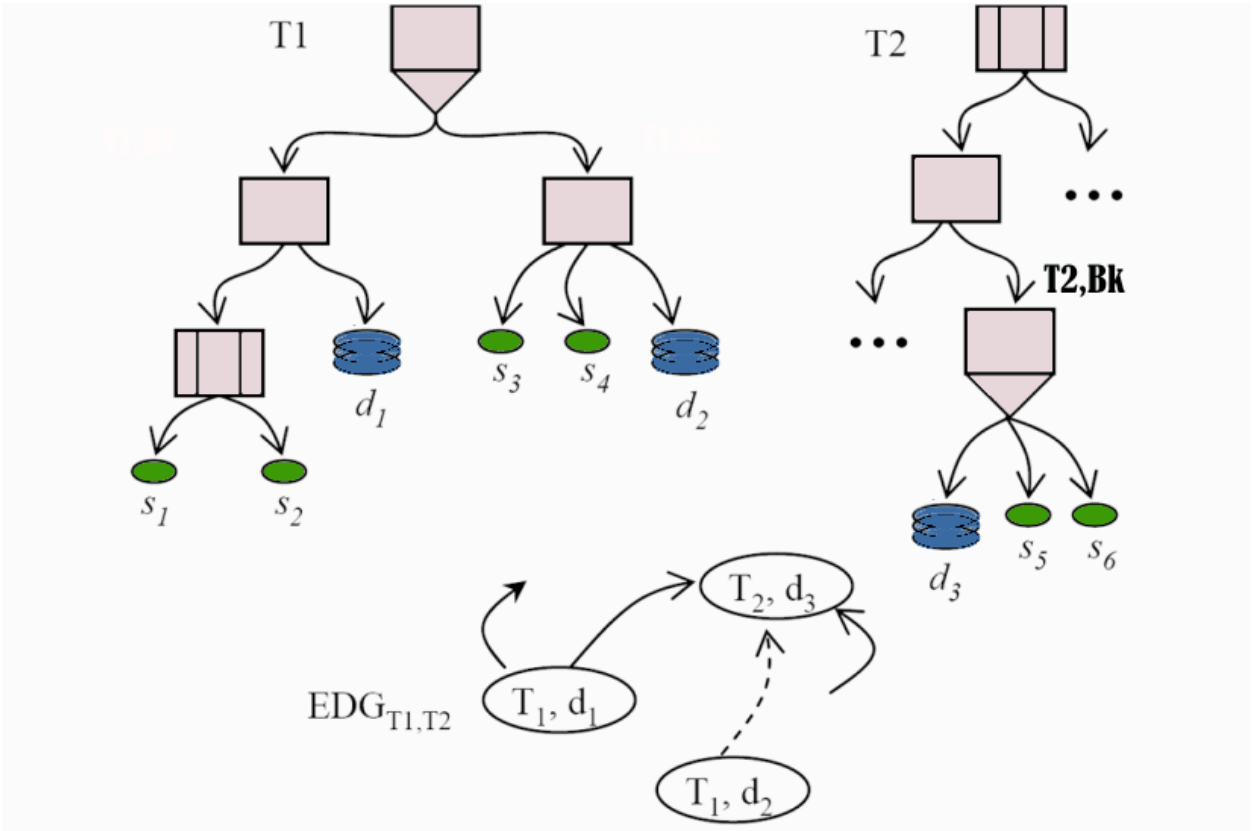


Figure 3.3 EDG for releasing partial results between T1 and T2

Now, if for some reason d1 (or any other subtransactions on which d1 depends, for that matter) was aborted, then d3 should also be aborted along with any sub-transactions of T2 which depend on it. This is because the results d2 will be using will no longer be consistent or recoverable. Based on the log information provided by the corresponding EDG and transaction trees, we would like to recalculate d3 based on the data items released by d2 and defer from aborting (at least part of) transaction T2.

So far we have described constructs necessary for setting up a distributed transaction that involves the execution of a number of services from different service providers. The transaction tree is the starting point for describing the flow of execution required for performing a given transaction. In other words, the transaction tree describes what services from different partners need to be deployed. The graphs (IDG, EDG) keep track of the dependencies that arise when the services are deployed and the log structures they provide are necessary so that each service provider can coordinate its services locally and have complete control over its own realisation level (data and service implementation). It is worth noting that loosely-coupled service is the basic premise of a service-oriented architecture that is sought after by SMEs in the regions (recall requirement R1, p.11).

The combination of these constructs, i.e., transaction tree, internal dependency graph, and external dependency graph, is the so-called *transaction context* as it is used to define the context in which a transaction is considered in terms of the parties involved, their services and the specific way in which they should deployed. The transaction context is used precisely to manage the deployment of the requested services, and this can be done in a distributed manner. So on top of the need for managing service deployment (requirement R2a, on p.11) it also allows a way to express the remote deployment of services whenever this is needed in a transaction (R2c, p.11).

It is also important to note that with this design each Participant only needs to know what input (calls to its services) it is expecting and what output it is obliged to provide (service response or new service calls) after executing its own services. In our example of Figure 3.3 all the data-oriented coordinator d3 needs to know is the information contained in the EDG, i.e., that it expects input from d1 of transaction T1 and failing that, it expects data from d2 of the same transaction. It does not need to know the reason for getting a result from d2 instead of d1 (what failure and where it occurred in T1) or what services were executed and in what order, e.g. whether s3 and s4 were executed in parallel or in sequence, whetherthey belong ot the same Participant of T1 or not, and so on. This design is based on what we call the *immediately before* and *after* principle. That is, every participating platform only needs to know what happens immediately before and immediately after its own services, which service/coordinator it expects results from and which service it passes result to, but no more.

It can be seen that this approach to coordinating distributed long-running transactions focuses on the *local* coordination of the services involved; each participant coordinates the execution of its own services and only communicates responses and requests, using web services standards or extensions thereof, without revealing any implementation details relating to the local state of execution of each service. Therefore, the support for coordinating transactions in the core DE architecture is stateless – requires no information about the state of the underlying services. The responsibility of managing the state of execution lies with each participant who can reveal only what it wants to reveal about the services it offers in the digital ecosystem. In the following section we will see how the local coordination can be achieved at the deployment level with a component-based design of the so-called Local Agent of each participant.

3.3 Local agents component-based design

In this section, we are concerned with the infrastructure necessary for coordinating services *locally* within the transaction contexts described in the previous section. A component-based design for the coordination of open long-running transactions over a P2P network supporting a community of SMEs is outlined. In particular, we describe the use of a *local agent* at each node, at each participant in a transaction, that allows for the underlying services to be coordinated in a truly distributed manner.

An instance of the local agent described here is what each SME would need to have on their computing infrastructure in order to participate in the digital ecosystem. The component-based design of the local agent we describe provides a lightweight solution for the coordination of the underlying service compositions and their execution, which is a requirement expressed explicitly by participating SMEs (see R2, R3, R4. p.11). In what follows we outline the basic ideas, more details can be found in deliverable D3.2 [RMK07b].

At the heart of the local agent structure is the *Local Coordinator* component which uses the information of other local agents to orchestrate the necessary interactions involved in performing long-running transactions, which as we have seen correspond to complex long-lived business activities.

Our design is based on the premise that web services do not need to make themselves known (in naming or, more importantly, in any particular implementation) to a centralised coordinator for the transaction. For this reason, we have designed an agent for each platform (owned by an SME) which handles all the communications with its services by applying the formal model (outlined in Section 3.4) on one hand and keeping the information about its local services and any external web services (those belonging to other platforms) on the other hand. Our live agent needs to have enough knowledge about its local web services to be able to deploy them based on the particular transaction protocol (considerations about transaction protocols have been analysed in deliverable D3.1 [RMK07a]).

Figure 3.4 shows an overview of the local agent structure. This 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.

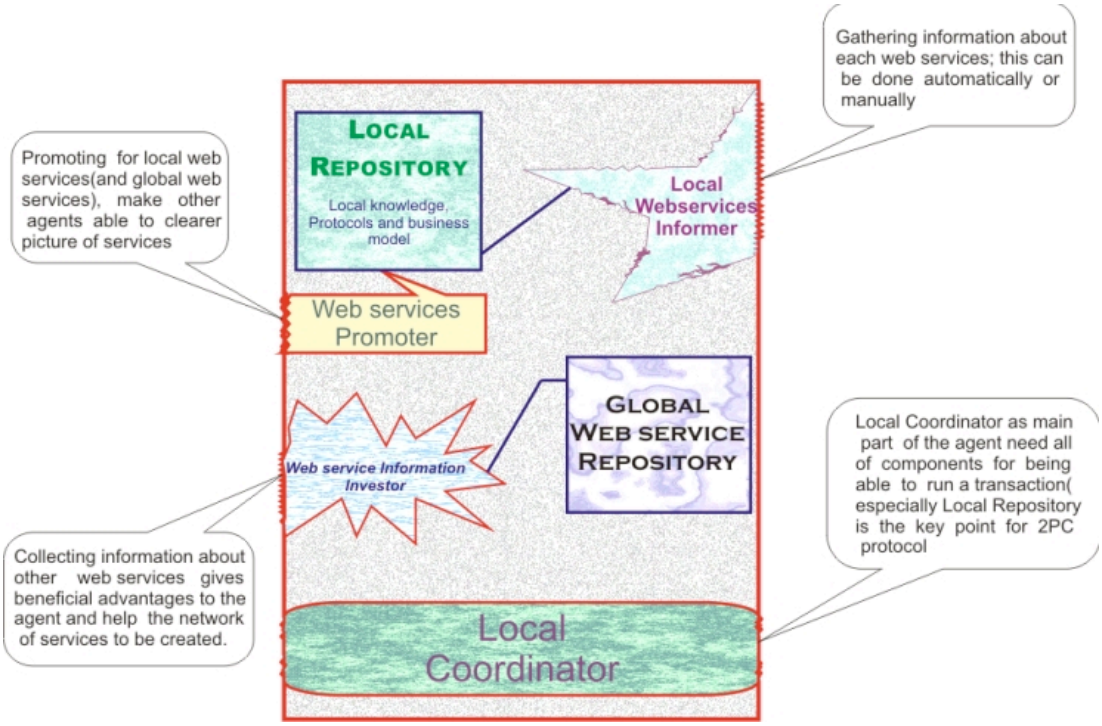


Figure 3.4 Overview of a local agent

In this report we will focus on the local coordinator. The other components of the local agent have been described in detail in Deliverable D3.1 and D3.2 and it perhaps suffices to say here that they are concerned with making services known to the rest of the world (to other SMEs on the network) and gathering information about services that are available in the outside world.

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-phase commit (2PC) protocol. This normally can be handled by a transaction context in response to receiving a transaction request (transaction 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. Recall that it is the Initiator that sets up a transaction context, as discussed in the previous section. This makes it possible to create the transaction script based on the knowledge of the other agents' web services. Finally, it also requires the interface from its local web services to able to invoke them (see Figure 3.5).

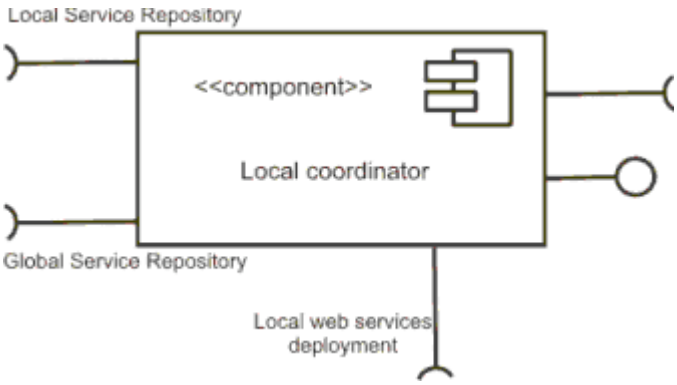


Figure 3.5 Local Coordinator

Figure 3.6 shows all components of a local agent (we consider this schema for any service provider in the system). The local agent by using two repositories (Local and Global service repositories) tries to provide detailed information about local web services, but also general information about other web services. This enables the Local Coordinator to invoke its local web services based on different protocols (for example *two-phase-commit* (2PC) or *three-phase-commit* (3PC)) and on the other hand, by using general information about other (remote) web services, in some sort of XML description, such as WSDL or SDL[4], it can create the transaction context (requirement).

The Local Service Repository should be updated by the Local Web Services Informer (any changes or updates can be effected on the Local Service Repository). Meanwhile the Local Service Repository can promote its services to other agents through Web Service Promoter.

The Global Service Repository can be updated by the Web Service Information Investor and at the same time, can promote these web services (which are stored in Global Service Repository) to the other agents (any changes will be promoted too, and in this way other agents can update their Global Service Repository).

[4] SDL (Service Description Language), it is a standard description language which is introduced in DBE project and it is popular in Digital Ecosystem community.

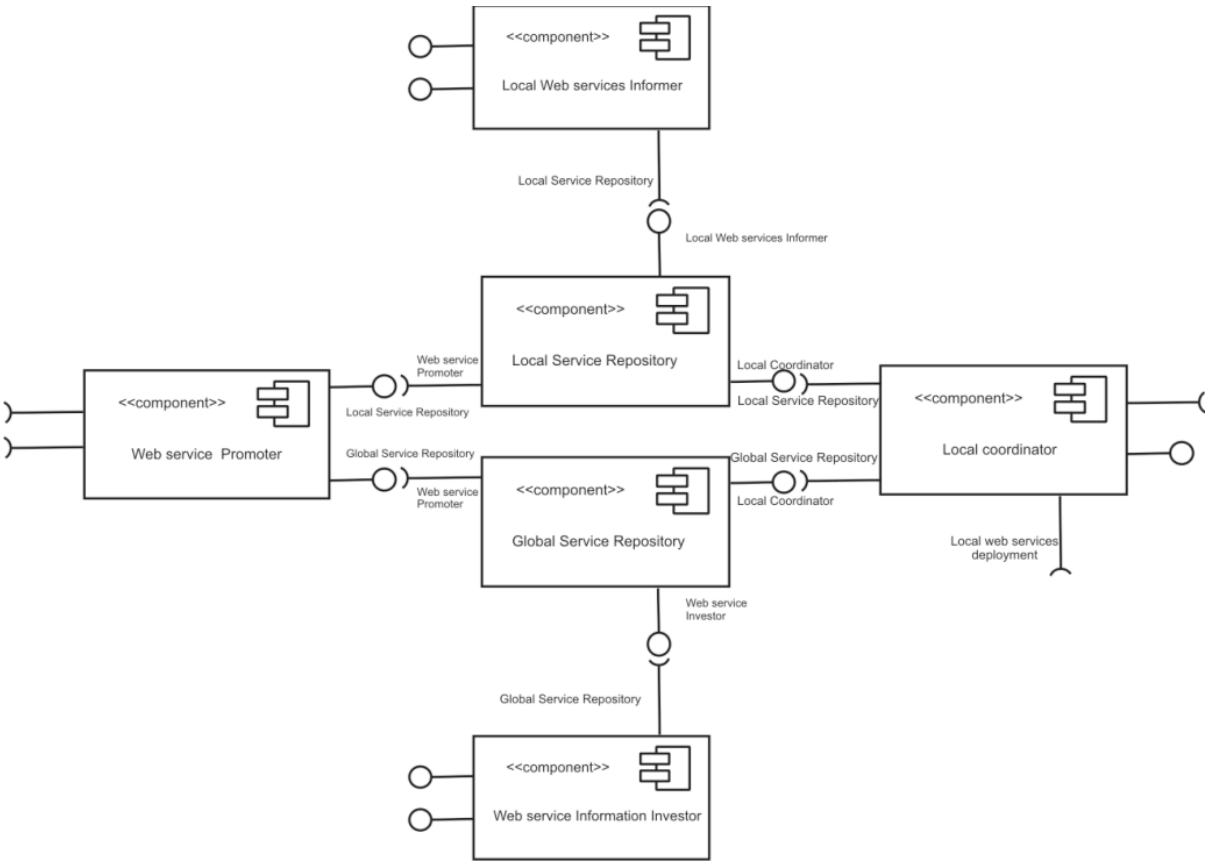


Figure 3.6 Local agent components

Figure 3.7 shows how communications between components of agents can improve the performance, can keep all agents' repositories updated and can provide enough information for the Local Coordinator of agents to run transactions.

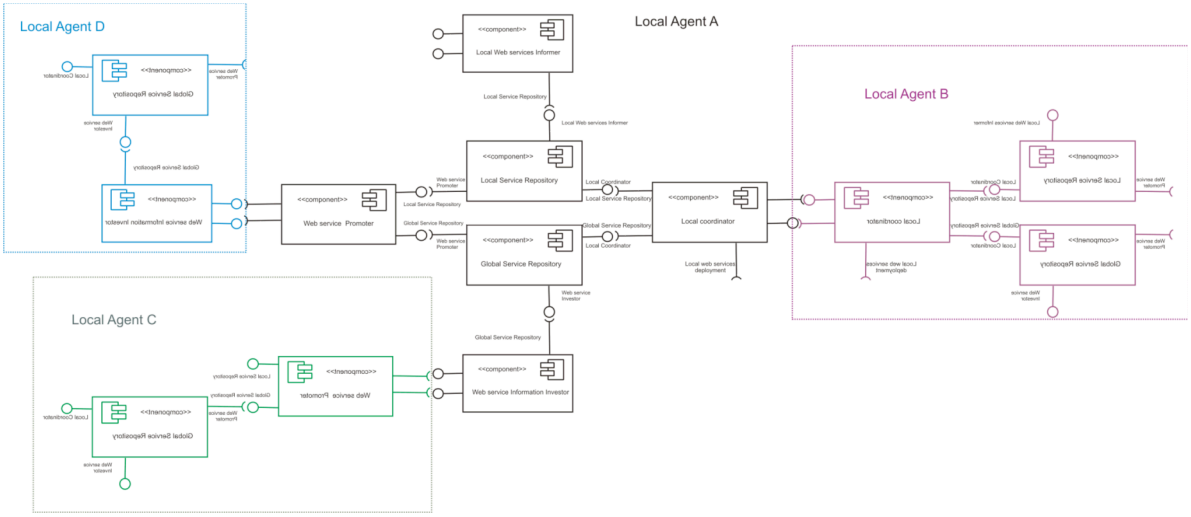


Figure 3.7 Overview of the multi-agent environment

It can be seen that this design allows the deployment of services in a loosely-coupled manner. All service invocations have to go through the Local Coordinator of each platform and do not have access to the Local Service Repository of the platform's agent which is the component responsible for executing the

particular service requested. In other words, all information about the local data and implementation (state of execution at the realisation level) is shielded behind the Local Coordinator and its Local Service Repository. This makes for a *stateless* architecture that supports distributed long-running transactions without violating the local autonomy of the participants. In fact, the service coordination support for transactions in the DE architecture is *technology-agnostic* so participants can use their technology of choice so long as they provide standard interfaces to their implementation that allows communication. This is a general requirement in a digital ecosystem but such a need has also been stated explicitly by surveyed SMEs as discussed in Section 2.1.5 (see requirements R1 and R4). The stateless architecture also covers the necessary compensation procedures which will be discussed in subsequent sections.

We conclude we a few notes on the implementation aspects of the local agent design. As the current crisis on experiencing serious performance deficiency from having to wait for a server's response during web service invocations (over the network such calls can take unpredictable lengths of time), JAX-WS 2.0 provides a new asynchronous client API, which can be applied for many clients, especially interactive ones such as JFC/Swing-based desktop applications.

By using this API, programmers are able to trust in the JAX-WS runtime to manage long-running remote invocations for them (they do not need to build threads on their own). In this way, asynchronous methods may be used in conjunction with any WSDL-generated interfaces as well as with the more dynamic Dispatch API. Sun Microsystems proposes two usage models:

- Polling model: you make a call. When you're ready, you request the results.
- Callback model: you register a handler. As soon as the response arrives, you are notified.

Note that when a WSDL document is imported, asynchronous methods are required to be generated for any of the operations defined in the web service. Furthermore, asynchronous invocation support is entirely implemented on the client side, so no changes are required to the target web service and there is no violation of service-oriented architecture or the SMEs' local autonomy. The prototype implementation of the University of Surrey has considered a few practical scenarios in which both usage models can be seen in practice. Further to this, as we rely on the software agent in our design in contrast to the conventional application, abstractly the life cycle of this agent is not limited and later on (in the next version), we plan to add mobile agents for additional performance and flexibility to our design.

It should be noted that this prototypical example has been used, and reported here, as a proof of concept. It does show that the component-based design for the transaction support in the DE architecture is feasible for implementation. In addition this has been done using standard technologies (SMEs requirement R6, p.11) and in a way that supports asynchronous communication in a transaction setting (SMEs requirement R7, R8 on p.11). Irrespective, this is a prototypical example and we feel SMEs can adapt the proposed implemented solution to what best fits their software and hardware constraints and expertise.

A prototype implementation of the local agent in the OPAALS transaction model has been demonstrated by means of a simple scenario given in Deliverable D3.2 [RMK07b] and is currently being extended to cover more advanced concepts behind our transaction model.

3.4 Reasoning about transaction behaviour

We have seen that distributed transactions involve complex interactions, in terms of service invocations through the local coordinators of each participating platform, which need to be coordinated in a principle manner to ensure that when the transaction is executed it will produce the desired outcome, or in software engineering terms, it will exhibit the *desired behaviour*. This concerns the case that internal actions of a

transaction (called subtransactions in previous sections) all succeed during execution leading to successful termination of the transaction as a whole. We refer to this part as the *forward behaviour* of a transaction. We have seen that coordinating service invocations in a transactional setting comes with the additional dimension of compensation routines which need to be executed whenever some failure is encountered during the forward behaviour. This means that we have to be able to go back and effectively 'undo' the effects of previously successful internal actions if a failure in some subsequent action makes this necessary. In order to handle the added complexity, a formal foundation for long-running transactions is required to give a through understanding of the dependencies that arise due to the necessary orderings in the corresponding service invocations and their compensating actions.

In this section, we describe the formal semantics of our model that provides a language for expressing forward behaviour (when things go well) as well as compensating behaviour (when some failure occurs). It should be noted that this along with the mathematical development that goes into the formal construction has been described at length in Deliverable D3.2 [RMK07b]. We find little reason to repeat this material here. Instead, we outline the basic ideas behind the formal modelling approach and focus on how it can be used to enhance the transaction model described so far.

In what follows we briefly show how the formal foundation for long-running transactions captures the patterns service compositions should follow to increase confidence in a successful outcome. We hint towards how it can be used to reason about the behavioural scenarios of the required service compositions involved in a transaction, and identify connections of the more general mathematical theory involved to models of *interaction computing*.

Admittedly, one of the most challenging aspects in modelling interactions between parts of a system in computer science has to do with capturing concurrency. The formal semantics for long-running transactions introduced in Deliverable D3.2 [RMK07b] and subsequently published in [MRZ+08], [MRK08], draws upon a generic model of *true-concurrency* [Shi85], [Shi97], [Maz88] which means that we do not identify concurrent execution with the nondeterministic interleaving of the actions involved, as is the case in process algebras such as the Communicating Sequential Processes (CSP) [Hoa85] or the Calculus for Communicating Systems (CCS) [Mil80]. The latter model concurrent actions by considering they take place in *either order* and typically use a sequence, or *trace*, to model the actions that take place on a process or component. This means that they can faithfully model a single access point (e.g. on a service interface). However, in a transactional setting there are a number of different participants and hence a number of access points (through the respective local coordinators) may be active at the same time. It transpires that rather than a single sequence we need a number of such sequences, one for each participating platform. This motivates the adaptation of Shields' *vector languages* [Shi97] which are sets of vectors, each capturing a specific action (in fact, a set of concurrent actions, or in mathematical terms, a simultaneity class of actions, which are unordered and so can take place *in any order*).

The set out of the notation has been described in detail in Deliverable D3.2 [RMK07b] and we refer the interested reader to that report for the full details. Here, we outline the basic ideas behind the formal modelling approach and focus on how it can be used to enhance the transaction model described so far. We also highlight the way concurrency is handled. The support for parallel execution and asynchronicity was identified as a requirement, R9, by SMEs in ITA's study which we outlined in Section 2.1.5 Each vector has a coordinate dedicated to each participating platform, which is used to record the actions that particular platform engages in. In other words, each vector provides a 'snapshot' of transaction behaviour in the sense that is says what actions have taken place and on which platform (local coordinator).

For example, the *transaction vector*

$$(s_1, s_2, \Lambda)$$

describes that portion of behaviour in which both s_1 and s_2 have happened on the corresponding platforms while the vector

$$(s_1 s_3, s_2, \Lambda)$$

describes an occurrence of s_1 followed by an occurrence of s_3 on the platform corresponding to the first coordinate, and an occurrence of s_2 on that of the second coordinate. Nothing has happened on the platform corresponding to the third coordinate.

It can be seen that the two example vectors differ in s_3. In fact, the second vector is obtained by concatenating the first (which describes what has happened until then) with the vector (s_3, Λ, Λ) which describes the action s3 (see Definition 2 of *column* vectors in D3.2 [RMK07b]). Transaction vectors are essentially *tuples of sequences* and consequently, this is done by lifting the usual operation of concatenation between sequences onto vectors. This is done by applying coordinate-wise, i.e. on the resopective coordinates and in our example produces the following.

$$(s_1, s_2, \Lambda).(s_3, \Lambda, \Lambda) = (s_1 s_3, s_2, \Lambda)$$

This already provides an ordering between such vectors which is pivotal in understanding the dependencies between actions of each participant in a transaction. The ordering is again based on the usual prefix ordering on sequences and is lifted onto vectors in the same way as before, i.e. by performing the operation coordinate-wise.

$$(s_1, s_2, \Lambda) \leq (s_1 s_3, s_2, \Lambda) \text{ since } s_1 \leq s_1 s_3 \text{ and } s_2 \leq s_2 \text{ and } \Lambda \leq \Lambda$$

Thus we have an ordering between the two vectors. This says that the smaller vector describes an earlier part of the behaviour described by the larger vector. In our example, ($s_1 s_3$, s_2, Λ) describes the occurrence of s_3 in addition to what the smaller vector described already.

A long-running transaction involves a number of actions and modelling each in this way results in a set of transaction vectors, which is the so-called *transaction language*. In mathematical terms, the set of all vectors that can be formed over a given sets of actions for each platform is monoid (that is, a semi-group with identity) under the operation of concatenation, with identity Λ. It is also a partially-ordered set undere the operation of prefix ordering, with bottom element Λ. Subsets of the set of all possible vectors form *transaction languages*. The idea is that the 'language' of vectors associated with a given transaction describes the necessary constraints on the orderings of the underlying service invocations.

In a partially ordered set some elements are ordered but some may be incomparable. For example, consider the vectors $\underline{u} = (s_1, \Lambda, \Lambda)$ and $\underline{v} = (\Lambda, s_2, \Lambda)$ for which neither $\underline{u} \leq \underline{v}$ or $v \leq \underline{u}$. Such vectors describe either alternative behaviour (there is a choice between the last actions that went into forming each) or concurrent behaviour (the last actions that went into forming each are concurrent). Any pair of incomparable vectors stands in one relation or the other, and this is determined by what other vectors are in the set of vectors associated with a given transaction.

If the incomparable vectors are bounded above – in other words, if they describe earlier parts of some common later behaviour – then they describe concurrent behaviours. If they are not bounded above, then they describe alternative behaviours. It is important to stress that this is determined by context, by what other vectors are included in the set for a transaction.

This is illustrated in Figure 3.8 which uses Hasse diagrams to depict the order structure of different sets of transaction vectors for a transaction with 3 participants. It can be seen that s1 and s2 are sequential (s2 can only be activated after s1) in Figure 3.8(i) while they are mutually exclusive (alternative) in Figure 3.8(ii) and they are concurrent in Figure 3.8(iii).

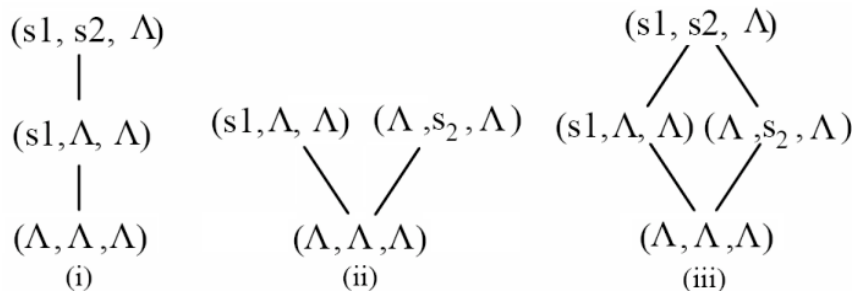


Figure 3.8 Order structure of transaction vectors

Notice that the set of vectors in case (i) does not include the vector (Λ, s_2, Λ). This, in addition with the fact that (s_1, s_2, Λ) is included, implies that s2 can only happen after s1 has (sequential dependency).

The set of vectors in case (ii) does not include (s_1, s_2, Λ). This has as a consequence that the vectors $\underline{u} = (s_1, \Lambda, \Lambda)$ and $\underline{v} = (\Lambda, s_2, \Lambda)$ are not bounded above in this case. Hence, the actions s1 and s2 are independent but do not take place consecutively in this case (one immediately after the other). This implies that there is a choice between doing s_1 and doing s_2 on the respective coordinates (alternative execution).

In case (iii) where the vector (s_1, s_2, Λ) is included, the vectors are bounded above and this implies that they describe the concurrent execution of actions s_1 and s_2 leading to the behaviour described by the vector (s_1, s_2, Λ). This is indicated by the familiar lozenge shape (or diamond) found in *Asynchronous Transition Systems* (ATSs) [Shi85], which marks the characteristic structure of a finite lattice [DaP90]. The incomparable vectors sitting at the middle of the lozenge are both available after the same behaviour (that is $(\Lambda, \Lambda, \Lambda)$ in this case) and occur consecutively leading to the behaviour described by the vector sitting at the bottom of the lozenge shape, i.e. (s_1, s_2, Λ).

Figure 3.8 might be instructive with regard to the subtle distinction between *independence* and concurrency, which was discussed in detail in Deliverable D3.2. Independent actions are concurrent only if they are both offered after the same behaviour (are both enabled at the same point during the course of execution of a transaction). Otherwise, they may be mutually exclusive or even sequential.

So far we have outlined the very basic construction that models forward behaviour. We now turn our attention to compensating behaviour. This is again based on lifting the well-known operation of *right-cancellation* on sequences onto vectors, and this is done by applying it coordinate-wise. When applied onto a sequence, right-cancellation produces a sequence in which the last element is missing. In other words, the right-cancellation operator '/' chops out the last element of a sequence.

In our example transaction vectors, and if we apply '/' on the larger vector we have,

$$(s_1 s_3, s_2, \Lambda) / (s_1, s_2, \Lambda) = (s_3, \Lambda, \Lambda)$$

It can be seen that the application of the right-cancellation, between a vector and its immediate predecessor(s), produces a column vector that can be shown to describe the very last action that went into forming the vector in question. In compensating for a failed transaction we need to perform the last action first. More generally, the compensating sequences must be performed in the reverse order to that of the forward sequences. The right-cancellation on transaction languages produces the action that must be performed first, in case the vector to which it is applied describes the last action in forward behaviour.

In other words, and simplifying somewhat, forward behaviour can be captured using concatenation and compensating behaviour can be expressed using right-cancellation. It might be worth noting that no additional notation is required to deal with compensating behaviour.

The transaction language shown in Figure 3.9 can be obtained from the transaction tree given in Figure 3.1 of Section 3.2. In looking at the Hasse diagrams and moving upwards we can see the forward behaviour leading to the two maximal vectors (one for each diagram) and this is done by starting from the empty vector and performing a series of concatenations with the column vectors representing the desired actions.

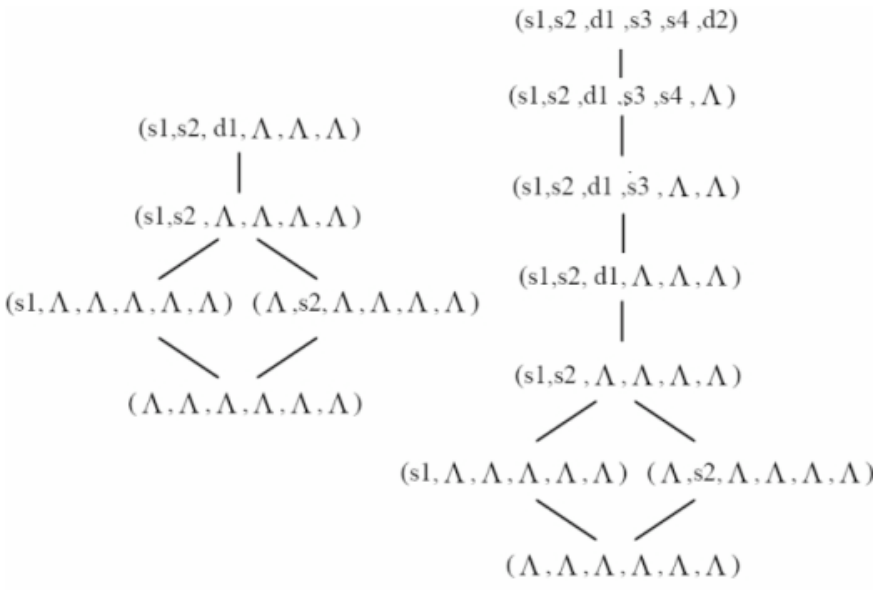


Figure 3.9 Transaction language for the transaction tree of Figure 3.1

Now if some failure is encountered half way through, we stop moving upwards and start moving downwards in which case we have the a series of right-cancellations, cancelling out each forward action in turn, and until we reach the empty vector. There are two Hasse diagrams in Figure 3.9, one for each branch of the sequential alternative coordinator used in the design of the transaction in Figure 3.1.

This formal analysis might appear tedious and is not to be performed by hand. However, transaction languages lend themselves naturally to computing machines as all the operations on them comedown to comparing the respective coordinates and are performed coordinate-wise. It can also be seen that this formal analysis provides a clear understanding of the various execution paths that are possible and this is particularly useful in addressing more advanced concepts in recovering transactions, such as forward recovery and omitted results.

Further, we have seen that transaction trees essentially describe scenarios of execution and these can be captured using UML interaction diagrams. In previous work [MKS07] we have described how languages of vectors can be obtained from UML2.0 sequence diagrams [OMG04]. This extension has as a consequence that we can perform formal reasoning against the order-theoretic properties of transaction languages, such as *discreteness* and *local left-closure* described in Deliverable D3.2 [RMK07b], and identify scenarios of interaction that may lead to pathological or undesirable behaviour. This aspect of the work on refining transaction scripts has been reported in a recent paper [MRK08].

The formal semantics we have given to long-running distributed transactions allows internal activities of a transaction to communicate and can capture parallelism. Unlike other approaches that use process algebras to model long-running transactions such as [BMM05], [BHF05], in our approach there is no need

to consider different sequences of actions within a transaction and then compose them in order to model concurrency. Concurrency is handled in terms of the actions themselves, and there is no need for all actions within a transaction to be independent (and isolated). This is one of the benefits of opting for a non-interleaving semantics, based on [Shi85] [Shi97], [Maz88], [Mos05] as it allows modelling *true-concurrency* between actions in a long-running transaction.

The proposed formal framework provides a formal language in which transactions can be expressed and the coordination of the underlying service executions can be checked prior to deployment. With modern software applications there is an increasing need for parallelism and asynchronous communication (see SMEs requirement R8 on p.11). The formal semantics for long-running transactions we have developed allows for concurrent execution of internal actions (subtransactions) and, perhaps even more importantly, the compensations for concurrent actions are also executed concurrently.

Further, we will see that the proposed extension to the transaction model with locks allows for traceable transactions. This has the implication that we can guarantee consistency of the model at all times and in combination with the analysis of transaction languages we can work with a much more sophisticated recovery mechanism. This includes the preservation of as much progress-to-date as possible (*omitted results*) and adding diversity into the framework by means of alternative scenarios for completing a transaction.

So far we have been mostly concerned with laying the formal foundations for transaction support in the core DE architecture. The mathematical theory underpinning the formal semantics is more general and can have wider applicability. In particular, transaction languages generate a certain type of algebraic automata, which exhibit true-concurrency, and the construction that makes this possible has a number of interesting properties. In fact the algebraic properties of the construction go beyond capturing orderings of service invocations (and compensating actions) within transactions and can inform the more general interactions considered in the work of WP1 concerning an abstract model of *interaction computing*. WP1 has in fact begun to work on algebraic automata theory to model metabolic pathways, and this work is being continued in the BIONETS project. We think it is worth pursuing a stronger connection with our work in the interest of developing a more general framework for interaction computing.

3.5 Handling data dependencies

We have seen that the interactions (e.g. in terms of service invocations) in a transactional setting can be rather complex as the service invocations involved are usually interrelated which means that dependencies arise during execution. Such dependencies may become unmanageable when the volume of services interactions increases. We introduced the IDG and EDG directed graphs for keeping track of such dependencies within a transaction context. We have also developed a formal model from which transaction scripts can be derived, which allow computers to understand such dependencies and, to a certain extent, reason about the possible scenarios of interactions involved in a given transaction context.

When talking about service composition in SOC there are two main aspects that need to be considered: order (of execution) and data. So far we have outlined our ideas on determining (and validating) the required orderings on the underlying service invocations. In this section we will be concerned with data dependencies that may exist within and outside a transaction. In Deliverable D3.1 Of OPAALS [RMK07a] we have described a fine-grained lock scheme which extends the conventional *Shared/eXclusive* (S/X) Lock model in order to allow for exchange of data and results both within and across transactions (the latter is often referred to as *partial results*). Here we outline the basic ideas behind our extended lock scheme for handling data dependencies and driving the recovery mechanism if some failure at some point during the execution of the transaction makes this necessary.

In conventional S/X locks models [Dat96], [Gra03], a data item that is locked by a Shared lock (S_Lock) can be accessed by other transactions or subtransactions of the same transaction. Whilst an operation is performed on a data item then this item is locked by an eXclusive lock (X_Lock) and access is restricted to the owner of the item. Following an S/X lock model for transactions, the X_lock is only released (converted to S_Lock) when the transaction commits. This does not allow for any exchange of data *during* a transaction (before 'commit'). We have seen (Section 3.1) that transactions within a digital ecosystem for business are *long-running* in nature and include the execution of a number of services (from different providers). This means that a transaction in this setting can take minutes or hours or even days to complete ('commit').

It transpires that applying a conventional S/X lock model for concurrency control and consistency in this transactional environment brings unacceptable limitations. It would not allow for exchange of results both within and between transactions, and this severely limits the range of business scenarios that can be covered with distributed transactions in the digital ecosystem. For instance, subtransactions of one transaction cannot access data produced by other subtransactions of the same transaction. Hence, releasing data within a transaction is prohibited.. Further, subtransactions of different transactions cannot exchange results before the transaction they belong to commits. This is often referred to as the problem of *partial results* and appears in most business scenarios in a B2B context.

In view of such limitations of conventional lock models for a transactional environment, we propose an extended lock system which has been designed with distributed transactions in a DE for business in mind. The idea is to relax the rigid eXclusive lock (X_Lock) by introducing intermediate locks that give leverage over access to data items within and between transactions. The additional locks drive the creation of the log-based graphs IDG and EDG, introduced in Section 3.2. The extended lock system allows for the concurrent execution inside a transaction but also across transactions and as a result maximises potential concurrency in our transaction model. Additionally, it has a significant role to play in ensuring consistency of the transaction model at all times, especially when it comes to recovery management, where the locks are intrinsic to the forward recovery mechanism and the way we deal with omitted results.

In what follows we outline the key ideas behind our extended lock scheme for distributed transactions. Fulle details of the scheme can be found in Deliverable D3.1, and have been subsequently published in [RMK07d].

In most B2B scenarios, subtransactions of a long-running multi-service transaction need to access and share results before the transactions commits. The use of X_Lock on data items inside a transaction implies that these can only be accessed by the owner of the lock. In our approach, we use an *Internal* lock, denoted by I_Lock, to relax X_Lock and allow subtransactions to access and/or modify data items inside a transaction before 'commit'.

The introduction of this intermediate lock, I_Lock, in between S_Lock and X_Lock as shown in Figure 3.10 makes deployed data items available to other subtransactions of the same transaction. When a subtransaction needs to release a result before the transaction it belongs to commits, then it uses I_Lock as a relaxed version of X_Lock. An item that is locked by the internal lock I_Lock can be accessed and modified by other subtransactions of the same transaction. Any subtransaction that wants to use it can do so, provided it adheres to the concurrency control principle and converts I_Lock to X_Lock while it does so (this is necessary to ensure consistency at all times), and then converts it back to I_Lock. The data item is then again available to other subtransactions that might require it.

In this way, the internal lock in combination with the corresponding IDG, allow for uncommitted results inside a transaction to be accessed and modified a number of times and for the duration of the transaction. This allows for a far more flexible service composition style in a transaction involving the execution of a number of services (recall SMEs requirements R2, R3, R4). Moreover, this mechanism facilitates the concurrent execution of different parts within a transaction. In a certain important sense, it allows for true parallelism in transaction execution. Following a conventional S/X lock model, each subtransaction (in fact, it would only be a transaction as there it would not make sense to have internal structure) would have to

execute in turn and commit before releasing tis result to other subtransactions. On the contrary, in our approach the various subtransactions can be potentially executed in parallel and only need to be synchronised on releasing/sharing data items before commit. All data items locked by I_Lock in our model are converted back to S_Lock only when the transaction commits. This means that the results of the transaction can then be used by other (different) transactions that require them.

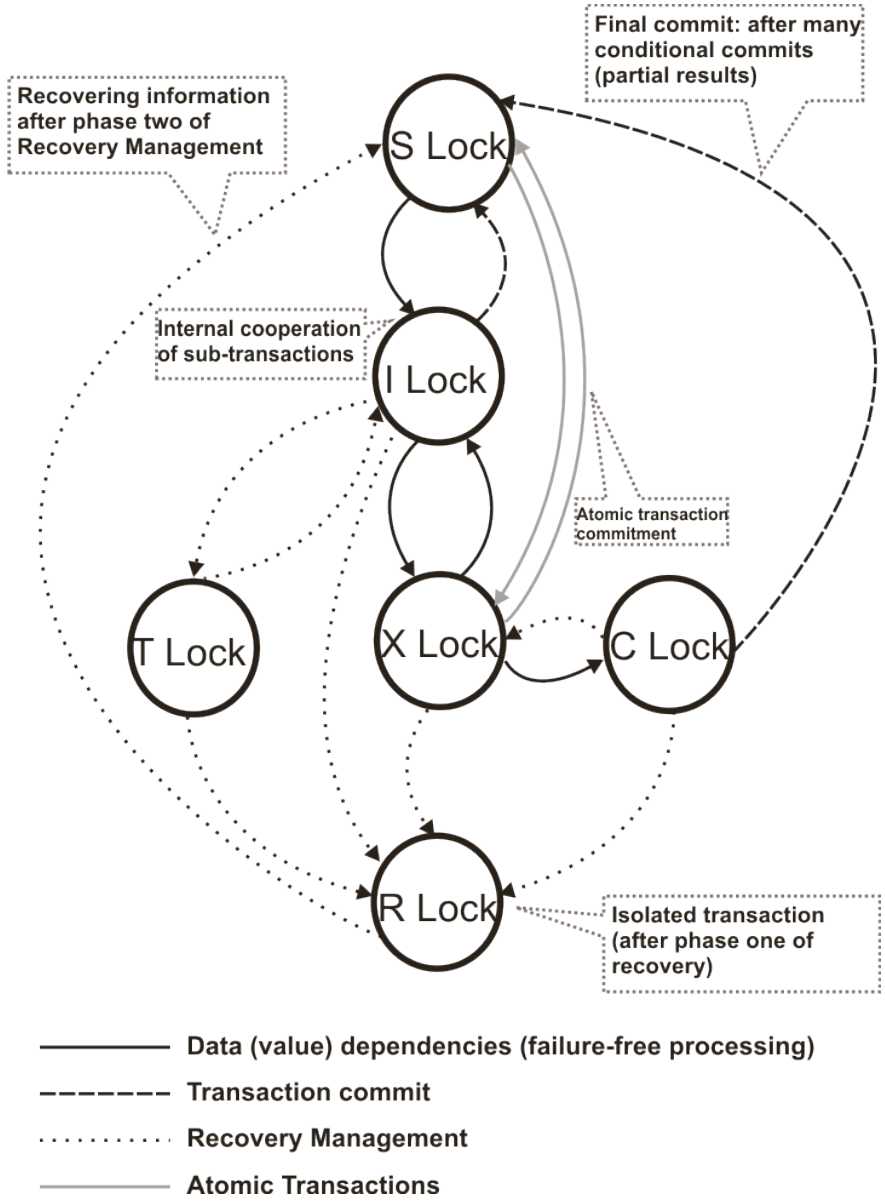


Figure 3.10 Extended lock schema for data consistency in distributed transactions

In distributed transactions within a B2B context of a digital ecosystem, it is often the case that transactions need to release results to other transactions before the transactions involved commit. We have seen that the SMEs surveyed in the Aragon region indicated their need for various types of service composition and the constructs to be able to manage their deployment (see R3, R4, R2 on p.11). Following a conventional S/X lock model such cases cannot be addressed as in these models a transaction can only release results after 'commit'. This limitation not only prohibits transactions from being executed concurrently (since, at best, the second has to wait for the first transaction to commit), but also can prevent

a number of different transactions from reaching their target and thus also abruptly stop the corresponding long-lived business activity.

The approach taken for partial results in our framework is similar to that of introducing the internal lock for releasing results within a transaction, described in the previous section. In particular, we use a *Conditional-commit* lock, denoted by C_Lock, which in combination with the corresponding EDG can provides a safe mechanism for releasing partial results to a subtransaction of another transaction before commit. The lock for this purpose is termed *conditional-commit lock* to reflect the fact that it is to be applied to interim results of a long-running transaction, in the sense that these results are produced by parts of a transaction (subtransaction as defined by a local coordinator) and are to be released outside the transaction before the transaction as a whole commits.

When a transaction wants to release a data item before commit it uses C_Lock on it. In fact, it converts the item in question from X_Lock to C_lock as shown in Figure 3.10.The C_Lock-ed data item is available to subtransactions of another transaction, but anything that uses it must update the corresponding EDG. The information required for adding an entry to the EDG is the id of the transaction (IDT) and the id of the subtransaction (IDS). Notice that there is no need to keep information of the parent explicitly since we are now concerned with dependencies between transactions. The dependencies within a transaction, for which we would need the parent coordinator, is kept in the corresponding IDG as discussed in the previous section. The data item that is locked by C_Lock is released from a data-oriented coordinator of one transaction to a data-oriented coordinator of the other. In case of failure, anything else that used it has to be rolled back too. This information is given by the corresponding EDG (for moving across transactions) and then following the corresponding IDG of the transaction we arrived at (for tracking the internal dependencies within the transaction that used a partial result). It goes without saying that recovery is a major issue of concern in transactions but compensation and principled roll back routines have been explicitly identified as key requirements of SMEs (see R.4 on p.11).

Data items that have been locked by I_Lock can only be used internally, i.e. can only be accessed/modified by subtransactions of the same transaction, and this is recorded in the corresponding IDG. So these items can be considered atomic and can thus be rolled back (wherever necessary) based on the corresponding IDG. Recall that every subtransaction that has used/accessed such an item has added an entry to the IDG.

Data items that have been locked by C_Lock can be used externally, by other transactions, and this is recorded in the log structure given by the corresponding EDG. So subtransactions of other transactions that have used these items (locked by C_Lock) can be rolled back following the corresponding EDG. Recall that every transaction that has used a partial result from another transaction has done so whilst adding an entry to the EDG. Hence, in case of failure, for C_Lock-ed items we look at the EDG which takes across to other transactions that have used these items, and within that transaction we then follow the IDG to see what other part (subtransaction) of this transaction has made use of the inconsistent data items. This ensures that the whole spectrum of dependent subtransactions is identified by the recovery routine.

It transpires that the part of the transaction in which failure occurred as well as all related or dependent subtransactions of other transactions need to be marked as soon as possible and isolated so as to avoid costly failure propagation. In particular, the rollback for C_Lock items, due to the external dependencies incurred, can result in chains of rollback which is costly in terms of time, but also in terms of distributed accounting which will be discussed in Chapter 5 of this report.

In order to ensure that the damaged part of a long-running transaction is isolated as soon as possible we introduce a *Recovery* lock, denoted by R_Lock, whose immediate effect is to restrict data to the recovery routine only. The idea is that, in the first instance, all locks are converted to R_Lock until further notice.

We now turn our attention to optimising the recovery routine of data items locked internally using I_Lock, in an attempt to address the dynamicity of the environment more effectively. In particular, we

introduce a *Time-out* lock, denoted by T_Lock, which introduces further flexibility in our lock system for distributed transactions.

For optimising the recovery of (I_Lock-ed) data items results we use T_Lock on internally locked items so as to allow for a time-out before rolling them back. The T_Lock restricts access to data to the recovery routine only - just as R_Lock does. But the T_Lock also sets a time-out after which the I_Lock-ed data item will be rolled back automatically. The way T_Lock functions within the context of the overall locking scheme is depicted in the extended lock schema of Figure 3.10.

We have seen that I_Locked items are addressed by the recovery routine in a binary fashion; either rolled back or not, depending on the corresponding IDG. It turns out [RKM07] that in some cases rollback is not necessary whereas in other cases we are certain about failed subtransactions. The diea behind introducing the T_Lock is to give the recovery routine an opportunity to reconvert T_Lock to I_Lock if rollback is not deemed necessary. If no decision is possible within a certain amount of time, then the data item is rolled back automatically.

In this way, safe results, those not affected by the damaged part of the transaction, do not need to be rolled back or re-computed in case the transaction needs to be re-started. Clearly, there is a relation between T_Lock and the handling of omitted results and this should become more clear in the following discussion.

Let us assume that during a long-running transaction, the connection between the local coordinators is lost. The situation is depicted in Figure 3.11.

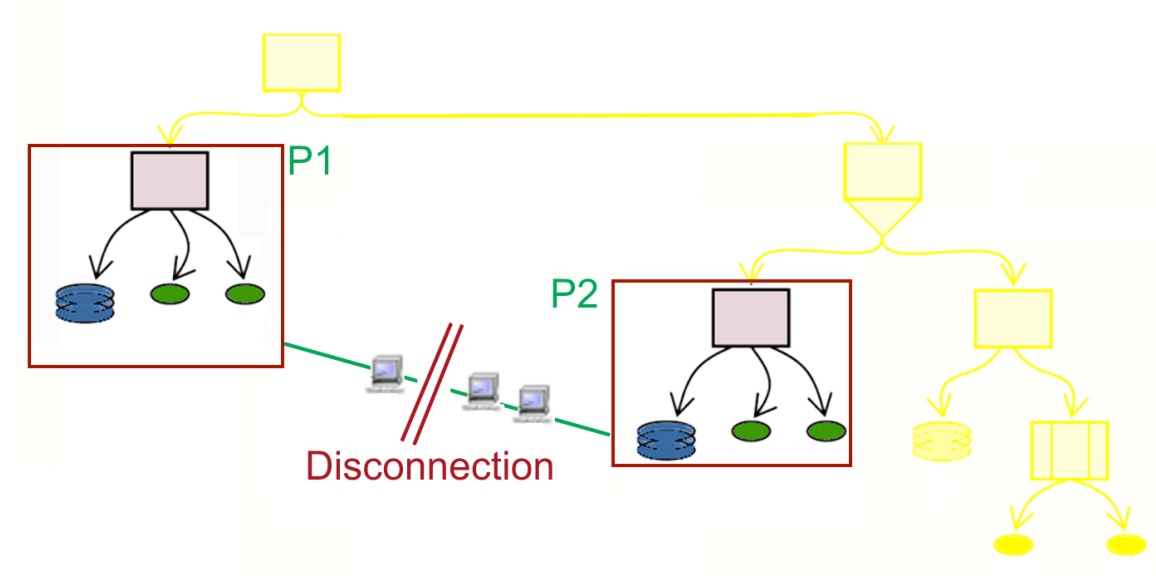


Figure 3.11 Failure due to loss of communication during a long-running transaction

Further assume that within the local platforms P1 and P2, some results need to be shared (between subtransactions). In our framework, this is possible by using I_Lock on such data items.

Now in the event of disconnection, say, the link between platform P1 and P2 is lost, the long-running transaction does not necessarily have to be aborted as a whole in our framework, as would be the case with conventional S/X lock models (and current transaction models). The objective in this case is to ensure that costly (in terms of time, network usage, execution of underlying services, etc.) results within each part of the long-running transactions are not lost. The use of the time-out lock T_Lock proves useful in this respect.

Results that have been used internally (within each platform or local coordinator) have been locked using the internal lock I_Lock. Once the communication between the local coordinators is lost, the recovery routine is applied to (the whole of the) transaction. This means that any data items locked by I_Lock are converted to T_Lock, which restricts access to I_Lock-ed items to the recovery routine only, and sets a time-out after which the items will be rolled back automatically. If within the time-out, the recovery routine determines that the items (or some of them) have not used inconsistent data or have not been affected as such by the failure in some other part of the transaction, then the items are converted back to I_Lock. This means that in case the transaction is re-started these data items do not have to be re-computed.

For instance, the connection between the two local coordinators may be established again, within the time-out, or there might be an alternative path through the supporting network for P1 to regain connection with P2. This is described in Figure 3.12 which shows that an alternative path through various other platforms is readily available and can be used to reconnect P1 and P2. Or, it might be the case that there is another platform, say P3, which is connected to P1 over the P2P network and also has connection to P2.It is important to note that this touches upon the very characteristics of the underlying P2P network such as replication and duplication, which are discussed in Chapter 4.

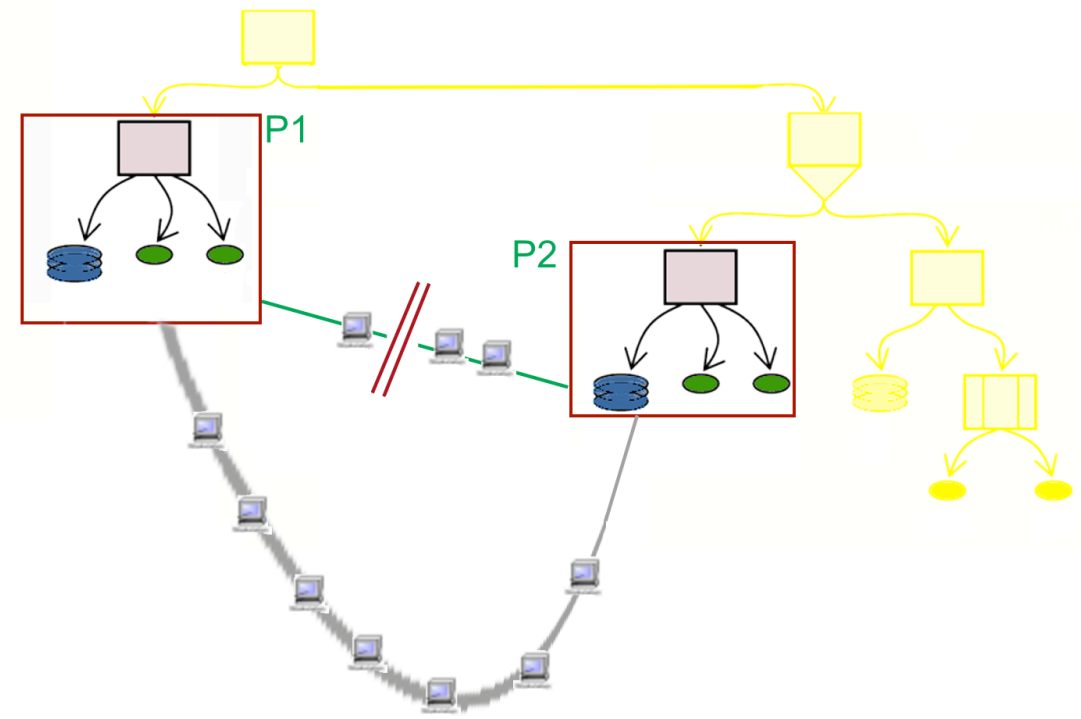


Figure 3.12 Reconnection between local coordinators in a long-running transaction

It can be seen that in case the communication between P1 and P2 is restored (there exists an alternative path and this can be identified) within the pre-set time-out of T_Lock, the long-running transaction does not need to be aborted. The results in each of P1 and P2 will not be lost, in fact hey will be converted back to I_Lock, since the whole transaction will not be aborted.

It is in this sense that the extended lock system for concurrency control and recovery management in our framework for distributed transactions over a P2P network provides additional flexibility and can be used to cover a wider range of B2B scenarios. Furthermore, the design of the transaction model has been based on the principle of preserving local autonomy and thus is well-suited for business transactions between SMEs.

3.6 Social science questions regarding the transaction support in the DE architecture

One of the major challenges with regard to this aspect of the core architecture design for the social scientists was simply formulating an understanding of the model as non-computer science researchers. It was clear from early discussion that that the need to support long-running transactions was a key aspect of the design rationale. The focus of discussion and critique in this area was why – in the context of the current environment for B2B interaction - this characteristic was of such significance?

Lengthy discussions on this point took place face-to-face and via email through which it gradually became clear that the requisite need for sufficient stability, reliability and recoverability to support long-running transactions has been used as a justification for maintaining centralised, proprietary B2B systems. It was previously the case that only the computing capacity and resources provided by large technology companies were sufficient to achieve a sufficiently robust network. However, the approach taken by computer scientists in OPAALS among with latest advancements in distributed web applications show that there are ways around this problem, which can remove the need for centralised command and control and its potential pitfalls.

Therefore, the distributed model for long-running transactions proposed here has potentially extremely important consequences for unlocking the current eB2B computing environment for SMEs. Although it cannot be presumed that a technology architecture can necessarily reconfigure human and business interactions, it appears, at a design level, that the transaction model holds significant potential to reduce the centralised and therefore potentially monopolistic nature of the current environment for B2B interaction. This raises an interesting point with respect to the approach to the socio-centric aspects of the development of the digital infrastructure for DEs. Here we have only touched upon the range of different approaches, from techno-determinism to socio-determinism, and this is a line of debate that will continue to occupy us in future work.

4 A Peer-to-Peer Network Design for DEs

In this chapter we turn our attention to the underlying network connecting the participating entities in a digital ecosystem. The main objective in the first instance is to support long-lived transactions. The basic requirements for the digital ecosystem network are that it must be distributed (a premise for lowering the barrier of adoption), resilient to failure (to perform transactions in a distributed manner), and have a dynamic topology that continuously evolves to reflect the changing needs of the interacting communities it supports. The *peer-to-peer* (P2P) network design we describe in this chapter exhibits such features and enhances the ability of the DE core architecture to address issues of power and control as well as avert monopoly phenomena and ultimately protect the democratic nature of DEs.

The work on the self-organising P2P network described in this chapter draws upon the main design features introduced in Deliverable D3.2 [RMK07b]. The overall design has been extended and fine tuned with experimental simulations, which have recently appeared in articles in International Conferences on *Digital Ecosystems and Technologies* [RMK08a] and *Computational P2P Networks* [RMK08b]. In what follows, we outline the basic ideas.

4.1 Towards a P2P network to support SME business transactions

The primary purpose of a business network is to enable networked organisations to engage in distributed business transactions that realise their core business activities. ITA's experience with SMEs in the Aragon region and elsewhere indicates that a digital infrastructure for SMEs stands a best chance to be sustainable if it can perform business transactions. We have seen that our approach to providing support for long-running transactions in digital ecosystems, outlined in Chapter 3, allows the sharing of uncommitted results within a transaction as well as the exchange of results across transactions before their final commitment (partial results). At the heart of the model are the distributed log structures, provided by the IDG and EDG graphs, which keep track of the dependencies that arise between the underlying service executions. We have also seen how these can be analysed formally to increase confidence in a successful outcome prior to deployment, as described in deliverable D3.2 [RMK07b] and [RMK08a], and outlined in Section 3.4 of this report.

The log structures are used to drive the recovery management which is optimised, in terms of preserving as much progress-to-date as possible (omitted results), and including provision for alternative scenarios or paths of execution (forward recovery). Further, in combination with the fine-grained lock mechanism introduced in [RMK07d], our overall framework allows for maximal concurrency during execution and supports the local coordination of a peer's services. In this way, any violation of local autonomy is avoided and the transaction can be executed in a fully distributed manner.

The execution of a distributed transaction results in the formation of a temporary network, as shown in Figure 4.1, 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.

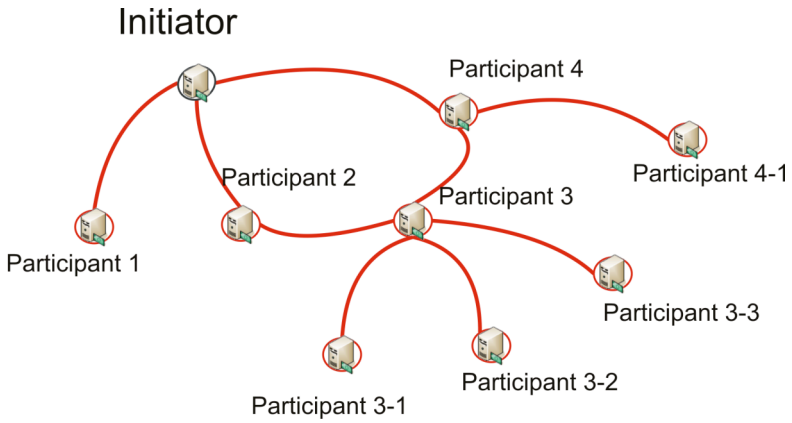


Figure 4.1 Temporary virtual network created by executing a long-running transaction

Admittedly, transactional aspects such as durability and reusability can pose further challenges. Durability refers to the effect of the results of the computations involved in a transaction while reusability has to do with recycling parts of previous (even unsuccessful) transactions in future transactions or in re-starting a transaction that was aborted. Stability of the transaction, especially given the dynamic nature of SMEs in this context, is a further concern.

The compensation routines that are necessary in case some failure causes the transaction to be aborted come with an additional computational cost and long chains of roll backs can incur significant delays in the corresponding business activities. It is thus important to avoid the abortion of the whole transaction even when some (or even all) participants are temporarily disconnected. This problem has been adequately addressed when one of the participants, at each nested part of the transaction, is disconnected (see Deliverable D3.2 and [RMK07a]). Here, we describe a more generic solution in terms of a P2P network design that provides a highly reliable environment which can cope with the versatile nature of SMEs and the high probability of disconnection.

In addition, it is also important to keep the result of a successful or even unsuccessful transaction (even by considering regular unavailability of the Initiator and other participants). Another issue has to do with the lookup algorithm which would require even more knowledge of unavailability of SMEs or their services, based on their regular behaviour. This can be argued in terms of probability of fragmentation on such a network [RMK07a]. The need for a distributed P2P network is at the centre of the OPAALS computing agenda from the start of the project, but is also reinforced by the requirements analysis of the regional SMEs who have experienced that peer-to-peer solutions are the only to provide a fair and open environment for business collaboration (see SMEs requirement R 11). Our goal here is not only to provide a network structure for addressing these requirements but also to reuse fragmented network structures formed by the transactions in creating a fully connected network that is highly resilient to certain types of failure.

4.2 From business activities to Virtual Private Transaction Networks

The network of SMEs in a particular business domain is a collection of weakly connected temporary networks such as that of Figure 4.1 with occasional overlaps between different transactions, something that is also transparent. These temporary networks, the so-called *Virtual Private Transaction Networks* (VPTNs), have a coherent (domain-specific) structure which can be exploited in the design of the overall P2P network. For this reason, information about these smaller networks is kept locally at each peer involved in the long-running transaction, as described in Deliverable D3.2 [RMK07b].

Figure 4.2 shows a collection of such temporary networks resulting from the execution of business transactions. It can be seen in Figure 4.2 (which shows the result of 5 long-running transactions from the online travel domain) that Participants 3 and 4 are the overlaps, effectively playing the role of conceptual hubs in a typical P2P network.

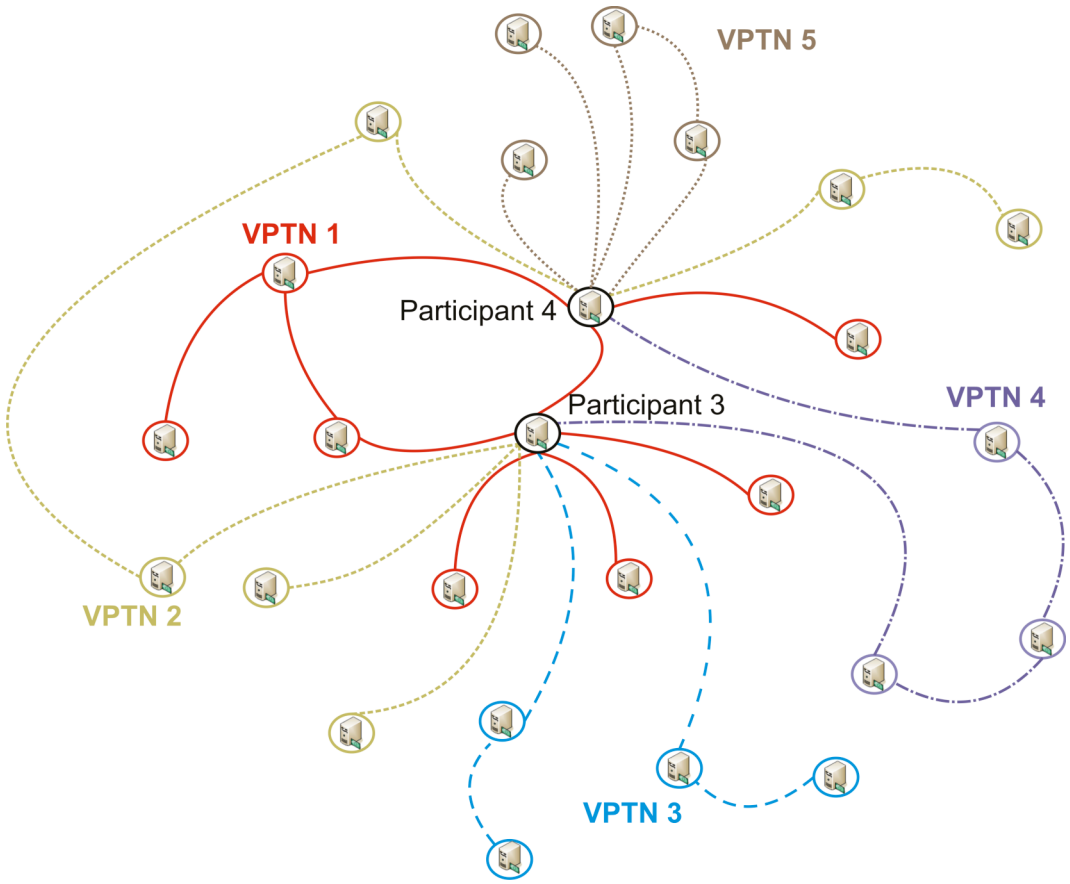


Figure 4.2 Virtual Private Transaction Networks (VPTNs)

One of the immediate benefits of considering the VPTNs is reliability and ability for prediction of the results on alternative scenarios. Based on the availability of platforms (services), a transaction can have more reliable execution and even during a temporary disconnection of a platform we can virtually keep the connection alive and give a timeout for return of the platform. The actual mechanism at the transactional level is implemented by the time-out lock (T-Lock) scheme described earlier in Section 3.5 of this report.

The current infrastructure avoids losing valuable information to some extent and provides practical methods for forward recovery that in the event of a failure would allow the transaction to continue following a different path of execution. Even the result of an unsuccessful transaction can be saved and reused in restarting the transaction. In this way, the underlying P2P network and the transaction coordination support combine to provide an environment in which long-running transactions can be run optimally and failures can, in certain cases as we will see, be detected and corrected without having to abort the whole transaction. This design is geared towards the need for managing resources of the digital infrastructure to ensure a most effective functioning with respect to business interactions expressed by SMEs in the regions (see requirement R10 on self-* properties and in particular, optimal functioning with regard to defined requirements (self-optimisation) and ability for discovering and correcting faults (self-healing)).

Such issues have informed the component-based design of the Local Agent outlined in Section 3.3. Having information about web services of different Participants and some information about their availability (which is kept in the 'Global Service Repository' of each peer, as introduced in Deliverable D3.2 [RMK07b] and subsequently described in our article on long-running transactions in last year international conference on *Digital Ecosystems and Technologies* (IEEE-DEST 2008), see [MRZ+08]), can be helpful for starting the same transaction or even a new transaction which shares some parts of the unsuccessful transaction. Meanwhile it is possible to analyse the reasons of failure and keep track of the operation of the transaction.

As each participant keeps the information about the others in its 'Global Service Repository', in this way the VPTN of a particular transaction is stable (and safe) during the transaction life time even if some participants are temporarily disconnected from the network. An exceptional situation may occur when all participants (including the Initiator) are disconnected at the same time. Unfortunately in these circumstances the reliability of the Global Service Repository can be questioned (as currently any update on a VPTN can be instigated by the Participants of the transaction). Existing approaches to addressing this problem include solutions such as the introduction of a powerful central node or a number of super peers [BG-M03], [LiM04].which have been discussed in detail in deliverable D3.2 [RMK07b] and [RMK08a], [RMK08b] and their analysis there shows them not to be appropriate in the digital ecosystem paradigm. In the following sections we describe the key design aspects of our approach and show how we can recover from a potentially fragmented network to a de-fragmented network using permanent clusters, which can guarantee the reliability of the stored information.

4.3 P2P network of connected VTPNs: further challenges

In this section we outline the key aspects that have gone into our P2P design that provide a fully connected network for the DE core architecture. The basic idea behind the design is to use the *local interactions* that take place between participants in long-running transaction as the main building block for the overall P2P network. These local interactions between peers in a similar domain come with the characteristics of interactions within a cluster and since they are part of a transaction the also come with useful information that can be exploited in providing a DE architecture that is purely distributed, dynamic, self-organising and highly resilient to failure.

In order to (re)use the local interactions at the VPTN level in designing the overall P2P network we have to connect VPTNs together. One of the first considerations in doing this concerns the choice of the best candidate from each VPTN. Careful consideration of all the necessary characteristics of the core DE architecture studied in Deliverable D3.1 [RMK07a] and D3.2 [RMK07b] and D4.2 [D4.2], which have also been outlined and reinforced form a socio-economic viewpoint in Chapter 2 of this report, shows that stability is of major concern. Thus, the best candidates for connecting VPTNs together are the most stable nodes from each VPTN. This is done by connecting the Global Service Repository of each candidate node from each VPTN. In the following section we define what we mean by 'most stable node' and provide a measurement of stability for the network.

4.3.1 Stability of the network

We have seen that fragmentation, the situation where some failure, or a series of failures resulting from a smart attack, divides the network into smaller networks that are no longer connected to each other, is a major threat for P2P networks as it is difficult to recover from. Even more so in distributed P2P networks

which are unstructured and have no central point of command and control. We have also seen in Section 2.1.5 the need for self-protection of the infrastructure, i.e. the provision for identification and protection against attacks (see SMEs requirement R10 in Section 2.1.5).

Fragmentation has to do with losing links between nodes and is tightly related to the degree of connectivity in the network topology. In other words, it relates to the average number of links nodes in the network have to other nodes. Taking this into account, the measurement of stability for a node is based on its contribution to the overall network connectivity. This can be expressed in terms of the online availability of each node. However, it would be unreasonable (and not feasible) to expect nodes to be connected (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, we introduce the so-called *Expected Availability Time* (EAT). This is the time the node is expected to be available and online in the network. Figure 4.3 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 experience disconnections. This 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 $NodeStability \leq 1$ and the closer the NodeStability function is to 1 for a node, the more stable the node is (which can be understood as more reliable or predictable). It is important to note that the measurement of stability is dynamic in the sense that it is not a one off measurement but it is continuously being calculated and changes accordingly over time. Factors that can affect node stability, as given by the above simple function, have to do with the DisconnectionPeriods which are the result of various types of failures at both the network and the service levels, e.g. bandwidth, traffic complexity and bottlenecks, or processing power, parallelism capabilities of the node, service unavailable, to name a few.

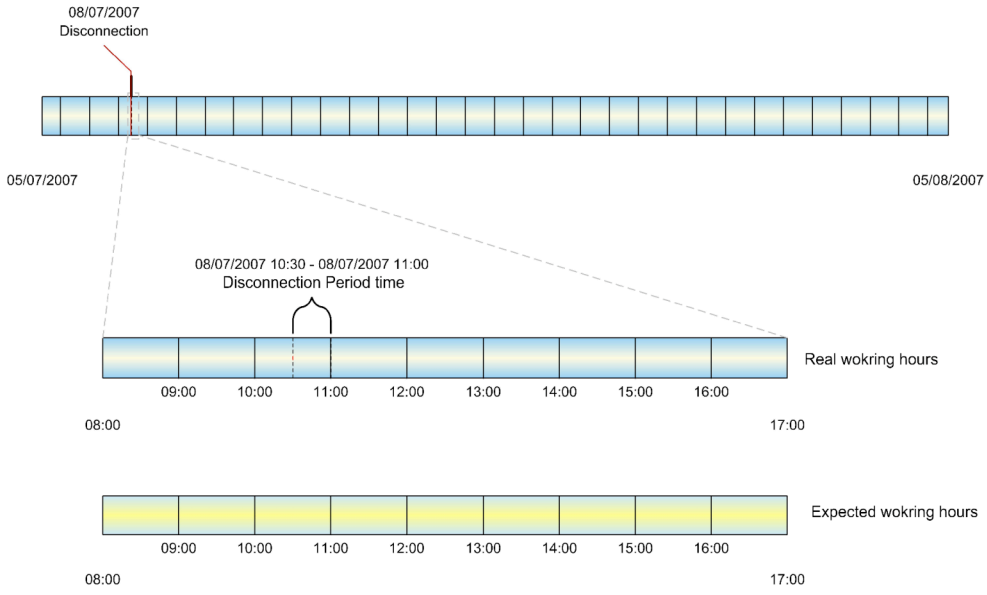


Figure 4.3 A measurement of node stability in terms of online availability

For example, in our experiments we have considered a traffic limitation on the topology as an additional parameter feeding into the DiscnnectionPeriods of the node stability measurement. If the infrastructural traffic reaches a specific percentage of a platform's bandwidth, then node experiences serious difficulties

(delays, loss of data packages, etc.) and it seems reasonable to regard it as being effectively disconnected. This will increase its DisconnectionPeriods and have a negative impact on its stability measurement. Regarding the regional practice of this work, in our experimental simulations the bandwidth of a node has been treated as a random number between 500kb to 4mb and the maximum percentage of the infrastructural traffic is 30% of the platform bandwidth. It should be noted that this can vary depending on the environment and the average transactional traffic of the corresponding VPTNs at a given time.

The EAT of each node is at the moment considered constant, or to be more precise is given and available before calculating the DisconnectionPeriods. The reson for this is that, at the moment, we have considered EAT to be part of or result of an SME's business model and consequently it is reasonable to expect that it is set by each SME on joining the network. In other words, the EAT parameter is fixed or can only change on the account of the SME providing it.

As far as calculating the stability function of a node is concerned, in the first instance we use its participants in a transaction (other nodes in the same VPTN) in the first instance. This is a most reliable source that can readily provide such information since the participating nodes already check the availability behaviour of each other. Put simply, when a node is disconnected the most likely nodes to be aware of this first are the ones which it does transactions with. In fact, the DisconnectionPeriods could also be calculated by the neighbouring nodes in the layer above the VPTNs, that of the Service Network Layer (VSN) which is not discussed as such here but has been described in Deliverable D3.2 of OPAALS. This is currently under consideration as a more objective measurement by means of adding redundancy in the process of calculating node stability.

By using the measurement of stability introduced above we can already 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. In short, this is because we still rely on so-called *permanent nodes*, i.e. nodes that will always be connected and available online. This means that the network is still dependent on each platform's availability and if the total online time of all stable (or permanent) nodes cannot consistently cover 24 hours the network will collapse for some period of time, precisely that in which all nodes are not available.

The issue of enhancing connectivity by considering the most stable nodes as permanent nodes in the network has attracted interest among researchers in the networking communities. Models which provide self-management capabilities at the service level [MBC05], [Kri+], [Sah+06], [LiM04] and *Quality of Service* (QoS) at the virtualisation levels [DaS06] can be seen to be another extreme solution for the Digital Ecosystem environment. But the network resistance against failure in the collaborative business activities and the corresponding long-lived transactions have not been considered. This is an important feature in the core DE architecture and not trivial to address. None the least because of the very nature of SMEs and their dynamic and volatile business models, which cannot be expected to provide the necessary permanent platforms for an overall connected network.

An approach that forms a dynamic and self-organising P2P network was demonstrated in FADA [FADA]. The hypothesis of this work is that in order to achieve a resilient and adaptive peer-to-peer network, each network node must proactively maintain a minimum number of edges. Apart of different method for achieving the solution, as the nature of nodes are different, the temporary availability of nodes do not play the crucial role and the same solution may not apply to the DE network as there may not be any permanent node in the network. As indicated by ITA's fieldwork with regional SMEs the explicit requirement for a scalable and reliable infrastructure (see requirements R5, R11, R10 in Section 2.1.5) does not come with promises that SMEs are willing (or in a position) to provide powerful permanent nodes. The approach taken in OPAALS to address these aspects adequately is centred around the construct of *permanent clusters* of nodes, instead of individual nodes, and it should be noted that these aggregations of permanent, or most stable, nodes are formed dynamically in our design. This results in the so-called *Dynamic Virtual Super Peers* (DVSPs) which are discussed next.

4.4 Dynamic Virtual Super Peers

In order to provide the most effective architecture that covers the characteristic requirements mentioned throughout this report, we have tried in our network design in OPAALS to move towards a more dynamic architecture that reflects the dynamicity of a digital ecosystem. In terms of the network topology, key to achieving this is to avoid conventional super peers, which are pre-selected statically and perform a specific role that does not change, while retaining the good aspects of this solution, namely the stability that comes with super peers.

As a first step towards moving to a more dynamic architecture which does not rely on just a few permanent nodes, we try to find permanent clusters in the network. More specifically, we identify 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 other words, we are trying to find permanent clusters through the most stable nodes from the VPTNs. In this way, the local interactions at the VPTN level can be diffused to the global level across the whole network.

It is important in determining permanent clusters that we identify aggregations of nodes from different time zones that can cover 24 hour availability. Any union of the stable nodes in the aggregations (which provides 24 hour availability coverage) are actual permanent clusters. These aggregations of most stable nodes from the corresponding VPTNs are the so-called *Dynamic Virtual Super Peers* (DVSPs), or VSPs for short. These aggregations of stable nodes are formed or, to be more precise, emerge dynamically since the stability measurement effectively determines the best candidates for joining a VSP. And since node stability is continuously being calculated the members of a given VSP also change over time. We will have more to say about this aspect of dynamic VSPs in the sequel.

Figure 4.4 shows the 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. In this case, two stable nodes from one time-zone have been included and three stable nodes from another. The green and creamy signs are used to denote the different time zones.

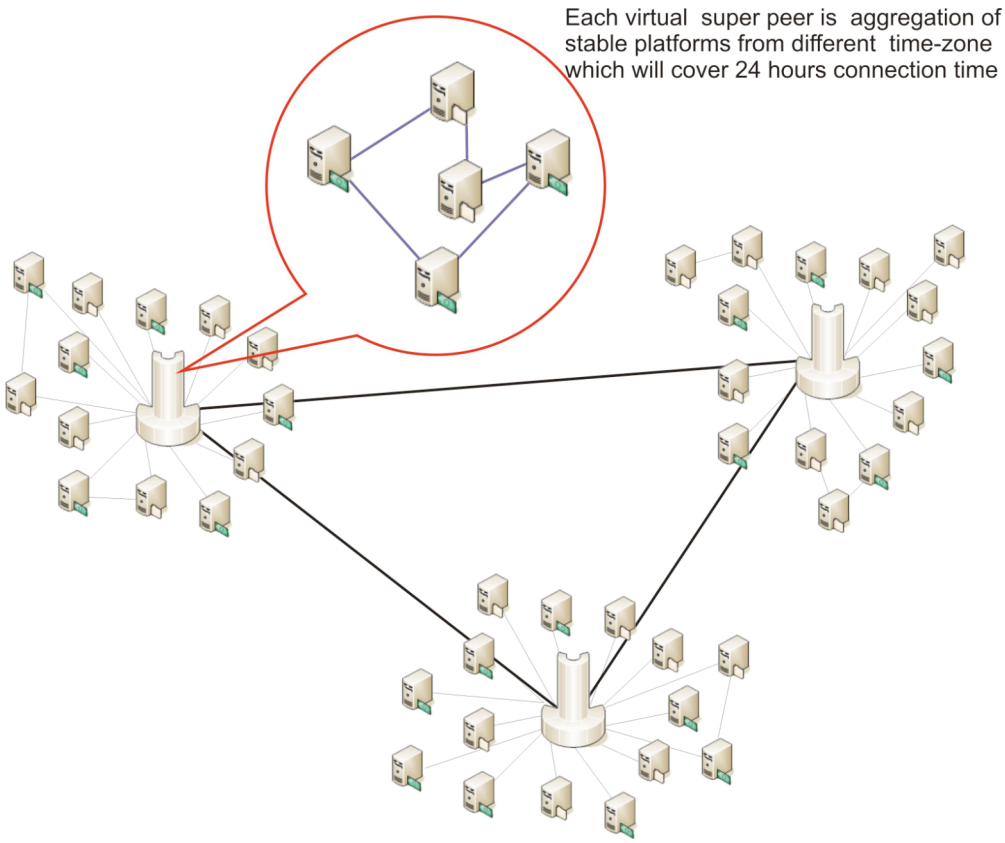


Figure 4.4 Permanent clusters and dynamic VSPs in the DE architecture

By considering stable nodes from permanent clusters, as shown in Figure 4.4, we can create 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 one hand and the connection between them and their nodes decrease 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. In this manner, the good connectivity can cause more reliable transactions at the VPTNs level.

Figure 4.5 shows a collection of small and medium enterprises which exhibit different behaviours. Terminal machines or personal computers represent small (and very small) businesses (VSBs) which could be reasonably expected to not be most stable. Their actual availability is typically very limited, i.e. their regulation for being part of the network will not follow their availability pattern. The server machine shapes are used to represent SMEs. Based on their business nature, these can be reasonably expected to be more stable – usually available during a given time period (EAT). The assumption here is that their availability pattern is more stable than that of terminals and over longer periods of time (larger EAT). In short, server-type nodes are more available than terminal-type nodes. In addition, there is a further distinction in that servers with a green sign have good stability, and are from the same time zone, while servers with a creamy sign have good stability measurement but they are in different time zones. The two time-zones together in this case can cover 24 hours.

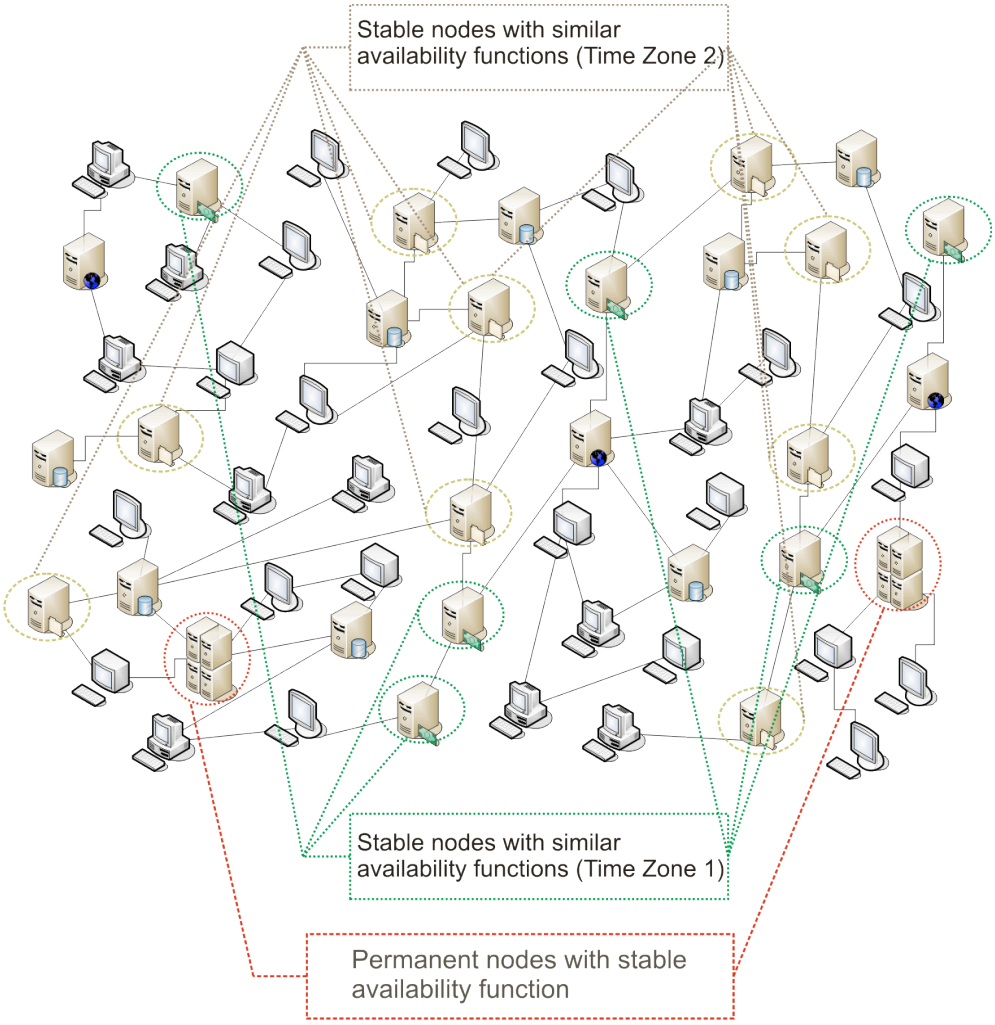


Figure 4.5 Using regional stable nodes to provide strong connectivity

In effect, the VSPs based design introduces a *virtual* layer in the network. as depicted in Figure 4.6, but does not impose any hierarchy or rigid structure. The nodes that are part of a VSP are normal nodes of the digital ecosystem that have the capacity (at certain points in time) to perform more network management operations than others.

The introduction of VSPs allows us to get a handle on a large-scale unstructured network and affect certain characteristics of the network with minimal intervention. For instance, the traffic is spread over the virtual super peers and there is less risk of bottleneck at peak time. At the same time, nodes within a virtual super peer need to keep information only about nodes in their cluster and about neighbouring VSPs, but not information of about the whole network as is the case with classic super peers. Such information needs to be updated regularly and synchronised across super peers and this at off-peak times when the network configuration does not change a lot can generate unnecessary data processing. Using VSPs the amount of redundant information processing at off-peak times is reduced dramatically as compared to the classical super peers solution.

It can be seen that the design of the P2P network around the dynamic virtual super peers not only covers issues of scalability and performance, which are explicit requirements from the regions of SMEs (see for example R5, R11 in Section 2.1.5) but does so a fairly optimal manner. By this we mean that the monitoring of resources is done in way that improves traffic complexity (does not induce unnecessary large data overheads) and the re-oraganisation of the network itself to reflect its usage over time is done in an open

and all-inclusive manner (self-organising) without requiring a central network mediator that choreographs the re-organisation process. Note that self-management and self-optimising properties were also identified as key requirements for the DE infrastructure from the regional SMEs' viewpoint (see requirement R10 in Section 2.1.5).

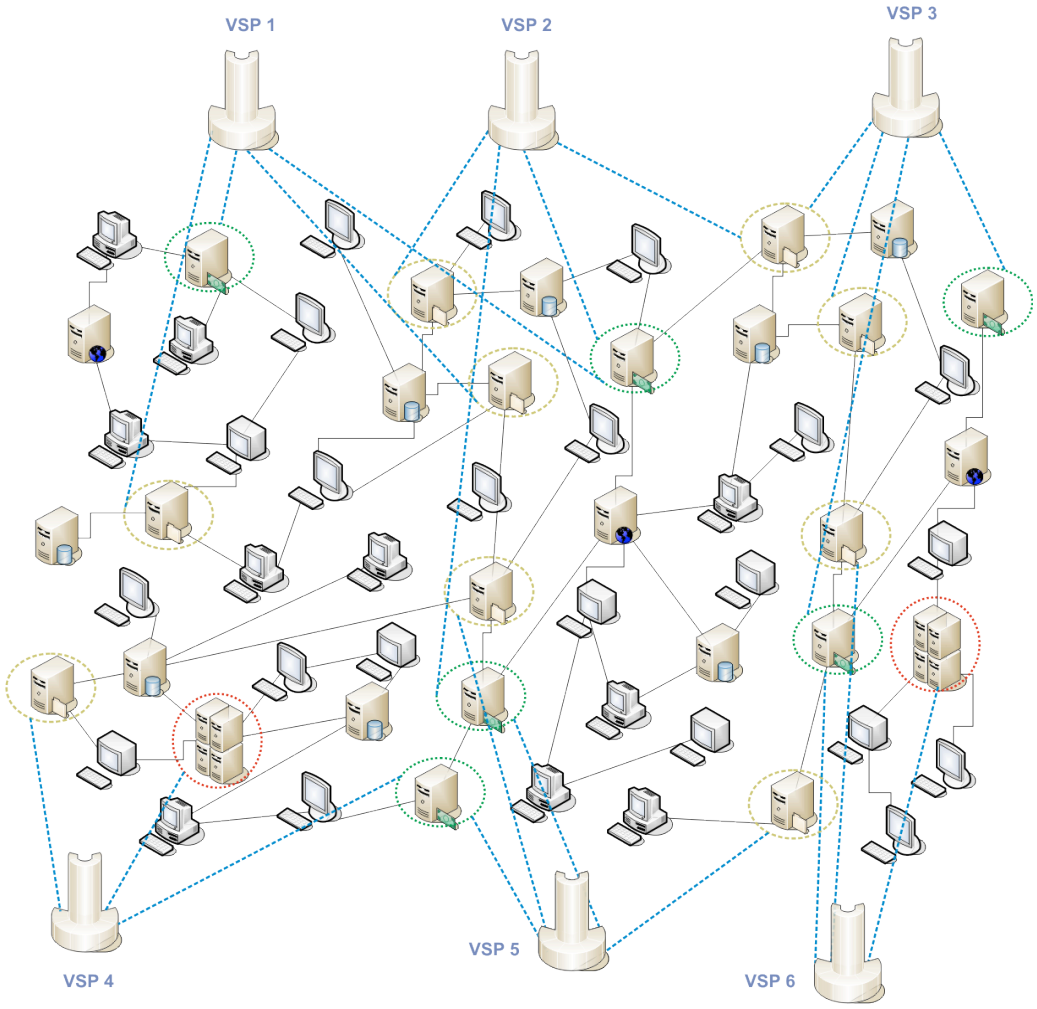


Figure 4.6 Using Dynamic VSPs for optimising the digital ecosystem network

Since choosing stable nodes is a dynamic process (it is done based on the stability function, EAT over DisconnectionPeriod of a node during EAT, whose value varies over time) the VSPs 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 VSP, when its node stability increases and overcomes some threshold, and 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 lose their virtual peer status. Within a digital ecosystem for business, SMEs would be expected to invest at that time (in hardware, processing power, bandwidth etc.) and become again part of a virtual super peer in future. It is in this sense that the topology of the P2P network evolves and adapts itself to reflect the usage and demands of the participants who at the same time benefit from and contribute to the 'sustainability' of the network.

Additionally, network congestion can change the maximum level of node stability 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 stability of each virtual super peer.

Furthermore, we expect a reasonable cluster coefficient on the account of having VPTN as the main building block which we have seen is 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 [YPH02]) and explorative service composition (see service-oriented computing roadmap in [PTD+06]) 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*. We will have more to say on this in the concluding section of this report.

4.4.1 A Stable Digital Ecosystem Network

The first implementation of a Digital Business Ecosystem done in the DBE project [DBE] was relying on FADA nodes [FADA] as the core infrastructure for the network. The diea behind this implemnetaion was that items (service proxies) are registered in any node of the FADA cloud, and they can be searched for starting from any node in the FADA cloud. The FADA nodes create a free graph, i.e., the topology is not enforced and not even known.

FADA as a conventional scale-free network [BaA99] is relying on a few hubs. The effect of this topology for SMEs is an inevitable risk of bottleneck at peak time. As a simple example Figure 4.7 shows how the core infrastructure may indeed rely on a few hubs. However, this not only causes high traffic at peak time (and as a result threatens the stability of the hubs even during this time), but also the possibility of fragmentation and creating ‘islands’ as the network grows. This should not be overlooked, especially when we take into account the regular unavailability of SMEs based on their business model and regional working hours.

The picture of the network shown in Figure 4.7 differs slightly from the typical picture of a scale-free network because our experimental simulations are targeted to a business network resulting from the corresponding VTPNs formed while transactions take place. It can be seen that there are a few hubs or peers with a large number of links (more than one would find in a typical scale-free network but they are still present) – these are the more dense parts of the picture – while there are other nodes / peers who have very few connections.

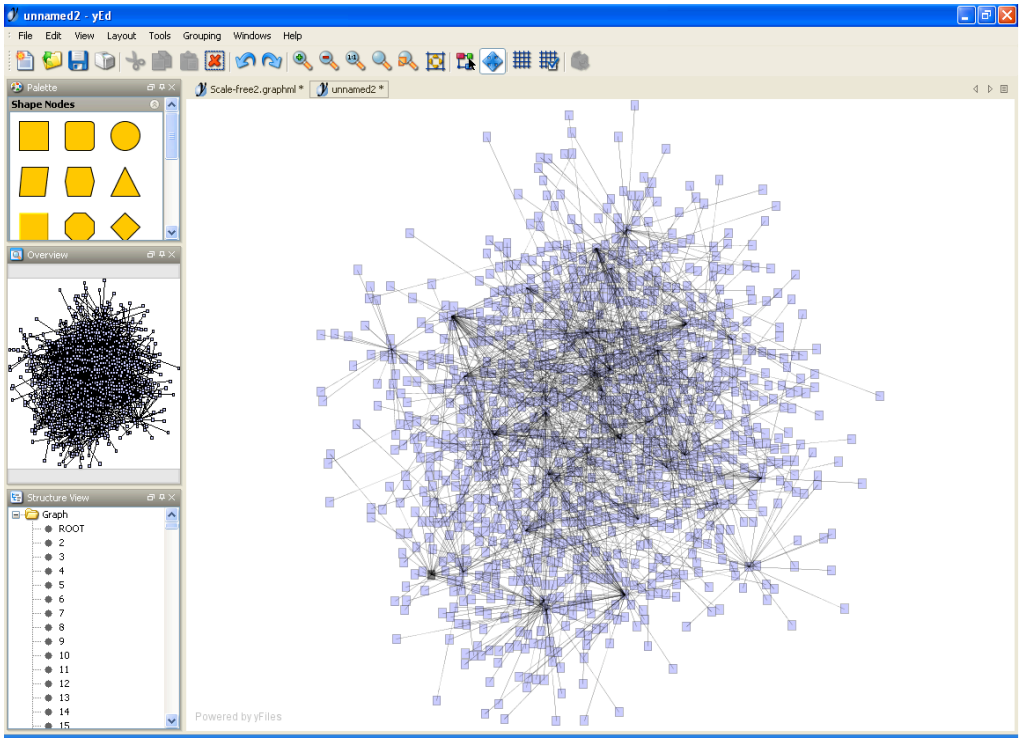


Figure 4.7 A scale-free network as a result of business transactions (via VPTNs)

This can perhaps be seen more clearly in Figure 4.8 which shows the relationships between the number of nodes and the number of links. It can be seen that a few number of nodes have the most number of links (high distribution degree) while the majority of nodes have just a few links (few links which mostly end up (are linked) to a high degree node). In our experimental simulation it turns out 8 nodes have more than 2500 links while more than 4000 nodes have less than 5 links. This is reflected in Figure 4.8 where the graphical representation of the distribution degree shows it to be rather close to the *x* and *y* axes (of number of nodes and number of links).

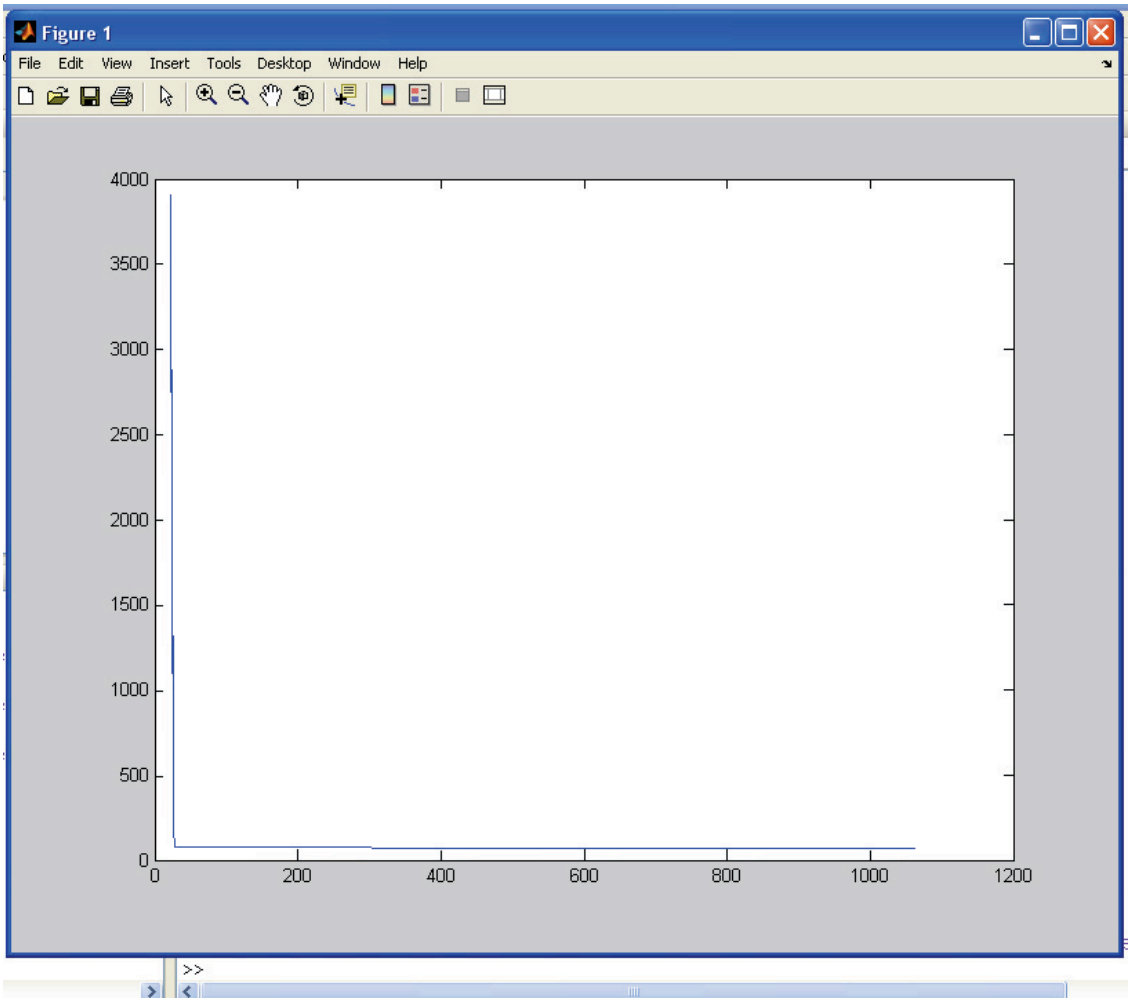


Figure 4.8 Uneven distribution of links in a typical scale-free network

This is typical of a scale-free network whose distribution degree follows a power law distribution. The problem with this network design is that any failure or smart attack on the hubs can cause immediate disruption at the transactional level (abortion of the transaction) and fragmentation of the network. These problems are addressed in our current design and the use of VSPs shows significant improvement on the infrastructure of the digital ecosystem as the dynamic topology of the network can react in response to failures or attacks on the virtual hubs.

By using a dynamic measurement for choosing nodes in VSPs, the dependency on a few nodes with higher distribution degree decreases dramatically. Figure 4.9 shows an example of a DE where links are propagated on different nodes.

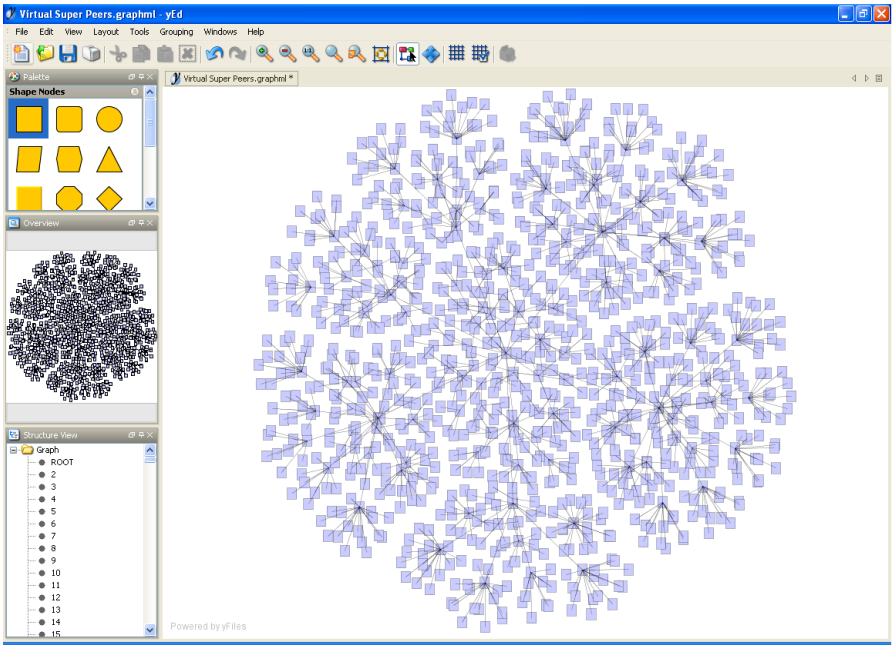


Figure 4.9 Reorganisation of the DE network using VSPs

Our primary results with the same number of nodes (5000) shows a dramatic shift in comparison with the typical FADA infrastructure (see Figure 4.10). In our experimental simulation it turns out that more than 80 nodes belonging to some VSP have more than 350 links. This may not be surprising but even their neighbouring nodes (the ones which have node stability close to 1) have a large number of links (about 900 nodes with more than 300 links). This means these nodes already become good candidates for joining the virtual super peers by substituting existing member nodes during failures or attacks on current VSPs.

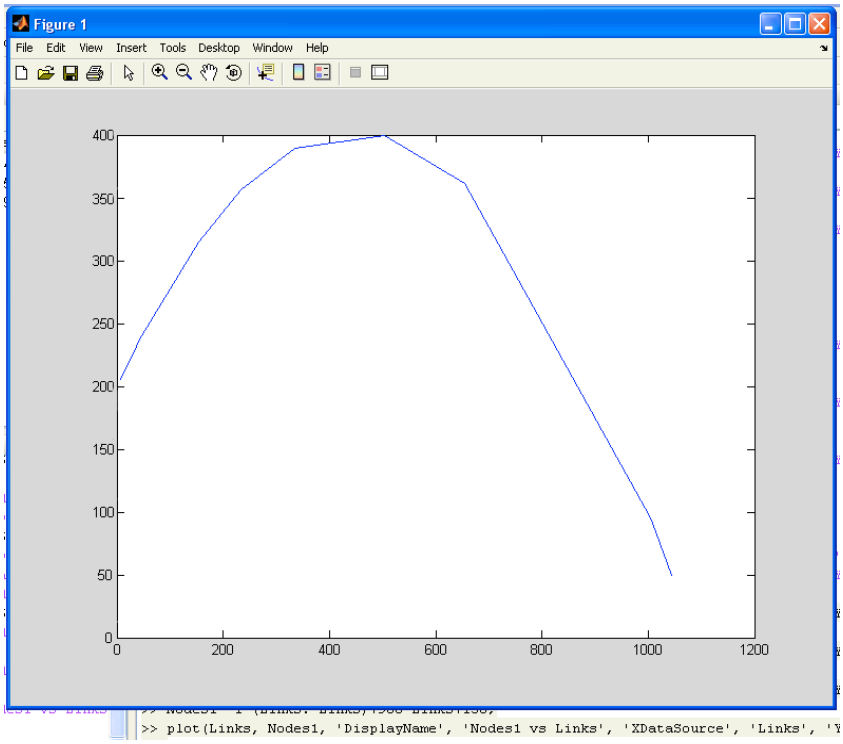


Figure 4.10 Fairer distribution degree in the DE network using DVSPs

Figure 4.11 shows the result of a simulation of 800 simultaneous attacks to VSPs nodes in DE network of 5000 nodes. This means that 800 nodes in the aggregations forming permanent clusters are attacked. The effect of the orchestrated smart attacks is that these nodes (in VSPs) lose all of their links. They can only rejoin the network through performing a few business transactions. As a result of the attack, they have very week stability while doing so (NodeStability closer to 0). Therefore, despite of their transactions they will still have a low number of links (less than 75 in the simulation of Fig. 4.11). Meanwhile, their neighbouring nodes have substituted them in the corresponding VSPs structure and their links have increased. To be more precise, the more stable of their neighbouring nodes will have now become part of the corresponding VSP in their place.

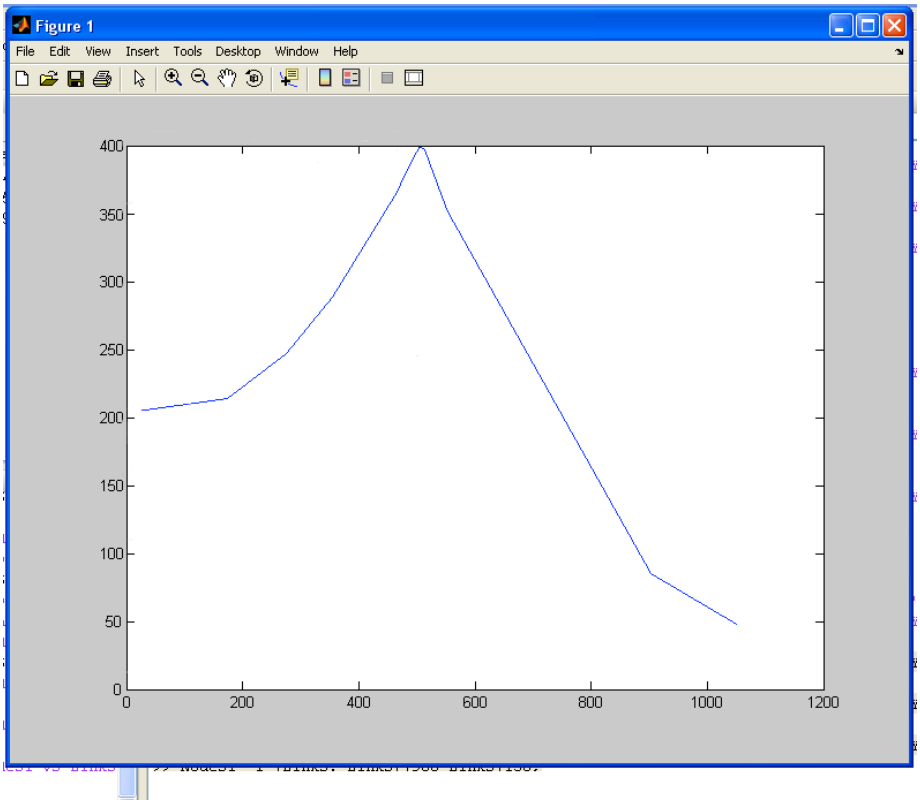


Figure 4.11 Reaction of the DE network using VSPs to smart attacks

So the overall effect of the attack on nodes in the VSPs is that they lose that status, they are substituted by the most stable neighbouring nodes in the corresponding VSP, but even this effect is temporary as they can again built up their node stability and become a good candidate for bubbling up into a VSP again. The interesting point however, is that still the network has not suffered any fragmentation. Some longer response time on business transactions involving these nodes may be experienced, but the DE network does not suffer the full failure. In the general case, this type of attack could cause several fragmentations and the transaction latency could be indefinite [RMK08a].

4.5 Social science questions regarding the P2P network in the DE architecture

One of the most closely argued areas of the core architecture design took place in relation to the Peer-to-Peer network design. In particular, social scientists sought qualification of the concept of a 'Virtual Super Peer'. Furthermore, they sought to understand the operational consequences of having virtual super peers for digital ecosystems users. This involved imagining scenarios of use and mis-use and asking the computer scientists to qualify and explain how technical procedures would be carried out. The main concern of the social scientists was that equity was maintained across the network. From this point of view, it was important to understand whether or not holding the position of super peer offered any operational advantage to an individual or organisation. In order to prevent domination or potential monopolisation of the network it was important for the social scientists to establish whether there was any potential for exploiting the virtual super peer system, either through some form of hacking, or through organisations with large computing capacities monopolising the super peer position.

In addition it was important to establish that all users regardless of size would be able to take advantage of the Peer-to Peer network. One important question in this respect concerned the network's use of 'spare computing capacity'. There was concern from social science researchers that this would produce a barrier to participation for small companies who would be unable or unwilling to contribute spare capacity. Discussion around this centred on ideas of reciprocity and the need to cultivate the idea that in gaining use of the digital ecosystem it was necessary to contribute computing power, but that only spare computing capacity would ever be used.

Discussion of the virtual super peer system resulted in social science researchers requesting that the architecture designers change the name from virtual super peer to dynamic virtual super peer. It was felt that the term super peer was suggestive of superiority and a possibly centralised network design. Dynamic virtual super peer was felt to be more representative of the flexible, constantly shifting network topology.

Some of the key questions asked by social science researchers on this topic are listed below, along with the answers that were provided by the computer scientists.

1. What are the "simple (but dynamic) measurements" that elect the best peer?

The basic factor that goes into determining the best candidate peer from the local pool of peer involved in transactions (those forming the VPTN temporary network) is reliability. In other words we need stable nodes that operate predictably in the transactional setting (at the local level of VPTNs) but also provide good connectivity for the overall P2P network (at the global level). This is because connectivity is a major issue of concern in large scale unstructured networks – networks which have a large, potentially ever increasing number of nodes and whose topology is not fixed by a central mediator node / operator. In addition, good connectivity is the best means to avoid fragmentation, the foremost threat on distributed P2P systems. As was pointed out repeatedly by the social scientists, we cannot expect every small company to be online, up and running 24/7, and therefore we have based the measurement of stability on the expected availability time. This is declared by each participant and its measurement of stability then depends on the length of the disconnection periods it experiences (for whatever reason) during this time. It can be seen, and this is an important point, that the stability measurement is not an one-off calculation but is continuously calculated over time. This means that it changes as the behaviour of the participating platforms varies.

Further, as a node moves up the layers of DVSPs it will have to commit more and more resources for network management operations, which means that it is more likely that it will experience disconnections or periods where it is unavailable for its own transactions. The effect of this is that it will no longer be a member of the DVSP, or to be more precise, it will drop a level down, and will be substituted by the best

candidate node(s) from the level below. In this way, the best candidates keep being 're-cycled' and the network topology continuously shifts to reflect these changes.

2. Are they dynamic only because we are assuming that a peer will never maintain the VSP position? Or does a VSP that continues to be the best maintain this position only for a certain period of time?

If a node continues to be the best candidate it will maintain its DVSP position. However, it will not necessarily remain in the same DVSP group. Depending on the number of nodes, and we have seen that this design around DVSPs works best as the number of participants grows, is will be moving up to DVSPs of the next level. On the one hand there is an incentive to maintain best candidate position and move upwards, on the other hand this moving upwards comes at the cost of committing more and more computing capacity to operations other than the business the company does. There is therefore a striking balance to achieved, and determined by each node itself, in the desire to keep moving to higher level DVSPs.

3. The core architecture designers have said that "peers will never know they are VSPs". Is it really impossible? Such knowledge could allow the network organisation to be 'hacked' and the information exploited.

A potential empirical example proposed was as follows: imagine that one SME is somehow well connected to - or part of - a bigger network - as a research centre spin off could be. This could mean it has very high computational & connectivity power and uses more resources than one could expect from an SME.

This might be theoretically possible but in practice extremely difficult. In particular, the process by which a participating node can find out the location of all other nodes in the network (location here understood as position in a DVSP, and in which in that) is not scalable. This is because there is no central entity dedicated to performing monitoring operations, and thus it would require of the node to monitor (gather the relevant information, for which it might well need other nodes' cooperation, since it will not be in direct communication with all other nodes on the network). This means monitoring, mostly through others, all message packages across the whole network, and then analysing this huge traffic data which is in itself problematic as it would require extreme processing power and take considerable amount of time. In addition, a DVSP is not a single node, it is an aggregation of stable nodes, and therefore it is not difficult for the monitoring activity of one peer to be picked up by other peers in the same DVSP.

However, and since this theoretically possible as mentioned before, assume that a node does succeed in getting an understanding of the location of other nodes. We have seen in this chapter that the network topology is not static, but it is changing continuously (self-organised network) so this will not pay off for the node as expected. What the node would know at say time t1 about the location of other nodes (assuming it could achieve what was just described) would not necessarily be valid at time t2, t3 and so on. In fact, this is one of the strong (and often unseen) advantages of having a distributed network that is self-organising.

4. The core architecture designers have said that "VSPs will not have any competitive advantage". Is it really impossible and why?

A node that is a member in some DVSP cannot take advantage neither in terms of current business activities nor in terms of causing disruption to the network, e.g. affecting network performance or connectivity. The very fact a DVSP is a group of permanent or most stable nodes, in combination with the fact that there is not a single DVSP but groups, or better, layers, of DVSPs across the network provides in-built redundancy in attempting to take advantage and monopolise the network. The benefit of belonging to a DVSP is that the node is aware of many more nodes and can interact (or perform transactions) with them directly. However, the DVSPs are interconnected VPTNs and thus not necessarily in the same domain. So there is potential competitive advantage but that comes in the from of being closer to a wider range of business opportunities for which though a peer requires participating peers.

5. Is it possible to perform an example/experiment to validate the previous 2 points?

A number of experiments are planned, however, in order to see how the network behaves under different conditions we need to define the relevant parameters as realistically as possible. This will involve incorporating data on actual or proposed use of the network taken from 'real-life' scenarios. The strong links we have with the social scientists will support and facilitate this process allowing empirical data to be channelled to us.

5 Identity, Accounting, and Trust in the OPAALS DE

We have seen (Section 2.2.2) that trusted relationships are of significance in a variety of social science disciplines and particularly in theories underpinning the building of active communities. Literature and empirical research in these fields acknowledge that trust is a critical enabler of e-business and the foundation of the digital economy. We have also seen that the notions of identity and reputation constitute key elements of trust both in social science and computer science. Hence, the core DE architecture must include the provision for managing identity. Any user entity on the infrastructure must be given an identity so that all peers can be identified when engaging in interactions (e.g. business transactions, knowledge services) so that it can be accountable for its actions. The need for this has been also identified as a key requirement (see R8 in Section 2.1.5) from the user perspective, that of regional SMEs. In this chapter we focus on trust, identity, and accounting from the point of computer science and describe the distributed models for establishing these notions in the core DE architecture in a way that reflects their understanding in social studies. In the final section of this chapter, we include a discussion on how the dialogue between the two viewpoints has been established. In what follows, we give an outline of the key elements while the full details of the models can be found in Deliverable 4.1 and 4.2 and 4.3 of OPAALS.

In recent times the terms "accountability", "governance" and "trust" and their inter-relationship have become increasingly popular research topics across a wide range of disciplines. Bullock [Bul06] provides a model linking governance, accountability and legitimacy (trust in the system), which is a useful reference point in showing the relationship between these subjects and how changes in governance can ultimately reduce trust in the system.

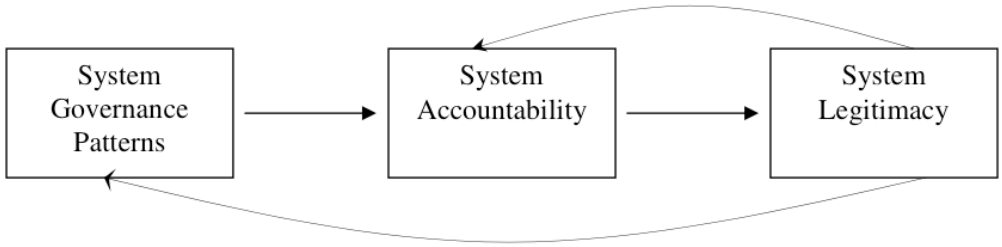


Figure 1. Bullock's Governance, Accountability and Legitimacy Model

In the model (Figure 1), changes in a system's governance pattern lead to a loss of accountability, which in turn depending on the degree of accountability loss can lead to a loss in the systems legitimacy (trust) in the system [Bul06].

Wang and Singh in [WaS07] recognised this relationship between trust and accountability and 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.

Work performed in the DBE project recognised this importance and categorised trust in such systems as three types [MTV03], [DBE-32.1]:

- **Trust Type X:**
  The trust that participants have that the technology and that it is capable of facilitating trustworthy relationships in business activities.

- **Trust Type Y:**
  The trust that the current members of a community that actions of new members joining the community can be accounted for.
- **Trust Type Z:**
  The trust that can be established among participants within the bounds of the system

The availability of accountability facilities satisfy these needs as follows:

Trust Type X: Accountability provides participants with an assurance that their activities and those of others can be proven without doubt in the case of disputes.

Trust Type Y: Accountability provides confidence in current members that adverse or malicious behaviour of new participants can be recorded and verified.

Trust Type Z: Accountability provides non-repudiable evidence of activities which can be used to evolve trust relationships among participants.

In order to provide these accountability facilities and trust mechanisms it is a basic requirement to provide a distributed identity system to track the identities of entities being accounted. Each of these models is described briefly below and an integrated view of these approaches follows these descriptions.

5.1 Distributed Identity model

The identity model was initially developed in WP4 (Figure 2) and is currently being refined with a view to implementation. The current work is focussed on an extensible, SAML [SAML] v2.0 based (Figure 3) model.

The following decisions were made in refining the model:

1. The model is based around SAML v2.0 Liberty Alliance standards, for the reasons given below (Section 5.1.1).

2. A meta model and tool kit was to be developed to support all possible models in order to provide maximum flexibility in the design of individual models. (See Section 5.1.2.)

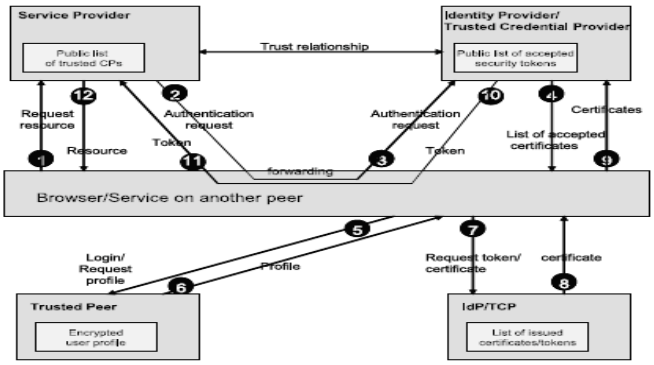


Figure 2 Initial Identity Model

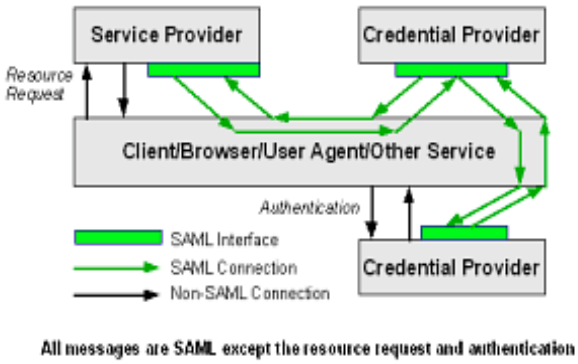


Figure 3 Pure SAML Approach

5.1.1 SAML Based

SAML is an implementation agnostic, open standard for security message passing between entities attempting to assert identity statements. It was decided to adopt a 'pure SAML' approach, whereby all entities are capable of generating and consuming SAML assertions; rather than a less prescriptive approach where entities must interpret various, potentially proprietary, potentially closed-source identity messages, certificates or tokens in order to assert identity statements. This is important since such statements may need to be asserted between entities in pre-existing, heterogeneous networks or federations, where a variety of identity technologies are in situ. The pure SAML approach has the drawback of requiring all entities to implement a SAML interface, but greatly simplifies the design, and most importantly, the maintainability and extensibility of identity models and implementations. In the pure SAML case, there is also no strict requirement for a user profile, as the identity model does not have to deal with and store heterogeneous tokens or credentials, but only pass SAML assertions.

It is also to be hoped that greater use of SAML will contribute to an increasing convergence on open identity standards.

5.1.2 Extensible Approach

A powerful, extensible meta model and tool kit was developed to support the development of identity models. This approach simplifies the design and modification of identity models. It also allows an identity model(s) to evolve alongside other developments in the OPAALS project, such as the work on the P2P infrastructure and potentially the work on trust and accountability, on which any model is dependent.

Taking inspiration from the SAML concepts of Profiles and Bindings, we designed a meta-model for identity models based on concepts such as Operations, Profiles, Connections and Actors, as described below.

The SAML v2.0 Single Sign-On (SSO) Profile outlines the connections between the entities, User Agent (UA), Service Provider (SP), and Identity Provider (IdP) required to accomplish a single sign-on task. This profile can be extended (or modified if necessary) to encompass an OPAALS identity model. The SAML SSO profile prescribes certain connections, but is agnostic of others, such as those required between the UA and SP to access a resource and such as those required for the Identity Provider to identify the principal (see Figure 4). Extending SAML's conception of profiles and bindings to the complete single sign-on solution, we can define the connections required to accomplish these functions (that SAML is agnostic of) as profiles

themselves, using particular bindings. We define this 'complete' solution as an Operation (or precisely an SSO Operation), which comprises a set of profiles, each performing its own modular task.

Figure 4 overlays this concept of an SSO Operation, outlining additional, non-SAML profiles, on a SAML SSO protocol flow or connection diagram from the SAML Profile v2.0 specifications. From the figure we see that in addition to the SAML SSO Profile, which is described, an SSO Operation must also give the 'Request Resource Profile' and the 'Authentication Profile', which are not detailed in the figure.

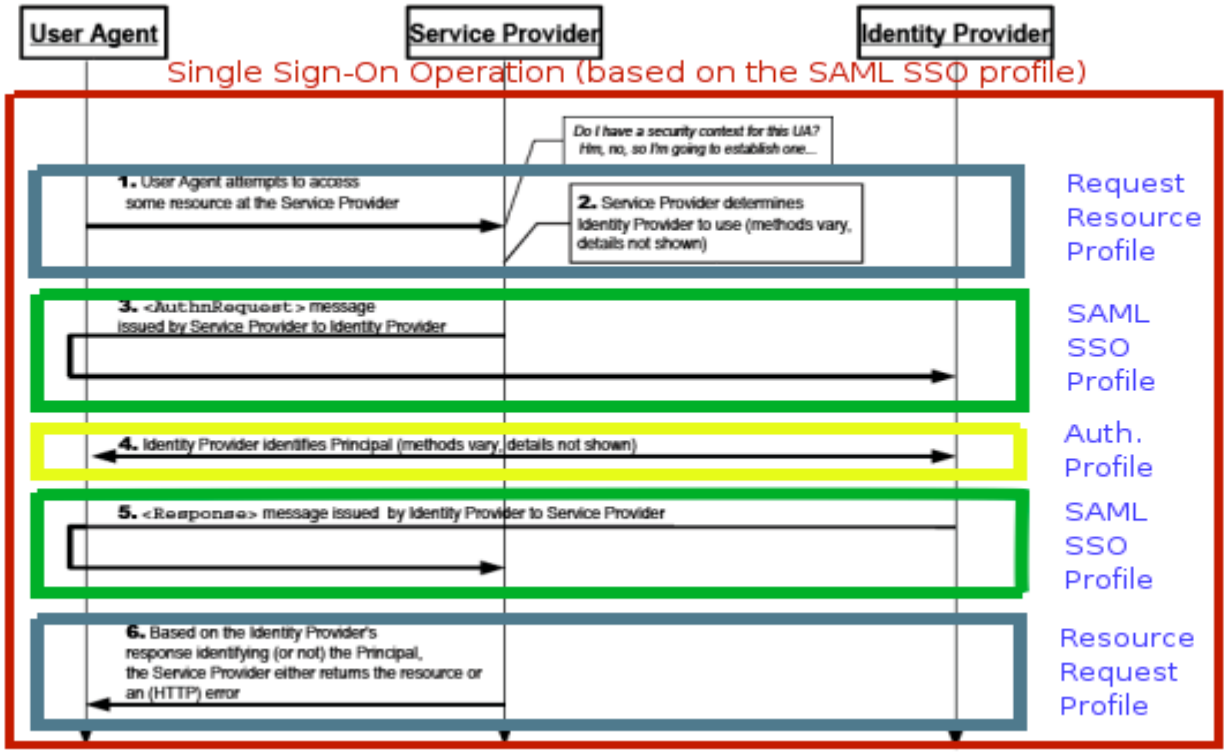


Figure 4 An Operation as a collection of Profiles

We model single sign-on, and other identity related tasks between entities, as Operations, consisting of Profiles. We further define Profiles to be sets of Connections between Actors, where Connections constitute message passing (e.g. requests or responses) between entities, and Actors are the entities themselves. In order to specify how messages are passed between Actors, we define the type of Connection used, and in order to specify a new entity in the Operation we define a new Actor. From a modelling point of view we have, therefore, the building blocks to describe any Operation providing any identity task.

Figure 5 illustrates the relationship between Operations, Profiles, Connections and Actors. Profiles are built from Actors and Connections and the protocol flow can be depicted by a digraph where the nodes are Actors and the arcs are Connections, as shown in Figure 6. Figure 7 gives an example of an Operation consisting of two Profiles. Profile A gives the Connections between Actor A and Actor B in the Operation (as illustrated in the protocol flow), while Profile B gives the Connections between Actor C and Actors A and B.

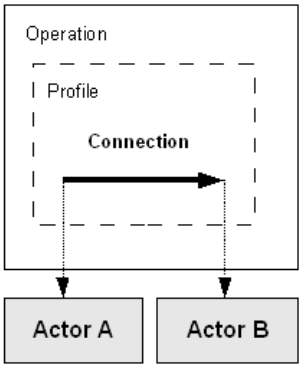


Figure 5 Identity Model Components

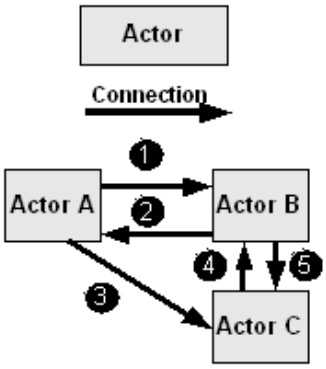


Figure 6 Profiles are built from Actors & Connections

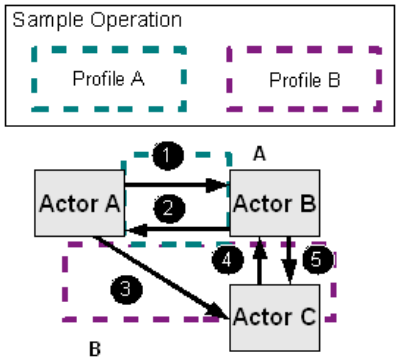


Figure 7 Sample Operation consisting of two Profiles

5.1.3 An Evolving Identity Model

An example of an Identity Model is given in
Figure 8. The figure outlines one potential identity model that may fit the needs of the OPAALS project. In this identity model it is assumed that Connections are HTTP GET Requests or HTTP GET/POST Redirects, and that the UA is a browser-like user agent. Integration with the work done on trust is also assumed, with the addition of the Trust Overlay in the figure, which may or may not be part of the SSO Operation. What Connection implementations are used and how trust is established between IdP(sp) and IdP(ua) is, however, potentially dependent on developments in the work on the P2P infrastructure and on trust. It is therefore of particular importance that parts of the Operation (e.g. particular Connections or Profiles) can be re-implemented modularly.

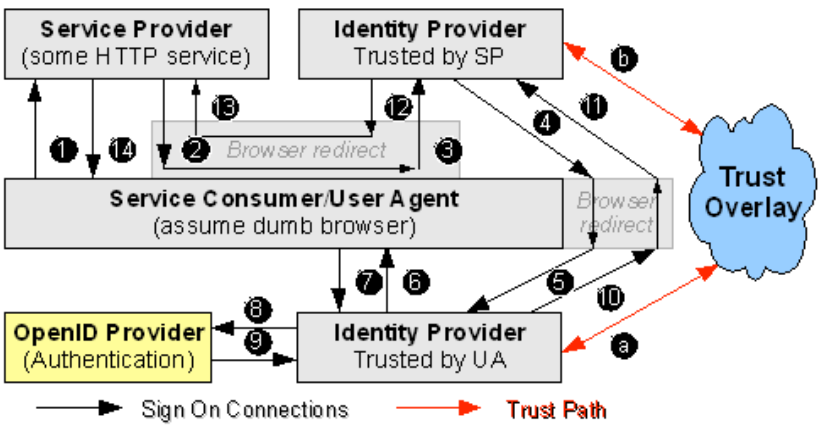


Figure 8 Evolving Identity Model

5.2 Distributed Accountability Model

In WP4 in Phase I of OPAALS a distributed Accountability model was described in Deliverable D4.2 [MaJ07]. The model was realised with a set of technical requirements derived from work performed during the DBE project. We note that these requirements are in line with the P2P network design and the distriburted transaction support in OPAALS, which were described in the preceeding chapters. The requirements for the distributed accountability model are as follows:

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.

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.

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.

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.

Contracts

Exchange of contracts or service level agreements prior to service consumption is a vital aspect of a commercial application of digital ecosystem deployment. It is desirable that aspects of these agreements can be assessed at runtime to ensure that parties are operating within the bounds of the agreement.

A review of the state of the art of accountability models was performed with a view to discovering if any existing models satisfied these requirements. While no single model was sufficient for our needs a combination of the two models [ZhL07], [HaS05] provided most of what was required.

The distributed accountability model is shown below in Figure 9. See the OPAALS Deliverable D4.2 [MaJ07] for a more complete discussion.

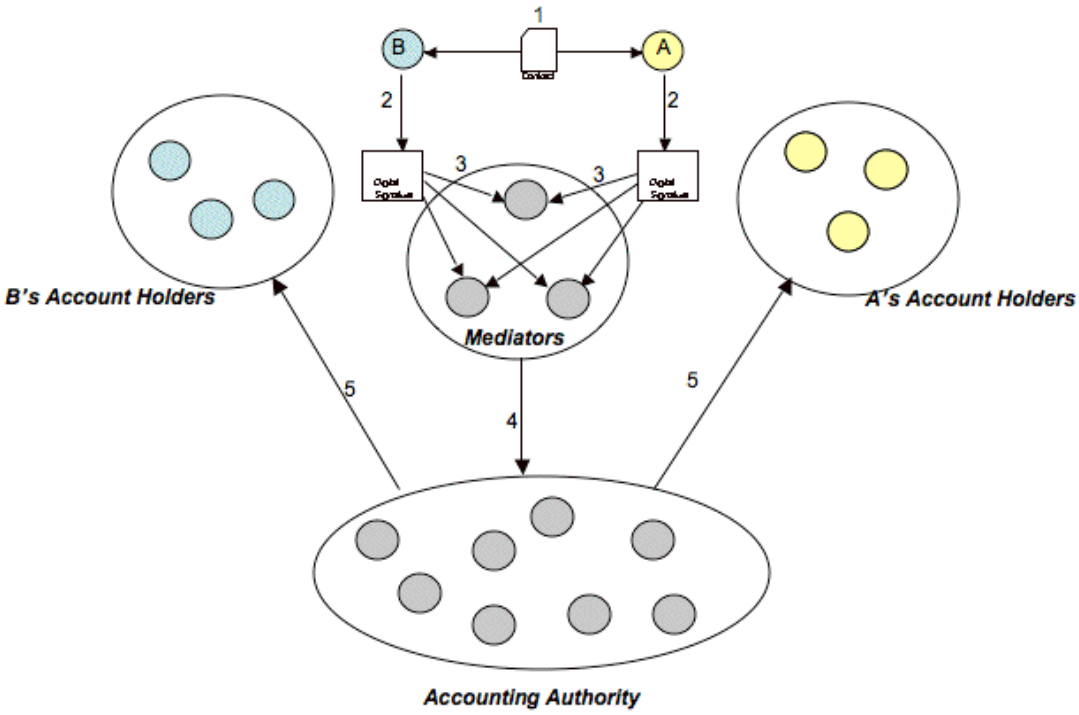


Figure 9: Distributed Accountability Model

5.2.1 The Roles

Each small circle represents an entity/node in a digital ecosystem. Nodes A & B are interacting via some service provision or resource exchange. Each of the rest of the nodes in the model are selected as needed. The actual selection of which nodes are used in each role is dependent on the actual infrastructure of the ecosystem. One such selection process (as in the explanations below) would be where each node is provided with a 128-bit ID ala a DHT and selection of appropriate nodes is achieved by performing a hash function on one or more of these IDs.

Account Holders

The role of the Account Holders is to hold long-term information about a node's accounted activity. The selection of the nodes would, in the DHT example, be based on performing the hash function on the accounted node's ID and the n nodes whose IDs are closest to the result would be selected for that nodes account holders, where n is the smallest number of nodes necessary for the network to sustain. If one of these nodes leaves the system, the hash function is again performed on the accounted node's ID and the next closest node takes its place. The data from the remaining nodes is copied to the new node entering the group for consistency reasons.

Mediators

The role of the Mediators is to ensure that neither A or B reports false activity during the course of the transaction. Performing a hash function on a combination of A and B's IDs and a timestamp is used for the selection of the mediators. A number of nodes whose IDs are closest in the system to the has function result are selected as the mediators. The lifetime of this mediator group is limited to the lifetime of the service interaction.

Accounting Authority

The Accounting Authority is responsible for ensuring the integrity of the accounted data in the case of composed services where more than one service is combined to perform a task. The selection of these nodes in the DHT case could be the identity of the master service.

5.2.2 The Process

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 charge record by the Accounting Authority. These charge records are delivered to the corresponding set of account holders for both A and B.

5.2.3 Further Work

The model as described in Deliverable D4.2 of OPAALS has limitations in terms of securing access to accounted data. Work is ongoing in developing a schema for data confidentiality and access control based on a public key encryption approach.

5.3 Distributed Trust model

5.3.1 Evolutionary Trust Model

This section presents the Evolutionary trust model we propose for DEs. The model is based on a multidisciplinary framework first introduced by us in [TeK07] and illustrated in Figure 10. 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.

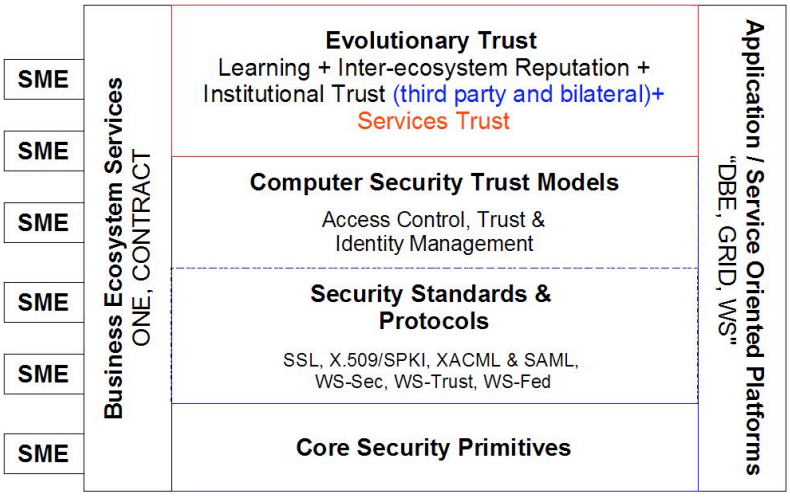


Figure 10: A Multidisciplinary Framework for Digital Ecosystems

We target a model that would provide trust at different levels:

- Ecosystem (Infrastructure and business domain)
- Services
- Knowledge (information or data)
- User

In order to provide trust at all these levels, we will monitor service execution using an accountability model. Using this together with a reputation model and evolutionary model would achieve trust at all different levels.

As motivated previously, for defining a trust model for Digital Ecosystems, we need to consider the following components:

1. Distributed identity management
2. Reputation
    a. Decentralized: Peer-to-peer reputation
    b. Centralized: Trusted rating agencies
3. Learning: evolutionary trust

Figure 11 shows these components and the relations between them.

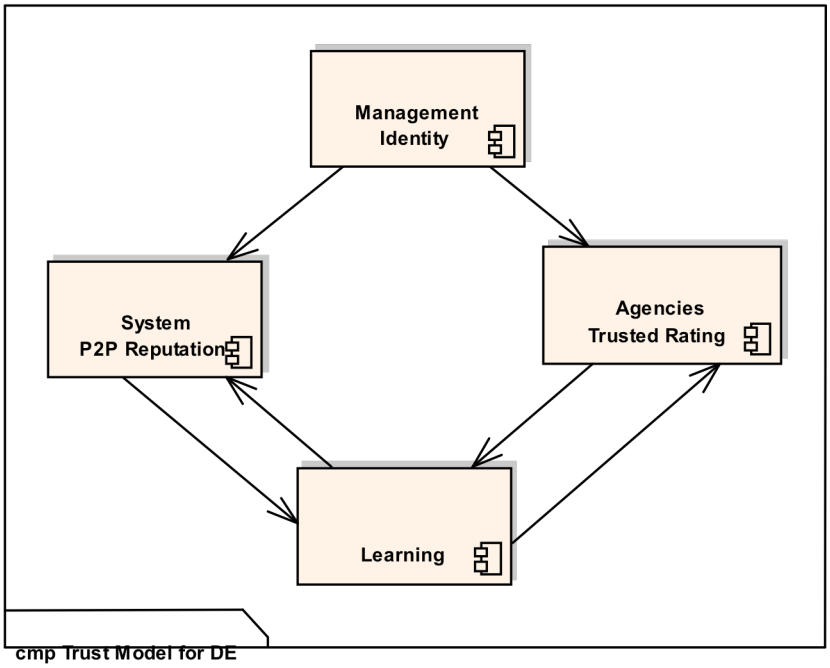


Figure 11: Trust Model Components

The Distributed identity management component already introduces a certain level of trust and security into the DE by assigning to each user an identity certified by other trusted entities in the system.

The Peer-to-peer reputation system and the Trusted Rating Agencies component rely on the Identity Management model and assign reputation values to DE identities. Users' reputations are built and evolve into the DE through transaction interactions with other users. To update the reputation values, the Peer-to-peer reputation system and the Trusted Rating Agencies run algorithms on transaction feedbacks.

Though the two components appear to have the same functionality, they actually address different aspects and needs of the DE. The P2P reputation system views peers in the DE as equal and self-organizing entities which cooperate with each other to compute the reputation values. Though any peer in the DE can be a rating agency and users are free to choose which agency to trust, the Trusted Rating Agencies model introduces a certain level of hierarchy which results in higher credibility of the rating certified by an agency. This model is suitable for commercial transactions with certain restrictions.

To learn from past transactions and users' behaviour, historical data produced by the two systems can be input by a Learning component. This component can derive new rules and patterns used by the two systems to improve their rating algorithms. In this way, the trust model we propose evolves in time together with the DE.

5.3.2 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. Note that this is consistent with Gambetta's widely accepted definition of trust as "*a particular level of the subjective probability with which an agent assesses that another agent or group of agents will perform a particular action, both before he can monitor such action and in a context in which it affects his own action*" [JIB07]. If trust is to be a probability measure, then the (0,1) range is natural.

5.3.3 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 12 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 12):

(1) Experience reports: Transaction engine → Trust manager

(2) Trust recommendations: Trust manager ↔ Trust manager

(3) Policy updates: Trust manager → Transaction engine

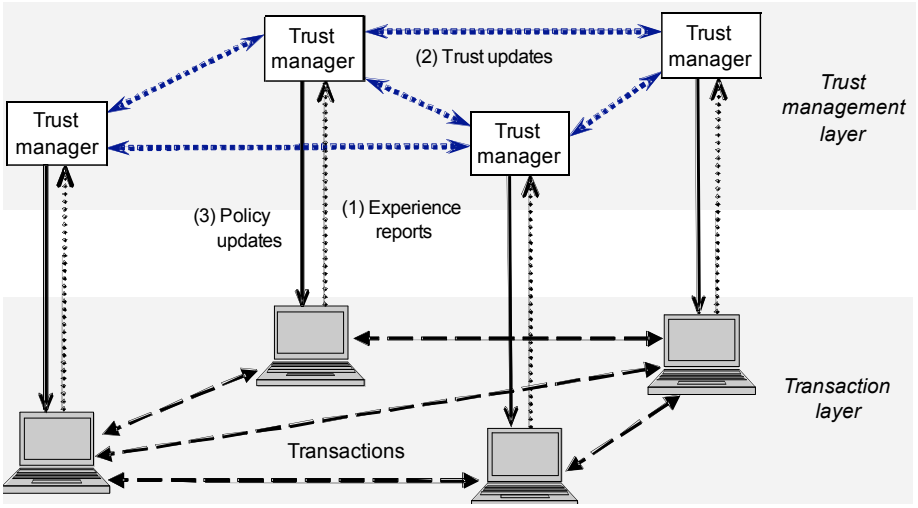


Figure 12: Trust overlay helps to secure transactions in a digital ecosystem.

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* [Dou02], 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 may depend on 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. It might be instructive for the reader at this point to refer back to the design concept of DVSPs in the P2P network, as this trust score could be added to the measurement of stability for selecting best candidate peers for a DVSP. The relevant discussions are in Sections 4.3 and 4.4.

5.4 Integrated view

Here we describe an integrated view of distributed identity, accountability and trust. The basic relationships are as follows:

- Identity provides a basis for both accountability and trust, i.e. who (or what) is being held accountable and trusted.
- Accountability provides evidence (via usage data) used in determining trust.

5.4.1 Integrating Identity and Trust Models

We first examine how the trust overlay described above integrates with our identity model in ascertaining the trustworthiness of identity assertions. In Figure 13 a Service Consumer requests access to a service provided by Service Provider. Both of the entities have an associated IdP. The identity assertions work in both directions, in that both entities wish to ensure that the other is indeed who they say they are. In order to do so, the associated IdPs exchange SAML assertions concerning the identity of their respective entities.

The IdP will need to determine how trustworthy this assertion is and requests from the trust overlay a trust value for the other IdP. This trust value has evolved over time via the process described above in Section 5.3. With this trust value the IdP can derive the degree of certainty that the other entity is who they say they are. This degree of certainty is passed to the respective Service Provider/Consumer and depending

on whether this certainty is above or below a pre-determined threshold, the service access may proceed. For example, SP1 may have determined for access to a ServiceX, there must be a confidence of above 90% in the identity of the requesting entity (i.e. 0.9 probability that SC1 is who it says it is). Similarly SC1 might have set a threshold of 60% confidence that SP1 is who it says it is before it is willing to consume ServiceX. Such thresholds allow for a flexible degree of service access, for example in the case of an expensive financial transaction, the SP would most certainly need to be 100% certain that the consumer was who they said they were, whereas in the case of access to information from a public forum there might not be a need for such a high threshold.

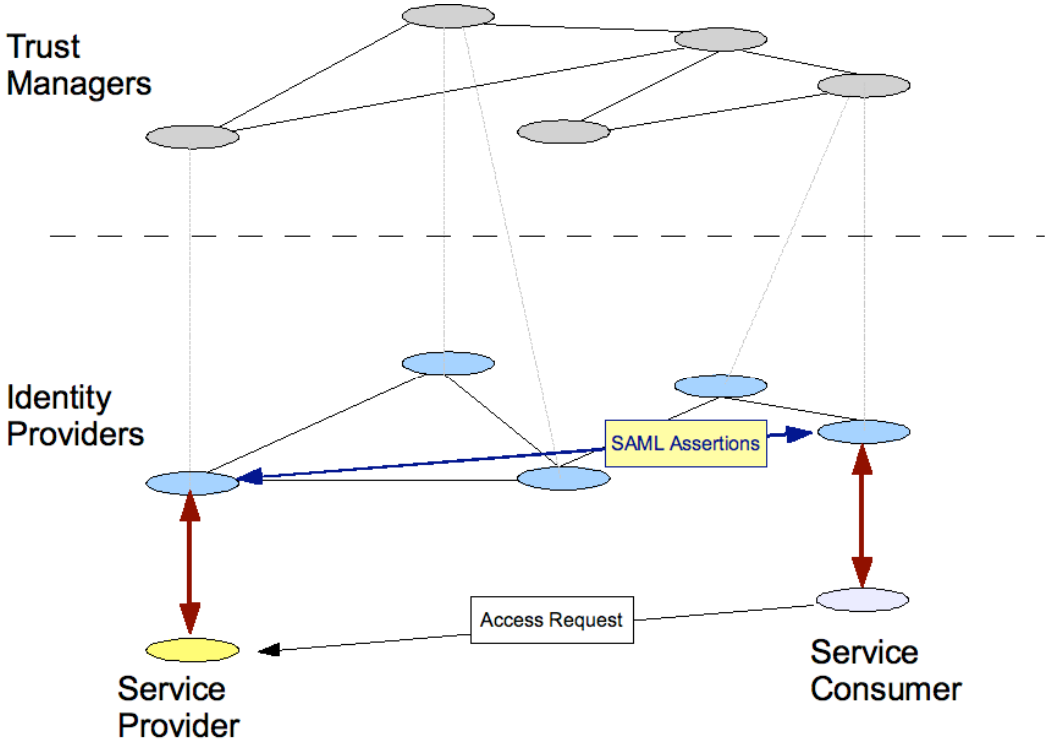


Figure 13: Distributed Identity Model Integrated with Trust Overlay Model

5.4.2 Integrating Accountability and Trust Models

In Figure 14 we describe an integrated view of the accountability model and trust model. We assume that identity has already been established and is above the threshold for both Service Consumer and Service Provider. Both entities request from their respective trust manager whether the other entity is trustworthy with the context of the current service request. It is important to note that this is not the same as ascertaining the identity of the other entity, but to determine the trustworthiness of that entity once identity has been established within the current context. When this trustworthiness is determined to be sufficient, the Service Consumer is allowed access to the service provided by the Service Provider. In accordance with our accountability model, the usage data is digitally signed and captured by a set of mediators (this aspect of the accountability model is not shown in this integrated view for clarity only). At the end of the transaction the mediators release the final usage data to the account holders. This final usage data (experience report) forms the evidence that is used to update the trust managers in the trust overlay, which in turn provide policy updates to the service nodes.

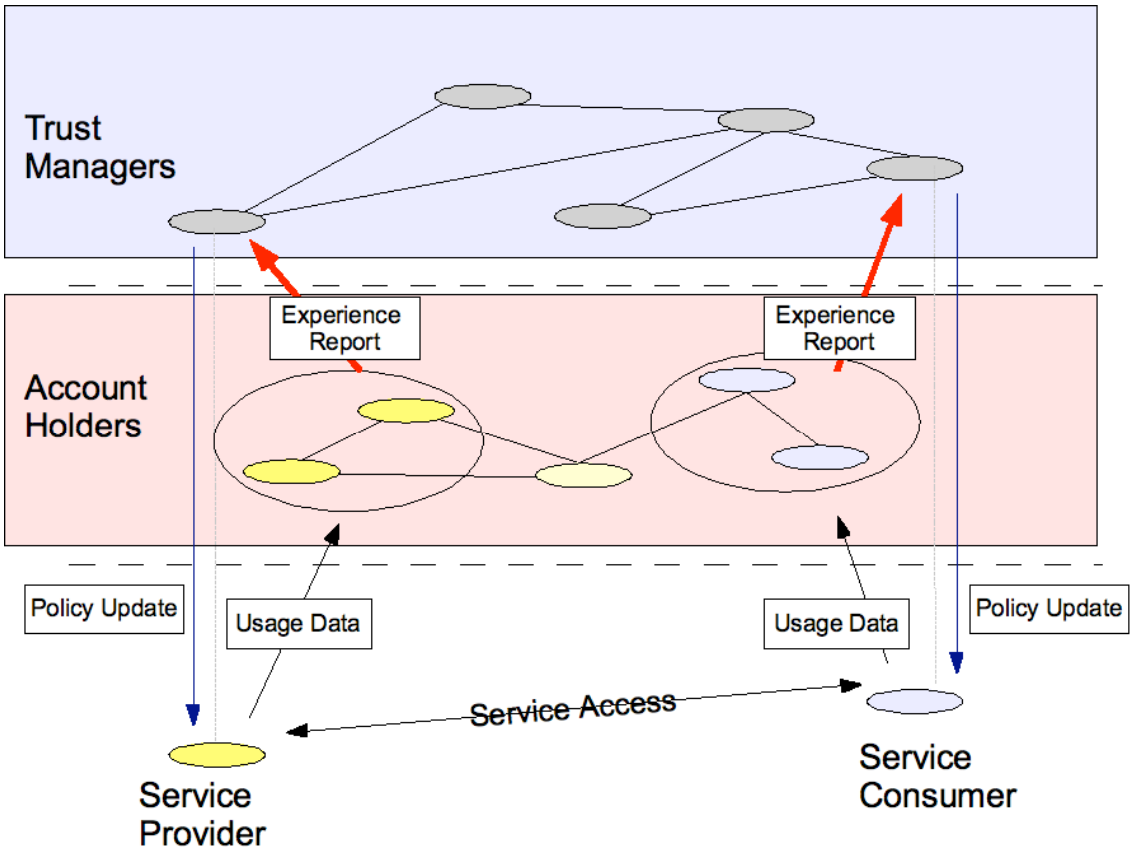


Figure 14: Distributed Accounting Model Integrated with Trust Overlay

(Simplified view without the accountability mediators or accounting authority)

5.5 Social science questions concerning identity, accounting, and trust

Within the relevant literature it is widely recognised that identity, trust and accounting form key areas in which electronic B2B activities are 'funneled' through centralised service providers. Although from a technological point of view this configuration 'makes sense' because it allows for the efficient verification of personal data the question of whether this is the most secure approach is debatable. There are grounds for arguing that holding large amounts of significant data in one place carries more risk than having data distributed between different providers. From a social science point of view, trust is analysed from social, economic and psychological viewpoints and according to the latter there is considerable reassurance gained in potentially risky situations when the service provider guaranteeing an operation is well-known to the service user. It is virtually if not totally impossible for a distrusted technology to become economically viable since it will simply not be used. This finding was reflected in the results of research carried out with SMEs as part of the DBE project where security and reliability were cited as being 2 of the key areas of concern. In this regard, SMEs pointed to the need for a governance model capable of consolidating these two aspects of use. Yet, here again, the desire is for some kind of unified, centralised authority that can guarantee transactions and oversee technological operations.

From this example in particular we can see the iterative relationship between the social and the technological. The psychological and economic advantages to having dominant authorities manifests itself in network and system design. This in turn generates social consequences which then require further attempts to counter. In this respect, digital ecosystems can be described as a policy-driven counter-measure to tendencies for centralised design and domination within the B2B technology environment.

For the above reasons, the questions OPAALS' social scientists ask about trust, identity and accounting in the core architecture design are focused around 3 specific dimensions:

1. Does this element of the core architecture design conform to the digital ecosystems requirement for decentralised system and network design?

2. Can the system proposed offset risk and inspire sufficient trust for it to enable an economically viable ecosystem infrastructure?

3. What kinds of governance conditions and measures can underpin digital ecosystems trust, identity and accountability?

Technological implementation of the concepts described in this section including integration with the core architecture will no doubt raise numerous technical questions, which may in turn be cause for debate among OPAALS social scientists. However, from the detailed design overview provided here, the trust, identity and accountability model appears to be highly innovative with respect to decentralised design. The trusts ratings and the accumulation of recognition within the network mirrors social processes of building trust. Rather than trying to define and centrally classify the identity of every user, this approach is based on the trustworthiness of identity assertions. This is much closer to a social model of trust than a technologically enabled one (in which large amounts of data can be classified, stored and searched). It also captures the dynamic relationship of trust in social networks where a trust breakdown leads to a sudden or gradual disassociation from a network. In terms of approach, this model of enabling trust relationships is much closer to a systems theory approach than it is a classical, scientific approach whose primary concern is with classification. From this point of view, focus rests on the dynamic 'sets of relations' formed rather than on categorically defining every object involved in a network of interactions. These research issues are pursued further in Deliverable 12.3.

In the current state of digital ecosystems development, question 2 is and will be one of the most fundamentally important. For the main part, the answer to this question can only be provided by digital ecosystem users themselves. Although it is possible to carry out simulations of how trust relations operate within a network, the modelling techniques used can rarely capture the complexity of the broader environment. Proof that the system is sufficiently secure and therefore inspires trust will only come from its use in risk-bearing transactions.

The governance requirements for digital ecosystems have always been a cause for significant debate and concern. Finding an organisational and decision-making framework that can mirror the de-centralised, distributed attributes of the technological architecture is a major challenge and an area of innovation in and of itself. Legal requirements for underwriting contracts and guaranteeing operational aspects of the architecture can lead to an inflexible technology environment based on fixed legal entities. Although the overall characteristics of digital ecosystem governance will have a profound impact on the way trust, identity and accountability are managed, it is these socio-legal challenges of that present one of the most significant challenges. These challenges are pursued further in task 12.2.

6 Concluding remarks and future directions

This deliverable provides a detailed description of the digital ecosystems core architecture, but it also goes beyond this, describing the critical, interdisciplinary dialogue that has underpinned architectural design decisions and the social science theory that has informed that critique. The interdisciplinary consensus building process described here will continue throughout the duration of the OPAALS project. Beyond this, it is hoped that valuable lessons in the distinctive kind of inter-disciplinary dialogue developed here will go on to inform the 'culture of communication' that can constructively support digital ecosystems governance.

Although a level of consensus has been reached over the conceptual and operational compatibility of the core architecture design and key social science lines of enquiry a number of very significant research challenges remain. These include achieving a functional implementation of the core architecture; integrating this with the Servent; and exploring the implications of these combined elements for the 'open knowledge space'. In addition, it is becoming increasingly important to re-engage with regions and individual SMEs in order to understand the 'real life' implications of the digital ecosystem technology design. One of the most significant and repeated findings of technology-focussed, social science research is that local interpretation and use of technology will always differ from that which was imagined prior to implementation. Considering the autonomous and self-maintaining SMEs identified in this report, this finding seems a particularly pertinent one to observe.

Technologically speaking, SME engagement with the digital ecosystem began with the trial integration of services with the DBE Servent. Therefore, to maintain any level of continuity with those regions who have existing experience of digital ecosystems technology, a link between the core architecture described here and the Servent needs to be clear. To that end, we have described an architecture in which the P2P and transaction support can overlay a network of Servents and can continue to support upgrades of the Servent to optimise its behaviour and offer additional functionality. This means that there is a clear upgrade route if regions using the Servent wish to continue to do so. Upgrades were shown in this report to be of tremendous significance to small and very small businesses who need backwards and lateral compatibility. In the meantime, the core DE architecture has a number of additional characteristics that come from the advanced support for coordinating distributed transactions locally and a P2P network that continuously adapts to ensure optimal functioning with regard to its vital characteristics (e.g. connectivity, resilience to attacks, re-configuration to off-load traffic, etc.).

Our work is directed at supporting the paradigm change in business modelling from linear value chains to *value networks (Allee, 2000)*. This line of thinking involves looking at the network dynamics [All00]. There are two keys aspects to this shift in understanding of the dynamics of business processes. Firstly, business typically does not operate in a linear, assembly-line like, flow of material enhancements to tangible assets. Instead, it is conducted through a network of collaborations that generate value in different, mutually beneficial, ways. Secondly, both tangible, and *intangible* assets generate "value" (Dini et al., forthcoming).

Although there are a number of larger organisations that have facilitated significant transformations by analysing their business from a value network perspective, this more "ecological" approach to business has been little exploited by the sector that stands to benefit the most from this: Small-to- Medium Enterprises (SMEs). Part of the reason for this is that such analyses and, more importantly, the computational infrastructure that supports the business transformation, is centralised on a single "governing" organisation. This provides a significant barrier to adoption by SMEs, as that central point of control will provide an unacceptable degree of governance of the participating SMEs.

In contrast, our work on the core architecture of the digital ecosystem in OPAALS aims to push down this barrier to adoption by providing a computational infrastructure that supports open collaborations of SMEs that enables a business community to evolve to meet emerging business opportunities, without

violating their local autonomy – they only reveal what they choose to reveal to other participants, hence maintaining control over both tangible and intangible assets and their own business strategy.

Our work on distributed identity, accountability and trust ensures that SMEs can transact reliably and transparently in the DE, independent of any central governing authority and/or large, commercial identity provider. There is therefore no need to share potentially sensitive or private information with third parties in order to participate in the DE.

In this deliverable we have focused on the core aspects of the DE architecture, namely the P2P and transaction support as well as trust, accountability and identity, aiming to foster an environment for open and trusted collaborations and the free flow of information between SMEs in a digital ecosystem. However, there are certain aspects of the core architecture that can facilitate the provision for the collaborative space for knowledge generation considered in the development of the Open Knowledge Space (OKS) in OPAALS. This can perhaps be most clearly seen in the support for collaborative editing.

In order for two (or more) people to edit the same document remotely and in real time, typically Write access to the document's data needs to be granted only to one person at a time. The other participants can only have Read access while the doc is being modified. In other words, to avoid more than one persons editing the same part of the document, and the associated risk of losing all edits in the end, one person can edit (effectively locking the document for the duration of their editing) while the others wait (are effectively locked out). In fact, this is the concept behind the conventional Shared / eXclusive (S/X) lock mechanism. Advances in web applications could be exploited so that the activity of collaborative knowledge generation can be improved. e.g. by allowing more than one person to edit at a time (on different parts of the shared document), or allowing people to take turns more quickly, to better realise the essence of real time collaboration. This would require however, a more sophisticated and flexible lock mechanism that could handle the more fine-grained interactions involved. This is where the lock schemes of the transaction model outlined in Section 3.5 of this report, and in full detail in Deliverable D3.2, could be used to enhance the capability of the online collaborative editing tool. A transaction, in other words, is not just a B2B sale/purchase. It can also be a remote editing action, like typing a word or changing the font size. Transaction coordination means that one user is allowed to 'deliver its service' (i.e. edit the doc) while the others have to wait for their turn in the 'service execution tree'. It is in this sense something that benefits the transaction model (and so relates to the business/economic perspectives) can also benefit the other manifestations of open knowledge (and so, is also related to the social/community perspectives).

In this report we have demonstrated the ongoing consensus building process in OPAALS and have argued that the careful examination of the core aspects of the DE architecture from a social science viewpoint shows the ability of this architecture to address power/control/monopoly issues and protect the democratic nature of DEs. The debate around these issues and the consensus building process will be ongoing. It has been proposed that the next set of cross-domain meetings tackles the implementation and research question surrounding digital ecosystems as an enabler of knowledge sharing and creation. This is an extremely dense and complex area of debate that will no doubt produce interesting and successful outcomes.

7 References

[AcK99] Achrol, R.S., Kotler, P. Marketing in the network economy. *Journal of Marketing*, 63: 146-63, 1999.

[All00] Allee, V. Reconfiguring the Value Network. Journal of Business Strategy, 21(4), Jul-Aug 2000

[Ash56] Ashby, W. R. *Introduction to Cybernetics*. London: Chapman and Hall. 1956 http://pespmc1.vub.ac.be/ASHBBOOK.html. (last accessed 15.02.05).

[AuG97] Auger, P., Gallaugher, J.M. Factors affecting adoption of an Internet-based sales presence for small businesses. *The Information Society*, 13(1): 55-74, 1997.

[BSH98] Baldwin, R., Scott, C., and Hood, C.. *A Reader on Regulation*. Oxford: Oxford University Press, 1998.

[BaC99] Baldwin, R., & Cave, M. *Understanding regulation: theory, strategy, and practice*. Oxford; New York: Oxford University Press, 1999.

[BaA99] A. L Barabasi, R. Albert. Emergence of Scaling in Random Networks. Science 286(5439): 509–512, 1999.

[Ber02] Berkey, J. Outline of International e-commerce regulatory issues. Intel/Unitar Campus of New Information and Communication Technologies and Diplomacy, New York, US. 2002. Available at http://www.un.int/unitar/intel_nct_campus/2002/conference_presentation.htm (last accessed 15.02.05).

[BG-M03] 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

[Bon01] Bond, J. Business uses of Peer to Peer (P2P) technologies. Netmarkets Europe White Paper, January 2001.

[BMM05] 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.

[BrH03] Brynjolfsson, E., and Hitt, L. Computing Productivity: Firm-level Evidence, *Review of Economics and Statistics*, June 2003.

[Bul06] Bullock, G., 2006, *Governance, Accountability, and Legitimacy*, Working Paper Series, Consumer Information Laboratory, University of California, Berkeley, available at http://nature.berkeley.edu/infolab/files/u3/InfoLab_WP06-01_Governance.pdf (checked 09/07/2008)

[BuF04] 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.

[BHF05] 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.

[CCJ+04] Cabrera, F. L., Copeland, G., Johnson, J. and Langworthy, D. Coordinating Web Services Activities with WS-Coordination, WS-AtomicTransaction, and WS-BusinessActivity. Available at: http://msdn.microsoft.com/webservices/default.aspx, January 2004.

[CDF+03] Ceponkus, A.; Dalal, S.; Fletcher, T.; Furniss, P.; Green, A. and Pope B. Business Transaction Protocol Version 1.0, An OASIS Committee Specification, 3 June 2002. Available at: http://www.oasis-open.org/committees/download.php/1184/2002-06-03.BTP_cttee_spec_1.0.pdf

[Cla02] Clarke, R. Trust in the context of e-Business. *Internet Law Bulletin* 4(5): 56-59, 2002.

[Col94] Coleman, J. S. *Foundations of Social Theory*. Cambridge; MA; London: First Harvard University Press, 1994.

[DaS06] Z.B. Daho, N. Simoni. Towards Dynamic Virtual Private Service Networks: Design and Self-Management. In Proc. *10th IEEE/IFIP Network Operations and Management Symposium* (*NCMS'06*), 2006.

[Dat96] Date C. J. *An Introduction to Database Systems*. 5th Edition, Addison Wesley, USA, 1996.

[DaP90] Davey, B. A. and Priestley, H. A. *Introduction to Lattices and Order*, Cambridge University Press, 1990.

[Del06] Deleuze, G. Foucault, London: Continuum, 1986/2006.

[DBE06] Digital Business Ecosystems (DBE) EU-FP6 IST Integrated Project No 507953. Available at http://www.digital-ecosystem.org

[DBE-32.1] The DBE Project, Regulatory Framework, A Literature Review. Available at http://www.digital-ecosystem.org

[DFM00] Dingledine R., Freedman M., and Molnar D. Accountability measures for peer-to-peer systems. In A. Oram (Ed) *Peer-to-Peer: Harnessing the Power of Disruptive Technologies*, pp. 271-340, O'Reilly and Associates, 2000.

[Din07] Dini, P. A Scientific Foundation for Digital Ecosystems. In Nachira, F., Nicloai, A., et al (eds), *Digital Business Ecosystems*, pp. 24-47, Information Society and Media, European Commission, Luxembourg: Office for Official Publications of the European Communities, 2007.

[DLM+08] Dini, P., Lombardo, G, Mansell, R. Razavi, A., Moschoyiannis, S., Krause, P., Nicolai, A, Leon, L.R. 'Beyond interoperability to digital ecosystems: regional innovation and socio-economic development led by SMEs' *Int. Journal Technological Learning, Innovation and Development*, 2008. *To appear*

[Dou02] Douceur, J. The Sybil Attack. In *Proc. Int'l Workshop on Peer-to-Peer Systems*, March 2002.

[D-DJA02] Drakopoulou-Dodd, S., Jack, S., Anderson, A.R. Scottish entrepreneurial networks in the international context. *International Small Business Journal*, 20(2): 213-19, 2002.

[DuS05] Dutton, W. H. and Shepherd, A. Confidence and Risk on the Internet. In R. Mansell and B. S. Collins (eds) *Trust and Crime in Information Societies*, pp. 207-244, Cheltenham: Edward Elgar, 2005.

[EC07] European Commission. ICT – Information and Communication Technologies. Work Programme 2007-08. Draft version 1, 2007.

[Fle94] Fleck, J. Learning by trying: the implementation of configurational technology. *Research Policy*, Vol.23, No.6, pp. 637-652, 1994.

[FADA] http://fada.sourceforge.net/

[Fou70] Foucault, M. *The order of things: An archaeology of the human sciences*. London: Routledge, (1970/2002).

[FDF+04] 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

[G-MS87] Garcia-Molina, H. and Salem, K., (1987), "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

[G-HWS06] R. A. Ghanea-Hercock, F. Wang, and Y. Sun. Self-Organizing and Adaptive Peer-to-Peer Network. IEEE Transactions on Systems, Man, and Cybernetics, Part B, Volume: 36, Issue: 6, 2006, pp. 1230-1236.

[HaS05] Hausheer, D., Stiller B., 2005, PeerMint: Decentralized and Secure Accounting for Peer-to-Peer Applications. In *NETWORKING 2005*, pp 40-52, Springer Berlin / Heidelberg, ISBN: 978-3-540-25809-4

[Hoa85] Hoare C. A. R. Communicating Sequential Processes. Prentice Hall, 1985.

[JIB07] A. Jøsang, R. Ismail, and C. Boyd, A survey of trust and reputation systems for online service provision, In *Decision Support Systems*, pp 618-644, March 2007.

[Kee00] Keen, P.G.W. Ensuring E-trust, *Computerworld* 34 (11), p. 46, 2000.

[Kri+] Kristiansen, L., et al. TINA Service Architecture and Specifications. TINA 1.0 Deliverables and Specification. Available at: http://www.tinac.com/specifications/specifications.htm

[LiM04] Z. Li, P.Mohapatra, "QRON: QoS-Aware Routing in Overlay Networks", Service Overlay Networks in the *IEEE Journal on Selected Areas in Communications* (2004).

[MacG04] MacGregor, R.C. Factors associated with formal networking in regional small business: some findings from a study of Swedish SMEs. *Journal of Small Business and Enterprise Development*, 11(1): 60-74, 2004.

[MaJ07] Malone, P. and Jennings, B. Distributed Accountability Model for an Autopoietic Peer-to-Peer Network. OPAALS project Deliverable 4.2, 2007. Available at: http://www.opaals.org

[MBC05] B. Martini, F. Baroncelli, P. Castoldi. A novel service oriented framework for automatically switched transport Network. In Integrated Network Management, pp. 295 – 308, 2005.

[MDS95] Mayer, R.C., Davis, J.H., Schoorman, F.D. An integrative model of organizational trust. *Academy of Management Review*, 20: 709–734, 1995.

[Maz88] 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

[MTV03] Meents, S., Tan Y-H. and Verhagen. T. Distinguishing different types of trust online B2B marketplaces. In *Proc. 10th Research Symposium on Emerging Electronic Markets*, pp. 53 – 65, 2003.

[Mil80] Milner A.J.R. Calculus for Communicating Systems. *Lecture Notes in Computer Science*, Vol. 92, Springer Verlag, 1980.

[Mos05] Moschoyiannis, S. Specification and Analysis of Component-Based Software in a True-Concurrent Setting. PhD Thesis, University of Surrey, 2005.

[MSK05] S. Moschoyiannis, M. W. Shields and P.J. Krause. Modelling Component Behaviour Using Concurrent Automata. In Proc. *ETAPS 2005 – Formal Foundations of Embedded Software and Component-Based Architectures* (*FESCA'05*), Electronic Notes in Computer Science, Vol. 141, Issue 3, pp. 199-220, Elsevier B.V., 2007.

[MKS07] S. Moschoyiannis, P. J. Krause, and M. W. Shields. A True-Concurrent Interpretation of Behavioural Scenarios. In Proc. *ETAPS 2007 – Formal Foundations of Embedded Software and Component-Based Architectures* (*FESCA'07*), Electronic Notes in Computer Science, Elsevier B.V., 2007. *To appear*

[MRZ+08] S. Moschoyiannis, A. Razavi, Y. Zheng and P. J. Krause. Long running Transactions: semantics, schemas, implementation. In *Proc. IEEE Int'l Conf on Digital Ecosystems and Technologies* (*IEEE-DEST'08*), IEEE Computer Society, 2008.

[MRK08] S. Moschoyiannis, A. Razavi, and P. J. Krause. Transaction Scripts: making implicit scenarios explicit. In Proc. *ETAPS 2008 – Formal Foundations of Embedded Software and Component-Based Architectures* (*FESCA'08*), Electronic Notes in Computer Science, Elsevier B.V., 2008. *To appear*

[Nac02] Nachira, F. Towards a Network of Digital Business Ecosystems Fostering the Local Development. Discussion Paper. http://www.digital-ecosystems.org/ (last accessed 15.02.05).

[SAML] OASIS Security Services (SAML) TC
http://www.oasis-open.org/committees/workgroup.php?wg_abbrev=security

[OMG05] OMG: UML 2.0 Superstructure Specification, Document ptc/04-10-02, 2004. Available at: http://www.uml.org

[ODC] Oxford Dictionary of Computing. *Oxford University Press*, 4th Edition, 1996.

[Pap03] M. P. Papazoglou. Service-Oriented Computing: Concepts, Characteristics and Directions. In *Proc. WISE'03*, IEEE Computer Society Press, pp. 3-12, 2003.

[PTD+06] 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.

[Pav02] Pavlou, P. A. Institution-Based Trust in Interorganizational Exchange Relationships: The Role of Online B2B Marketplaces on Trust Formation, *Journal of Strategic Information Systems*, 11(3-4): 215-243, 2002.

[RKM06] Razavi R.; Krause, P.J.; and Moschoyiannis S.; Digital Business Ecosystem Transaction Model. DBE Project Report D24.28, University of Surrey, 2006.

[RMM07] A. Razavi, P. Malone, S. Moschoyiannis, B. Jennings, P.J. Krause. A Distributed Transaction and Accounting Model for Digital Ecosystem Composed Services. In *Proc. IEEE Int'l Conf on Digital Ecosystems and Technologies* (*IEEE-DEST'07*). IEEE Computer Society, 2007.

[RMK07a] 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. Available at: http://files.opaals.org/OPAALS/Year_1_Deliverables/WP03/

[RMK07b] 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. Available at: http://files.opaals.org/OPAALS/Year_1_Deliverables/WP03/

[RMK07c] A. Razavi, S. K. Moschoyiannis and P. J. Krause. A Coordination Model for Distributed Transactions in Digital Business Ecosystems. In *Proc. IEEE Int'l Conf on Digital Ecosystems and Technologies* (*IEEE-DEST'07*), IEEE Computer Society, 2007.

[RMK07d] A. Razavi, S. Moschoyiannis and P. J. Krause. Concurrency Control and Recovery Management for Open e-Business Transactions. In Proc. *Communicating Process Architectures* (*CPA 2007*), pp. 267-285, IoS Press, 2007.

[RMK08a] A. Razavi, S. Moschoyiannis and P. J. Krause. A Scale-free Business Network for Digital Ecosystems. In *Proc. IEEE Int'l Conf on Digital Ecosystems and Technologies* (*IEEE-DEST'08*), IEEE Computer Society, 2008.

[RMK08b] A. Razavi, S. Moschoyiannis and P. J. Krause. A Self-organising Environment for Evolving Business Activities. In Proc. *Computational P2P Networks: Theory and Practice* (*CompP2P 2008*), IEEE Computer Society Press, 2008. *To appear*

[ReT00] Rehfeldt, M. and Turowski, K. Business Models for coordinating the next generation enterprises. In Proceedings of the *Academia/Industry Working Conference on Research Challenges* (AIWORC 2000), 2000.

[Sah+02] Sahai A. et al. Automated SLA monitoring for web services. In Proceedings of *13th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management: Management Technologies for E-Commerce and E-Business Applications*, p.28-41, 2002.

[SUS02] Shankar, V., G.L. Urban, and F. Sultan (2002) Online trust: a stakeholder perspective, concepts, implications, and future directions. *Journal of Strategic Information Systems*, 11: 325-344, 2002.

[Shi85] Shields, M. W.; Concurrent Machines. *Computer Journal*, 28:449-465, BCS, 1985.

[Shi97] Shields, M. W. *Semantics of Parallelism*. Springer-Verlag, London, 1997.

[Spa01] Sparrow, J. Knowledge Management in Small Firms. *Knowledge and Process Management*, (8)1: 3-16, 2001.

[Sto94] Storey, D.J. *Understanding the Small Business Sector*. London: Thomson Learning, 1994.

[SUG+02] Sultan, F., Urban, G et al. Determinants and Role of Trust in E-business: A Large Scale Empirical Study. MIT Sloan Working Paper Series, 2002. [WWW document] http://e-commerce.mit.edu/cgi-bin/viewpaper?id=231 (accessed 5th March 2007).

[SwR04] Swan, M. and H. Rosenbaum. The social construction of trust in e-business: An empirical investigation. *Americas Conference on Information Systems*, 2004. [WWW document] http://aisel.isworld.org/pdf.asp?Vpath=AMCIS/2004&PDFpath=SIGEBZ01-1766.pdf (accessed 5th March 2007).

[Tan35] Tansley, A. G. The use and abuse of vegetational concepts and terms. *Ecology* 16: 284-307, 1935.

[TeK07] 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.

[WaS07] Wang, Y., Singh, M., 2007, Formal trust Model for Multiagent Systems. In Proceedings of the *International Joint Conference on Artificial Intelligence* (IJCAI'07), 2007.

[YPH02] 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

[ZhL07] Zhang, Y., Lin, K., Hierarchical Management of Service Accountability in Service Oriented Architectures. In Proceedings *IEEE International Conference on Service-Oriented Computing and Applications*. IEEE Press, 2007.

8 Appendix A

In what follows we provide a summary of results of a pilot study conducted by Jo Stanley (CAM) for EEDA. The work presented here deals with very small businesses (VSBs) and draws on data collected at the Cambridgeshire survey by Jo Stanley.

1. Networks Supporting an Open Knowledge Space

No man is an island John Donne *Meditation XV11*

It is a logical extension of the social and business networks discussed by Granovetter[5] that a computerised *community* platform to facilitate transactions between members should be constructed.

However, the term 'computer networks' is a broad one.

We are just as much 'on a network' when we use the same file formats as others as when we communicate on a *physical* network peer-to-peer or, via a client machine with the office server. That is why consistent digital formats are vital to users, as is the interoperability of all software and hardware internal interfaces. This goal coupled with the tangential goal of preserving a firm's local autonomy requires a software system that sensitively reflects all the requirements of an OKS, satisfying access to and preservation of efficient private and public faces.

1.1 The Software Migration Funnel

All mankind is of one author and is one volume; when a man dies, one chapter is not torn out of the book, but translated.

Meditation XV11

The pattern of updates in Microsoft operating systems has become one of upgrading both systems and application software on a very frequent basis. On update, some backward compatibility is almost invariably lost. Gejlsbjerg and Stanley6 explored this 'migration funnel' effect in the case of Microsoft's network server operating systems shortly after the European Commission's *Microsoft Decision*. They found this:

> When a [Windows] Domain having a legacy system is first installed with Windows 2000 it is in *mixed mode*. This can be changed to *native mode but the change is not reversible*. In native mode, Windows NT 4.0 (earlier) Domain Controllers cannot participate in the Domain[7] and *therefore a server that is only interoperable with Windows NT 4.0* operating system ***can no longer be used***[8].

[5] Papers on this theme by Mark Granovetter include *Economic Activity and Social Structure: the Problem of Embeddedness*, (Am J Sociology at 811 1986), *Business Groups and Social Organisation* (in *Handbook of Economic Sociology,* ed Neil Smelser and Richard Swedberg [2005], (Princeton Univ. Press), *The Impact of Social Structures on Economic Outcomes*, Mark Granovetter, (Journal of Economic Perspectives, Vol 19 No 1, at P33-50).
[6] See Anna-Rosa Gejlsbjerg and Jo Stanley *Network Protocols: the Essential Intellectual Facility*, BILETA, [2005].
[7] See *Step-by-Step Guide to Managing Active Directory* (www.microsoft.com/windows 2000).
[8] See also para 170 of the EC Decision in the Microsoft case.

> The full advantage of any upgrade to Windows 2000 is only felt when workgroup servers in the Domain (other than the Domain Controllers) are Windows 2000-compatible[9]. Microsoft therefore recommends: 'To receive maximum benefit from Windows 2000 technologies and fully realize your migration-related goals, it is recommended that you switch your Windows 2000 Domains to native mode as soon as possible'[10].

The frequency of the upgrade cycle is increasing[11], and upgrading of business software is perceived in small firms as a *point of failure*. Instances of lack of backward compatibility were commonly reported in our interviews, with the more demanding and intensive users of software programs (notably editors, requiring precise change tracking, and manual writers requiring consistency in numbering sequences) caused the most inconvenience.

Increasingly globalisation confers greater power on the multi-national Corporations. Correspondingly corporations pledge to show 'Corporate Social Responsibility'. However, commentators point out that businesses are established to increase value for the owner across time, and that is their sole raison d'etre[12].

Milton Friedman regards any deviation by corporate servants (CEO *et al*) from returning maximum benefit to their principal as tantamount to usurping the government's mandate to tax, but with no corresponding mandate from their 'electorate' of shareholders. It is not the business of a CEO to pre-empt disposal of the shareholders just deserts before it reach them.

2. The 'Small Firms Computing Project

On a commissioned request from EEDA, Stanley *et al* conducted a survey of the use of ICT by very small businesses (VSBs). The broad conclusion was that the dominant issue with computerisation was a problem with interoperability (enlarged upon below).

2.1 Sampling Issues

We defined the group of VSBs as firms having up to ten workstations. (Defining in terms of machines used rather than employees)[13]. The study was also across sectors, to expose the more generic concerns in computerisation.

We lapsed occasionally from sampling VSBs into sampling SBs and chain store branches, for purposes of comparison. To illustrate, we (informally) interviewed a Church-based support group, operating from a church building. The group showed signs of market niche-ing applied to their clientele, certainly used computers, and had a web site. In short, it had some of the properties of a VSB, but we had to consider the 'corporate' affiliation with the Church of England as finally ruling it out. For broadly the same reasons we mostly omitted sampling the chain store branches which abound in the local shopping Mall though, interestingly some of these, in general conversation explained their use of *utility computing*[14], as did one

[9] Saved in:\Windows\System32\Kerberos.dll.
[10] Microsoft, *Deployment Planning Guide*, at Chapter 10: *Determining Domain Migration Strategy, Migration Goals*. Downloaded from http://www.microsoft.com/windows/techinfo/reskit/dpg/default.asp.
[11] Interview with Chief Sales Executive, Gray Matter, a large independent software vendor..
[12] Elaine Sternberg, *Just Business*.
[13] The reason being that a building firm or joinery with say, eleven workers and one office computer would probably not yield sufficient valuable insights into computerisation for the study. On our definition a firm might be larger than micro, but still fall within our bounds
[14] *Utility computing:* purchasing one's entire computer service just like any other utility, having a contract and agreement as to tariff with the provider.

small business in the telephone survey, and one VSB in an in-depth interview. That tiny cohort had not one bad word for the service they received.

Thus our collection of data was catholic. Since we were cross-sectorial, notions of physical location could be set aside for this study. Retailers were on thoroughfares, and professionals broadly tended to be in the blocks and streets behind the main thoroughfares, which led us to the loose categorisation of 'High Street', 'Back Street' businesses. But this was no criterion for sampling.

2.2 Core-ing out the Main Category

After open, then theoretical coding had been completed, it became clear that our core category was ***interoperability*** in all its guises, human to human, human to machine, machine to machine, and was most often encountered as its invert: lack of interoperability.

We then set about enriching our understanding of the properties of interoperability by reviewing incidents of it that were similar across all the data, to strengthen the category's status, and those that *differed*, in order to locate the category's properties.

We found it fruitful to explore the category *interoperability* in terms of the participant entities in interoperability relationships: interoperability amongst knowledge domains, media, humans, humans and machines, and machines.

Across these associations lie the cardinalities of the relationships: one-to-many, many-to-many, one-to-one.

We examined reports of updates and compatibility loss, in relation to media changes and program upgrades. We recorded justifications for program upgrades, and the users' perceptions of upgrade value for money and the drivers towards and away from upgrading software in businesses.

We located *switch points* at which businesses run out of road with their current software applications, and *at that point* are again prepared to review the offering of applications of similar functionality. This new look may be driven by the cost to upgrade and, with growing experience, the companies seem prepared to look at freeware as well as proprietary software, or consider changing brand allegiance.

We have explored the interstices between programs where there is a failure of interoperability, and observed that companies find roles for human mediators and assistants, and use assistant software to bridge the gap. We have noted that making good the mismatches between human skills and aptitudes and technological service provision is providing a living for some of the VSBs as facilitators. We have seen how readily small companies grasp middleware work-around programs, and wondered if these will weather subsequent upgrades in the software systems on either side of the middleware 'bridge', and noted that with such solutions the number of interfaces proliferates.

We looked (albeit sceptically) for support for the belief that creative powers in people are inversely related to their technological aptitudes, and found a little candidate evidence.

We have seen the effects of incompatibilities of time zones, language, (written and spoken) which change the work habits of firms in the UK collaborating with offshore workers, and the solutions they found for working with global associates.

2.3 Interoperability Amongst Knowledge Domains

To enable courier services to fill the missing link in the e-commerce order-to-delivery cycle, we found one consultant working on the 'fault line'[15] between RFID technologies and postal logistics.

There is a need to skill legal professionals in new technologies, (the target subject matter of much IPR and company law). We found an academic hybrid on this fault line, advising the developers of new University courses in ICT Law.

2.3.1 The Facilitation of the Man-Machine Interface

We found some evidence of technology aversion amongst authors and other very creative people. The sole proprietor of one VSB assists US artists (authors, photographers and painters) to achieve visibility in the UK. He solves their web site design problems, where the jargon of the design package and technical issues are hurdles the artist does not wish traverse in person.

Three ladies clothes retailers stated that they relied on a relative or partner to supply the technological know how that the business needed, indicating that their role was considered by them to be on the artistic side of the business, and marked their transferral of technical responsibility to another participant.

The hairdressers we encountered had no appointments software. This was, however, a general finding for personal services: the optician felt that negotiation was paramount in making appointments and associated *electronic* appointment systems with inflexibility.

A will-writer thoroughly enjoyed interviewing his clients, and steering them through the maze of legal traps by a sequence of counterfactual scenarios. However, he disliked the task of formalising the material using the expert legal software.

2.3.2 The Essence of Face-to-face Human Interoperability

A herbalist and clairvoyant, who initially worked from home, subsequently set up a High Street shop to gain more contact with the public. She believes that because a web site is broadcast (one-to-many) and not one-to-one then trust cannot be established. She also laid stress on the personality of the salesperson being 80% responsible for the sale.

She commented that word-of–mouth is unmatched as a marketing tool, stating that she found peers are intensely effective in disseminating commercial information amongst themselves, but that across generations, word-of–mouth can lose its power as the age difference between imparter and receiver increases.

This trader advocated a policy of selecting from the electronic toolkit, but was not persuaded that human imagination, flexibility and business personality could be translated into an electronic representation.

2.3.3 One Sender Many Receivers

The question of the visibility of a company to its market raises a plethora of interoperability issues. Traditional advertising is, on the whole, considered expensive by VSBs. Email was considered the most

[15] A phrase suggested to me by one interviewee.

important means of contact between the VSBs and their suppliers and customers after the phone. But here issues of cost and legality are met with. Interviewees pointed to legal blocks by postal and telephone preferences to accessing potential clients.

Mass email sends were described as 'flawed'. One interviewee recommended the services of a mail services specialist but pointed out that such services come at a cost. The parsimonious small businesses tended to solve their problems by work-around software solutions, frequently manual (leafleting).

A web designer accustomed to sending sophisticated content by email learned to use redundancy, guarding against failure at the receiver end by sending image, text, and a link to a web site. He also learned to test his web designs on four kinds of common browser, and now also tests on a text-only browser for the benefit of disabled recipients.

Standards in such as HTML, XHTML and PDF were cherished for their common currency. One heavy user of PDF documents expressed satisfaction with the version compatibility found when using the PDF writer/editor Those attempting to fix document appearance where that was of the essence – a web designer and a technical author – complained most of losses of formatting by failure of backward compatibility between versions of word processors, and glitches during conversion.

A publishers' editor reported failure of interoperability between two dominant proprietary workstations, even though the same word processor is same version was used. Also where referencing function was used in conjunction with auto-numbering on a machine produced by one proprietor, but on an application from another, files were corrupted.

2.3.4 One System Many Upgrades

A technical author reinforced the message of substantial lost time in her description of failures of current word processors to read past formats in tables and numbering, where continuity is essential to the manuals she produces. An editor estimated a 20% loss of productivity from interoperability faults.

The technical author also commented that if images were imported into word-processed documents, and the original discarded, there might be important loss of authenticity, flexible handling and ability to return to the source document. We have coined the term 'The Librarian's Predicament' due to a college librarian's description of the issues involved if early digital holdings are to be maintained, and the last electronic reader program for a particular document format disappears from the Library community[16].

2.3.5 When to Upgrade

Overwhelmingly the companies wished this to be a ***business*** decision, rather than a coerced technical distribution of the new upgrade. They saw the point of update as a likely to be a point of failure, with business time being lost. The CIO of a small ironmongesr[17] believed that upgrades frequently lost simple, familiar and useful functionality, and added unnecessary functionality. The will-writer remained happily with his initial WordPerfect-based software for 6 to 7 years, until it was no longer supported.

One Linux aficionado suggested that a switch to an Open Source operating system as 'no worse than a version change in proprietary software, and easier for start-ups'[18]. Few of the companies we spoke to showed any interest in changing from the prevailing Windows operating system with which they were familiar.

[16] A Cambridge college librarian first brought this to my attention.
[17] Using a total of 37 computers across all functional areas.
[18] Reported by Bob Dowling, Open Source expert at the Cambridge Computer Labs.

Overwhelmingly the companies saw software upgrades as unnecessary until the purchase of *new hardware* was contemplated or their *present systems broke*. One reported that he would need to be convinced of a 'technological leap' before he would be induced to upgrade. However, another interviewee with experience of the upgrading habits of much larger companies believed that small firms were more flexible in the matter of accepting the need to upgrade, if only because of the comparative scale of the operation.

But this picture changed when it came to specialist professional software. A kitchen designer believed that her firm updated their specialist design software every few months. An accountant keeps five versions of the Sage accountancy package, so as to be compatible with his various clients. The will-writer is obliged to use a particular the legal text, which is accompanied by its own software, thus the upgrade is coerced.

2.3.6 Parsimony, Simplicity and Interoperability

One interviewee started his company by exchanging shares in the company in return for an Open Source server software build. His method of parsimony was to consult experts – some friends - who enjoyed explaining and discussing solutions to technological problems purely as an intellectual exercise, and to 'call in favours' on the barter principle. He used and still uses word-of-mouth as the predominant marketing tool, with only an occasional boost from PR. The interviewee regards some, but by no means all, public relations services as *money* ill-spent, and the fruitless pursuit of contracts for work, notably with the major companies, as *time* ill-spent.

The owner of a company that answers questions online[19] has a policy of simplicity at the business's interfaces. Users communicate their questions using the ubiquitous SMS, which is reported to give him gives virtually no interoperability problems. Enquirers are charged a premium SMS flat rate however complex or easy the question. The response is expressed with very few exceptions, in a maximum of 153 characters. Search-workers only require an open connection and a browser in order to do the work, and are paid a flat rate per question[20]. The only complexity in this system is the backend software for routing the calls.

2.3.7 Complexity and Interoperability

The CIO of a small ironmonger's firm commented on the multiplicity of supplier interfaces. Some of the larger suppliers coerced e-commerce ordering on their own web sites, which left no record in the ironmonger's files; e-commerce systems differed from supplier to supplier. Sometimes a list of goods ordered on supplier sites was printed back to the trader out-of-original-order and incomprehensibly sequenced, which wasted time in checking and auditing. Middleware could help to solve the problem but was very expensive, and suppliers seemed unwilling to invest. Problems in e-commerce communication had to be solved over the phone.

2.3.8 Interoperability When Working with Offshore/Global Associates

The information searchers in our question-solving firm are drawn from a global community, and number several hundred. The pattern of work of each worker is occasional, but collectively the system must run

[19] Enquiries range from where a solicitor might find a café in a strange town near the courthouse to meet a client before court time, through how to cook a particular dish to quite complex scientific queries.
[20] The catholicity of questions accepted, the flat rates charged and paid to customer and searcher respectively all reflect the simple, untiered interfaces of the business.

24/7, therefore several hundred searchers are employed. Searchers, once recruited by passing an online test[21] drop in or out of circulation according to their other work load and preferences.

2.3.9 Online Working and the Human Computer Interface

Our will-writer felt that online wills would be a disaster for the client, since the choices of action offered were not accompanied by sufficient information, nor expert prompts to achieve the result that the client actually needed.

Conclusions

The cross-sectorial VSB study, together with one being conducted on the small eco-firms of Peterborough, will likely provide a spectrum of requirements, from the pragmatic, technical demands for interoperability needing to be addressed by DE design, to other requirements that could more appropriately shape criteria for the OKS and its tools.

Investment in eco-industry companies is deemed problematic for the following reasons.

Firstly there is marked information asymmetry between the entrepreneurs' and investors' knowledge of the new eco-technologies[22]. This problem is frequently solved in more mature high tech investment houses in Cambridge. These ensure that product assessment can be carried out, either by employing in-house subject specialists, or by hiring free-lance special knowledge consultants. A University town such as Cambridge is rich in specialist knowledge, but a new industry such as the eco-sectors represent, in an aspiring town cluster with no University at its heart has no such advantage. Therefore the pressure to use an efficient and far-reaching digital platform must be all the greater.

Secondly the eco-industries may require funding for certification. The eco-industries are regulation-led, and the required licenses are complex and expensive to gain.

The axis of investor/eco-firm is critical in these industries; reports identify financial barriers as the biggest obstacle faced by the firms to their survival or growth in 2006.

[21] The test is based on speed, ability to perform a quality search, and grammatical rendering of the response. It appears that around 25% of those who take the test do so simply as a challenge, similar to the Mensa test. Only 1-2% of those tested pass, and half of those are accepted, yet there is no shortage of applicants.
[22] Dee Ford and Garnsey, ifM Cambridge

9 Appendix B

Int. J. Technological Learning, Innovation and Development, Vol. X, No. Y, XXXX

Beyond interoperability to digital ecosystems: regional innovation and socio-economic development led by SMEs

Paolo Dini*, Gabriella Lombardo and Robin Mansell

London School of Economics and Political Science,
Houghton Street,
London WC2A 2AE, UK
E-mail: p.dini@lse.ac.uk E-mail: g.lombardo@lse.ac.uk
E-mail: r.e.mansell@lse.ac.uk
*Corresponding author

Amir Reza Razavi, Sotiris Moschoyiannis and Paul Krause

University of Surrey,
Guildford, Surrey, UK
E-mail: a.razavi@surrey.ac.uk
E-mail: s.moschoyiannis@surrey.ac.uk
E-mail: p.krause@surrey.ac.uk

Andrea Nicolai and Lorena Rivera León

T6, Via Genova 30, 00184 Rome, Italy
E-mail: a.nicolai@t-6.it E-mail: lorena.rivera-leon@t-6.it

Abstract: This paper shows the early results of new research on how Digital Ecosystems can promote new modes of sustainable e-business practices, for Small and Medium-Sized Enterprises (SMEs), using an open architecture for content sharing and Business-to-Business (B2B) interactions in the knowledge economy, and within a framework of open standards. The current e-business practices and technologies do not always encourage openness but instead tend to promote established models of proprietary e-business development based on centralised network and service infrastructure. Governments can promote open developments by supporting opportunities for new entry through supporting and augmenting a market environment for the productive coexistence of large and small companies in the B2B e-commerce domain.

Keywords: digital ecosystems; regional innovation; P2P networks; distributed transaction coordination; Small and Medium-Sized Enterprises; SMEs.

Reference to this paper should be made as follows: Dini, P., Lombardo, G., Mansell, R., Razavi, A., Moschoyiannis, S., Krause, P., Nicolai, A. and León, L.R. (XXXX) 'Beyond interoperability to digital ecosystems: regional innovation and socio-economic development led by SMEs', *Int. J. Technological Learning, Innovation and Development*, Vol. X, No. Y, pp.xx–xx.

Biographical notes: Dr. Paolo Dini is a Senior Research Fellow in the Department of Media and Communications, London School of Economics and Political Science (LSE), UK, and has been working on digital ecosystems research since 2002. Dr. Dini has a multidisciplinary and interdisciplinary research background in applied aerodynamics, physics, electronics, new media, computer science and social science.

Dr. Gabriella Lombardo is the Deputy Head of LSE Academic Partnerships, the unit that develops and supports higher education institutional alliances worldwide. She has a PhD in Economic History from LSE and co-directs an OPAALS workpackage on socio-economic models for digital ecosystems. Gabi's research background is on guilds in early modern Europe and the ability of craft communities to promote skill training rather than pursue a rent-seeking behaviour.

Dr. Robin Mansell is Professor of New Media and Head of the Department of Media and Communications, LSE, UK. She focuses on the political economy and sociology of innovation with respect to information and communication technologies.

Amir Reza Razavi is a Research Fellow and PhD student at the University of Surrey, UK. He received his BSc degree in Computer Software Engineering from Shiraz Azad University in 1996 (Iran), an MSc in Computer Software Engineering from Iranian University of Science and Technology in 1999 and a second MSc in Internet Computing from University of Surrey in 2003. During his career, he has taught database and transactional systems in Shiraz Azad University and Sepidan Azad University (1999–2001). His main interest has been long-running transactions in non-conventional environments. Much of his research has been directed towards analysis and design of distributed transactional systems and network infrastructure support for SMEs. He has contributed in three EU projects: ASPIC, DBE and OPAALS.

Dr. Sotiris Moschoyiannis is currently a Research Fellow in the Department of Computing, University of Surrey, UK. He received a BSc degree in Mathematics from the University of Patras, Greece, and an MSc in Information Systems and PhD in Computer Science from the University of Surrey, UK. His research work concerns the application of mathematical models in designing computer systems and the analysis of their properties prior to deployment. His research interests include semantic models of concurrency, formal languages and automata theory, service orchestration in distributed transactions and supporting P2P architectures. His work has been published in a number of journals and peer-reviewed conferences.

Paul Krause has over 20 years experience in research and application of advanced software engineering techniques. He has been Professor of Software Engineering at the University of Surrey, UK, since January 2001. Specific research contributions are in the areas of formal modelling of interacting software components, automated generation and execution of software test cases for highly concurrent systems and software quality prediction using Bayesian Networks. However, since moving to Surrey, his research interests have developed in the field of 'digital ecosystems' as a partner in the EU funded DBE and OPAALS projects. He is Computer Science Coordinator in the latter.

Mixing a humanistic background (Literature and Cinema) with a long professional practice in the area of Media Industry and ICTs, over the last 12 years, Andrea Nicolai has coordinated a substantial number of large international projects dealing with innovation in ICT use. Building on his experience in international negotiations, legal and contractual aspects of international agreements, and the management of large multicultural teams of

practitioners and researchers, he has been working on the development of innovation strategies in the area of media and online industries, SMEs and sustainable local economic development.

Lorena Rivera León consults for T6 as a Researcher in digital ecosystems, while working as a Research Assistant in the Department of Media and Communications, LSE. She is also responsible for the coordination of the Regions for Digital Ecosystems Network. Her research focuses on the socio-economic foundations of digital ecosystems and the impact assessment of their deployment. She has worked for the European Commission (DG INFSO) and the United Nations Industrial Development Organization. She received her MA jointly from the Department of Economics, University of British Columbia, Canada and the Université Pierre-Mendès-France, in 2006.

1 Introduction

This paper is being written in the middle of a visible transformation of the web from a distributed and interconnected information repository to a platform for social networking and content sharing and technologies that are collectively referred to as 'Web 2.0'. The paper considers this phenomenon as a symptom of the direction in which e-business practices and technologies are moving.

The paper argues how a decentralised and flexible socio-technical approach can support the formation and growth of global production and innovation networks. This is premised on a perspective that originated with the economic sociology field and with the embeddedness concept in economic decision-making (Granovetter, 1985), of which Web 2.0 could be seen as a recent manifestation. It is also premised on the need for more sophisticated technology that can support the distributed coordination of loosely coupled Business-to-Business (B2B) transactions in reconfigurable value networks, thereby preserving local autonomy and avoiding dependence on centralised transaction servers.

The digital ecosystems initiative is funded by the European Commission (EC) but we believe that both these aspects are centrally important in development contexts: (a) the local autonomy, because it is about social constructivist understandings of self-determination and (b) the independence, because, by empowering individual players, no matter how small, to play in the B2B market at the same level of multinational corporations, it achieves in the electronic B2B space a similar flattening and democratising effect the web has already reached in the content-sharing space.

Section 2 briefly looks at the Web 2.0 phenomenon and argues how traditional conceptualisations of the exchange of tangible goods and services can be greatly enriched in the knowledge economy by loosening the topology of the business interconnections, by extending the notion of 'value', and by opening up the interoperability standards. Section 3 summarises the economics of online B2B transactions, with attention to Small and Medium-Sized Enterprises (SMEs), to provide an assessment of the market context. This is followed by an in-depth discussion of the University of Surrey's distributed transaction model and the implications for the next generation of the underlying Peer-to-Peer (P2P) network architecture. Section 4 discusses the role of government in promoting an environment that supports innovation and growth in the new spaces of the knowledge economy and Section 6 concludes.

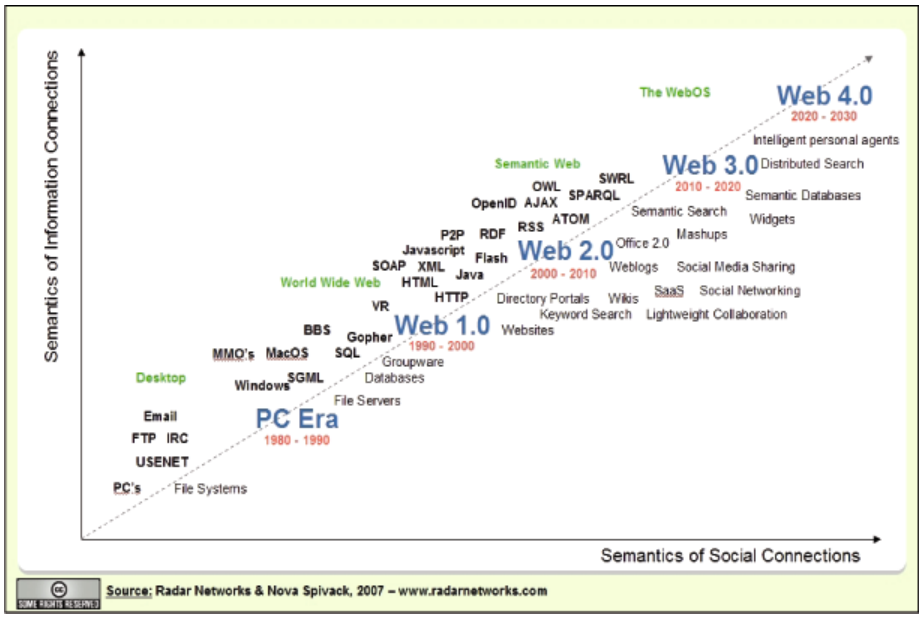
2 Knowledge, value, social ties and interoperability standards

Figure 1 shows how developments in the semantics of information and in the semantics of social networks appear to be progressing at an ever-increasing pace, mediated by sophisticated technologies. Such trends can be altered by unexpected developments, but in this paper we assume a progression along the lines depicted here.

This figure is interesting because it suggests that there will be a move from Web 2.0 to Web 3.0 to Web 4.0. Assuming this linear path of development, this move will happen in the USA, India, China and Brazil, and the like. The question is whether Europe wants to ride the wave(s) of innovation or follow in their wake. These waves of online innovation are influenced by a powerful connection between media technology and people, reminiscent of the success of text messaging in mobile phones. The Google advertising model was among the first examples to take advantage of this coupling and showed how free content sharing and social relations could be leveraged to drive sustainable business models in the traditional sense. But the figure also implies the emerging business models that are innovative in their reliance on the value generated by social relations *directly*. These are characterised as falling on the boundary between the Exchange Economy and the Gift Economy, such as Open Source, without which most of the web servers now in operation would not be running.

Figure 1 and the future of the web are more concerned about what is referred to as 'content sharing'. This tends to involve the interactions between individual users and the Business-to-Consumer (B2C) domain. As Information and Communication Technology (ICT) literacy rises in different countries, sectors and among people, we are likely to see a convergence between content sharing and the exchange economy, further strengthening the tendency of the service economy to rely on knowledge-intensive services mediated by ICTs.

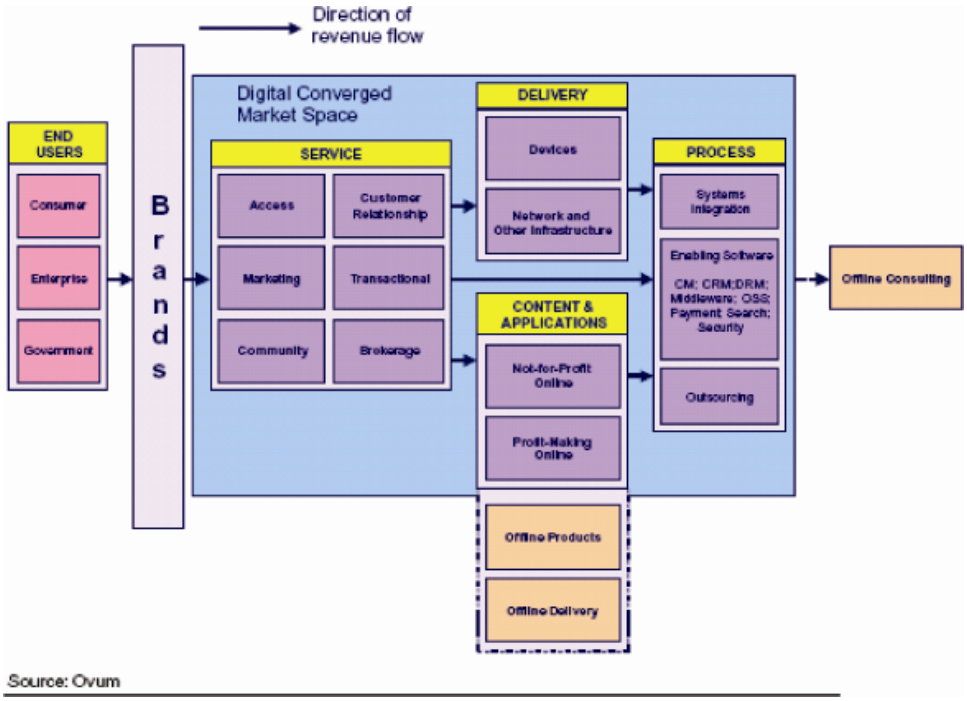
Figure 1 Constructive interference between information semantics and social semantics (see online version for colours)



This trend is strengthened by Open Source and similar phenomena (Creative Commons,[1] sharing of unused capital (Benkler, 2004), Community Currencies[2] and so on), which predate Web 2.0 by at least 10 years but are based on a similar blending of the social and the economic dimensions. Such trends are in the end connected to debates on intellectual property right protection, digital rights management and software patents that are part of conceptualising new value systems as the basis of new business models and as enablers of innovation in the emerging environments of the knowledge economy. Let us see how current understandings of the market for digital services measure up to these emerging trends.

Figure 2 shows a view of the digital market that is more concerned with the provision of services as a business model for the provider.[3] From this viewpoint, economies of scale and technological efficiency call for a well-integrated and interoperable platform, for which there is not much difference between citizens as consumers (B2C) and companies as purchasers (B2B). The figure also includes Government as one of three typologies of 'end-user'. The difficulty in such a framework is that it does not leave room for interactions that do not participate in economic production processes in the traditional sense, i.e. based on revenue. This model of economic exchange that was well suited to the material economy does not fit the knowledge economy so well. Notice also the fairly linear and synchronous character of the 'value chain' as depicted here.

Figure 2 Integrated B2C and B2B service provision framework (see online version for colours)



Source: Ovum

The first step in a 'constructive deconstruction' of this more traditional model is to distribute the source of value from the consumer and the value chain to the whole 'value network' (Allee, 2000):

> "It is no longer enough to think of a firm as a member of a closed system subject to uncontrollable outside shocks. It is part of a network that produces its own change. So, in analyzing the network all aspects of the network must be included: customers, suppliers, competitors, allies, regulators, complementors and any other network players whose presence in the network can influence value creation of the firm." (Peppard and Rylander, 2006)

The second step is to recognise the value of 'intangibles' in business transactions:

> "Interest in intangibles and corporate transparency has increased as business thinking evolves from bureaucratic and mechanistic models to more organic perspectives emerging from biology and living systems theory. … the basic pattern of organization for business is that of a network of tangible and intangibles exchanges. Tangible exchanges equate to flows of energy and matter [in living systems]. Intangible exchanges, such as knowledge, point to cognitive processes and intelligence. … [There] are serious attempts to develop new indexes, equations, measures, and analytical approaches for calculating knowledge assets and for understanding intangible value creation. All this adds up to a serious attack on traditional accounting and enterprise models that regard only revenue and physical assets as 'valuable', and that regard people as liabilities rather than important resources and investments." (Allee, 2002)

The third step is to come to terms with the fact that, as shown in the perhaps too simplistic and linear depiction of web evolution in Figure 1, in the content-sharing space of the web, it is difficult to set a price for the direct exchange of information that results from many different social interactions.

The most successful response so far was indirect revenue models, which predate Web 2.0, also known as 'related revenue models' (Mansell and Steinmueller, 2000, p.304). For example, the Google advertising model is successful precisely because it does not try to charge directly for the information being exchanged. The Google model creates a coupling with other services that are semantically about the information that is being viewed or shared for free. Revenue is generated from *micro*-payments between third parties that do not involve the end-user directly. Even giving due credit to this model, clearly the huge popularity of the new Web 2.0 phenomena such as Facebook, Flickr, and the like, points to the need to generalise our thinking about value systems towards the social space.

Though the strength of social interactions that the architecture of the web has enabled is contributing to transforming the content sharing landscape and to a healthy rate of innovation in the B2C space, the same cannot be said of the B2B space. Although, if we measure innovation by market size, this statement would appear erroneous, given the much greater volume of B2B relative to B2C, we mean innovation here in the sense of organisational and cultural change. Willingness and ability to change often lead to significant growth in volume in any sector. For example, Sassen (2006) argues that the ability of financial markets to embrace global electronic networks in the early 1990s led to their impressive growth since then, which she contrasts to the damping effect of the strict accountability constraints imposed on NGOs by funding bodies which prevent them from creating cross-country networks of shared knowledge and resources, and therefore also from benefiting from economies of scale. We see the barriers to organisational and cultural change in the B2B sector as arising mainly from the following three factors:

1 *Convergence leads to lock-in*. The challenge to support business interactions in a distributed and heterogeneous environment of differing syntactical service interfaces, semantic service description languages and messaging protocols is motivating the

larger infrastructure and ICT providers, at all layers of the stack, to make as large a share of the market as possible 'interoperable'. Where the market must be shared between a few players, industry standards are developed to promote the kind of convergence shown in Figure 2 (for example Bluetooth). When convergence is coupled too tightly to business models and market share, it leads to lock-in, even though lock-in is not a characteristic that emerges only in competitive markets, and slows innovation.

2 *Branding formalises lock-in in the public consciousness*. When premature lock-in occurs, which comes from the lag of what current ICT can support relative to the more dynamic demands of business interactions, this leads early entrants in the digital marketplace to protect market share by promoting proprietary standards and recognisable brands. Open standards can offer the consumer an alternative to these well-established brands, thereby creating new market entry opportunities for new players. Adopting open standards may be fostered by some form of government intervention (e.g. the involvement of United Nations Organisations in business modelling frameworks). In the content-sharing space, the Semantic Web initiative is about these two points.

3 *Intermediators constitute barriers to emergent and socially driven business activity*. B2B interactions may be strongly influenced and motivated by social phenomena such as small-world networks, family ties and geographical proximity. In some cases, the need to rely on third-party platforms can limit the formation, range and complexity of new value chains, business collaborations and business transactions. The ability to support a 'Web 2.0 for Business' could be beneficial to SMEs. In this segment of the market, government has the opportunity to foster opportunities for *many* open standards to emerge, which may provide a basis for socio-economic developments that are less constrained by the short-term business incentives of dominant market incumbents.

Though common standards can protect the ability of new entrants to compete with established brands, and government can help in this regard, the maximum possible theoretical interoperability level could be defined as making interoperability independent of standards altogether. Although this is an unrealistic goal in practical terms and undesirable from the view of socially constructed shared languages and technologies, it sets a useful limiting case for the more technical aspects of the interoperability debate. As discussed below, this was addressed first through the principle of loose coupling of the Service-Oriented Architecture (SOA) of ICT environments and more recently through Digital Business Ecosystems (DBE) research. DBE research has raised the awareness of the challenge that also the Semantic Web programme has posed. But the goal of interoperability at all levels, business interactions, content or anything else, remains worth pursuing. Our research challenges the means by which this goal may be gained by encouraging a wider interdisciplinary debate that encompasses social constructivist perspectives on the role of language as a medium of power relationships, and functionalist arguments and models, which are inspired by how biology has solved this problem.

This paper, therefore, is concerned with the barriers and opportunities for innovation and economic growth experienced by SMEs in the context of the potential available for 'constructive' and synchronous B2B interactions, besides those best characterised as

'defensive' moves by dominant players to promote asynchronous measures aimed at intellectual property protection. We leave this to a broader discussion of (open) knowledge and innovation networks (and of the extensive research literature in this area). In this paper, we look at some of the implications for the B2B environment of open architectural and design choices for an ICT infrastructure that supports e-business transactions specifically for SMEs. Though the paper starts to integrate a social science perspective with technical architecture considerations, it does not discuss or address the influence of biological solutions on the second.

The digital ecosystems initiative[4] offers an interesting view of new modes of economic organisation that aim to leverage loose and dynamic business networks in the online B2B interaction space in a way that is similar to the Web 2.0 content sharing environment. This view places a premium on the contribution of SMEs to economic growth by their flexibility and ability to form loose and dynamic business partnerships in response to changing market conditions. Before discussing the implications for the technology of such flexible business behaviour, we discuss some economic data on the online B2B space.

3 Basic economics of online B2B transaction environments

B2B online transactions or e-commerce have moved out of the early adoption phase and their long-term prospects are strong. In the EU-27, the percentage of enterprises' total turnover from e-commerce via internet doubled between 2004 and 2007, passing from 2.1% (2004) to 4.2% (2007) of total turnover (Eurostat, 2007). On average, 15% of EU enterprises received online orders in 2007, up from 9% in 2003. In addition, online sales by EU enterprises grew on average from 13% in 2003 to 27% in 2007 (United Nations Conference on Trade and Development (UNCTAD), 2007). In the USA, total e-commerce sales for 2007 were estimated at USD 136.4 billion representing a rise of 19% from 2006 (US Census Bureau, 2008). In 2005, B2B explained 92% of total e-commerce in the USA (US Census Bureau, 2007), while the volume of European B2B online trade raised to almost half of firms' purchases occurring online (European Commission, 2005).

Most B2B applications of e-commerce are in three areas corresponding to the different phases of the related business processes. Transaction preparation applications (pre-sale/pre-purchase phase) include advertising, catalogues and stock lists, price comparisons, information services/information about offers, and negotiation between seller and buyer. Transaction completion applications (sale/purchase phase) include ordering, billing and payment, finance and delivery. And transaction support applications (after sale/purchase phase) include information capture, information management, market analysis, market development, guarantee management, credit administration and handling returns (OECD, 2002; European Commission, 2007a).

However, e-business is more than just e-commerce. Although higher efficiency of business processes, internally and between trading partners in the value chain, continues to be one of the most important promises of e-business because of its direct impact on cost reductions, it is argued that innovative firms see e-business as an opportunity to deliver against key business objectives such as the delivery of high-quality goods and services, high-quality management, and marketing for improving customer service (European Commission, 2007a). E-business thus involves business processes in the

entire value chain, or value network. Firm size matters when talking about e-business. The continuing challenge is to promote the adoption of e-business by SMEs. According to the e-business Watch Survey in 2006 (European Commission, 2007b), there are roughly 50 SMEs engaged in e-business for every 100 large enterprises. Nordic SMEs are the most engaged but other differences among countries like France, Germany, Italy, Spain and the UK are not clear because of the uncertainty in the data.

For B2B transactions among SMEs, on average only about 11% of SMEs use software solutions or internet-based services for e-procurement. Moreover, there is a massive gap between the percentage of SMEs placing at least some orders online (53% of total) and those that use special software for this (only 11% of total). SMEs without special software place orders mainly through websites or extranets of suppliers (European Commission, 2007b). The result is a lack of digital back-office integration of procurement-related processes among European SMEs. Despite this, the e-business Watch survey also showed that 84% of small companies consider that e-business is an important feature of their business operations as compared with 81% of large enterprises that report that this is an important feature.

Estimating the share of the economy that e-business explains is difficult because the intensity, focus and the impact of e-business vary by business sector and by specific value chain in which an enterprise operates. Even so, according to the Eurostat Community Survey on ICT usage in enterprises (2007), in most European countries the volume of internet and other e-commerce transactions[5] is rising as a share of total turnover. In 2006, Denmark, the UK, Ireland and France were reported to have the highest shares in Europe, with 17% of enterprise total turnover coming from e-commerce in Denmark and the UK, and 16% of the total in Ireland and France.

For enterprises to have an incentive to adopt e-business and e-commerce strategies and tools, the benefits must be larger than the investment and maintenance costs of the tools. Public policy is directed to promoting e-business and e-commerce readiness and connectivity, but it also needs to promote more mature e-business strategies that integrate internal and external processes. Analysis has shown that technology neutrality is important in fostering these developments (OECD, 2004a). Research has shown that cutting the barriers to e-business adoption by promoting interoperable systems, the extension of network infrastructure, and related support services, offers a means of raising incentives for adoption. For B2B transactions across European countries (cross-border), it is also clear that reliable trust systems and an adequate legal and regulatory framework are needed.

The characteristics of B2B transactions suggest the need for an open infrastructure that is interoperable and allows enterprises to move freely in the market, thereby avoiding lock-in from 'principal agent problems,'[6] which may arise from market failures like information asymmetries, uncertainty and high risk. Policies can encourage these developments by creating incentives for new entrants in the market through fostering competition and investment in innovative technology infrastructures. The OECD Recommendations of the Council of Broadband Development in 2004 (OECD, 2004b) suggested that public financing to expand coverage of infrastructure networks to under-served groups (i.e. SMEs), with little negative effect on competitive market forces, is needed.

Although the above argument was hugely controversial at the time and was not supported by clear evidence, if adopting a perspective of 'global public goods' for the provision of ICTs and infrastructure networks as suggested during the United

Nations' World Summit on the Information Society (Geneva, December 2003), negative externalities arising from public action can be internalised (Binger, 2003). As explained by Binger (2003), "if the cost associated with a negative externality is effectively attributed to the responsible agent the externality is regarded as internalised". This approach suggests that governments have to assume responsibility for the negative effects that their own actions might generate and correct them accordingly (Kaul et al., 1999).

The next section presents the latest technological advances developed as part of the digital ecosystems initiative specifically to support dynamic B2B transactions between SMEs interacting through global production networks.

4 DBEs built on open-standards infrastructure and dynamic P2P architecture with local autonomy

The Digital Business Ecosystem (DBE) is unique because it offers a new approach to modelling business standards. Rather than assuming service offerings will converge to a common standard or promoting compliance with a centralised data model or architecture, the DBE supports an evolutionary approach that helps the formation of dynamic service chains match the flexibility of business partnerships, formed through web-enabled communication and Web 2.0 environments. The technical implications of such a vision are considerable, but they offer the potential for the digital ecosystems approach to deliver a disruptive innovation that could challenge the leading players in the market.

Current implementations of SOA for addressing B2B requirements tend to underestimate the negative impact of the new unique proprietary functional models. In many cases, efforts to ask competing businesses to use a unique data schema, or service model, proved unsuccessful, as competing standards cannot be enforced even when they were defined by a government or by a standards body. There are complex mechanisms that motivate ICT and business communities to adopt some standards that may become de facto standards. 'Good' standards do not always emerge from competition, and current implementations of SOA are unsatisfactory because they can be shown to violate important principles.

For example, the goal of Service-Oriented Computing (SOC) is to enable applications from different providers to be offered as services that can be used, composed and coordinated in a *loosely coupled* way (Papazoglou, 2003). This is the predominant computing paradigm in a DBE. In this paradigm, each participant in the B2B space does not need to expose the details of its internal workflows and business models as this model requires only that it present a service interface. The architectural approach of SOC is called SOA (Papazoglou et al., 2006) and it is applicable when many distributed applications are running on various technologies and when different platforms need to communicate with each other. SOA offers the promise of keeping full local autonomy for participants (i.e. loosely coupled such that their local state is invisible).

However, current web service protocols violate the SOA principle. This violation was justified as providing for safe recovery of aborted transactions (WS-BusinessActivity, 2004), but it is also the case that, until recently, the technology was unavailable to support a loosely coupled architecture. Digital ecosystems research shows that loose coupling is now within reach (Razavi et al., 2007b; Razavi et al., 2007c), although further research is needed to consolidate work in this area and to develop proper standards that will have a strong chance of being adopted. We argue that this chance will be

strengthened if the potential for spontaneous adoption is complemented by government intervention aimed at promoting this alternative market interaction mechanism.

Business transactions in a B2B context usually involve interactions between many partners, either service providers, or service consumers, or both. These interactions require partners to behave to some extent in a coordinated way – partners must follow an agreed protocol to execute transactions (Razavi et al., 2007a). A B2B transaction between SMEs in a digital ecosystem may involve simple usage of a web service or composition of several services from various service providers. A business transaction may be finished over a period of minutes, hours, or even days – thus the term long-lived or *long-running* transaction. Executing a long-running transaction corresponds to conducting a business and often comprises several *sub-transactions* that involve many underlying services.

Current implementations of transaction support in distributed e-business environments rely on a centralised transaction server. This is mainly because the orchestration and composition of separate services that cooperate in the delivery of a complex workflow is difficult to reach, so a centralised solution was the only possibility until recently. In contrast, digital ecosystems research is developing the concept, formal model, architecture and implementation for a *distributed* transaction management system. This system relies on Local Coordinators, one for each service, with a specification derived from a workflow model. The Local Coordinators act independently of one another to guarantee (local) consistency and recovery support across all the transactions in a complex workflow representing a business process.

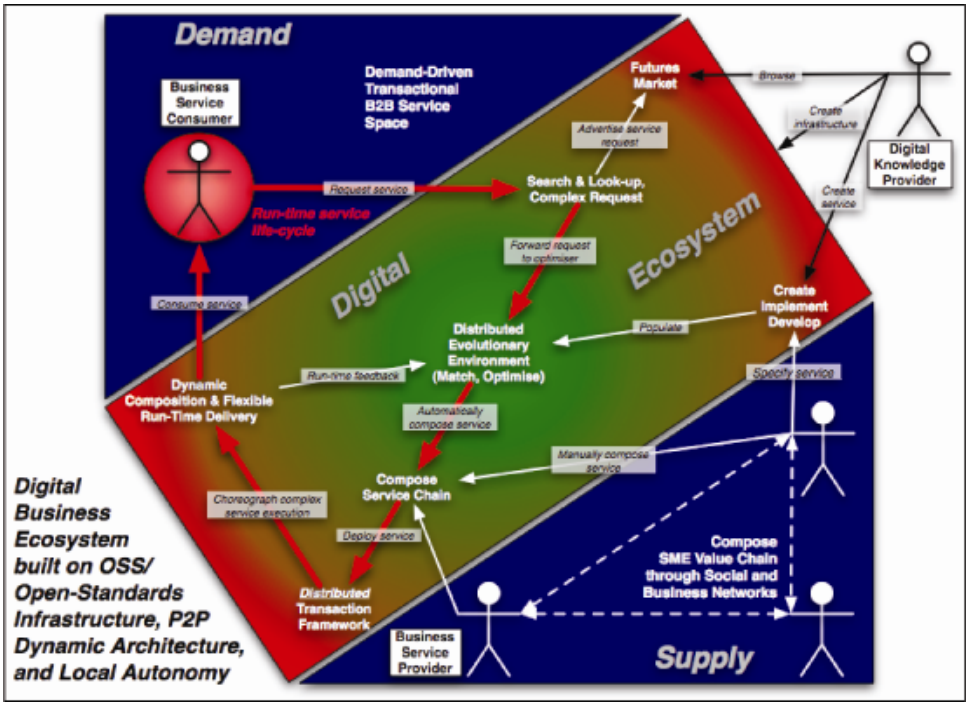
A Digital Ecosystem is able to ensure local autonomy and loose coupling because it relies on a distributed rather than on a centralised transaction manager. It is important to realise that in the digital ecosystems distributed transaction coordination model rollback recovery does *not* imply that the model is stateful.[7] Since there is no central server, the information about the previous state can *only* be stored on the servers of the participating SMEs who own this information. The Local Coordinators can only flag a failed step in the transaction and roll back to the previous stage (not state), where the information about the previous state is requested *again* from the same participating SMEs. So, the digital ecosystems distributed transaction coordination model is stateless and compatible with the SOA principles. A Digital Ecosystem integrates a distributed agent architecture with a distributed web service architecture, thereby reaching a true SOA. Figure 3 shows the complexity of a DBE, suggesting how the dynamic system of services in the Digital Ecosystem mirrors the dynamic system of social networks and business partnerships.

A business network enables networked firms to engage in distributed business transactions and to realise their business goals. The Digital Ecosystem provides support for B2B interactions between SMEs in a fully distributed way (no central point of control for transaction or network operations), and offers a consistent model for performing transactions. The model must be highly resistant to *fragmentation* – a situation where the network gets divided into smaller isolated networks – as this may inhibit collaborative business interactions. It must also be designed to cut the risk of failure at the transaction level. Transaction recovery must be supported by a procedure and must be helped by the underlying network that must support the choice of alternative paths/scenarios of transaction execution.

In the DBE, rather than having one service provider (like Google), there are thousands of them. In contrast to Google that has many regular servers around the world

(which are not transaction servers), the DBE supports B2B transactions while respecting SOA principles and the local autonomy of businesses engaged in transactions. The aim of DBE research is to develop a distributed coordination framework for long-lived transactions using a P2P architecture.

Figure 3 DBE as a mediating SOA for e-business (see online version for colours)



In the DBE architecture, not only does the network propagate traffic smartly, but also each 'virtual server' or 'virtual super-peer' is formed by a *collection* of servers cooperating to provide the functionality of one server in a reliable and flexible way that is not feasible when a centralised server is used. Servers in the P2P network are elected to become members of a virtual super-peer cluster by their availability to share resources and by their recent history of reliability. By the same token, they can be downgraded to regular peers if their reliability or availability decreases. In either case, it is foundational in the architecture that *the peer remains unaware that a change in its status has occurred*; similarly, the members of a virtual super-peer cannot belong to the same organisation. In this way the functional concerns for quality and reliability of service, which require a hierarchical network topology, were decoupled from any aggregation or centralisation of power or control, which an architecture that is merely functionally optimised could (wittingly or unwittingly) support. It is also relevant to point out that a centralised solution cannot easily compete with a decentralised or distributed solution on a cost basis. A centralised solution cannot be regionally customised and cannot adapt to peak-time traffic congestion by using localised propagation models – because it is centralised.

These discussions of network dynamics, topology, efficiency and costs mirror the challenge of understanding the interaction between organisational forms, decentralised

decision processes and information technology, and their effects on organisational performance, efficient operations and decision-making in different business settings (Galbraith, 1977; Mintzberg, 1978; Burgelman, 1988; Huber, 1990). There are other considerations relevant to this discussion, such as decentralised organisational configurations (Galbraith, 1994; Galbraith, 1995), intense use of new communication technologies (Bettis and Hitt, 1995; Fulk and DeSanctis, 1995) and intensified competition across industries (D'Aveni, 1994; Thomas, 1996). The main point, however, is the greater ease with which smaller companies can react to changes in the economy, and the role ICTs can play in easing change. Andersen and Segars (2001) show that computer networks can provide decentralised decision-makers with instantaneous access to relevant information, which will speed up the decision-making process, compared with formal approvals moving along several hierarchical layers of authority, where information overload can inhibit timely decisions (Mintzberg, 1992).

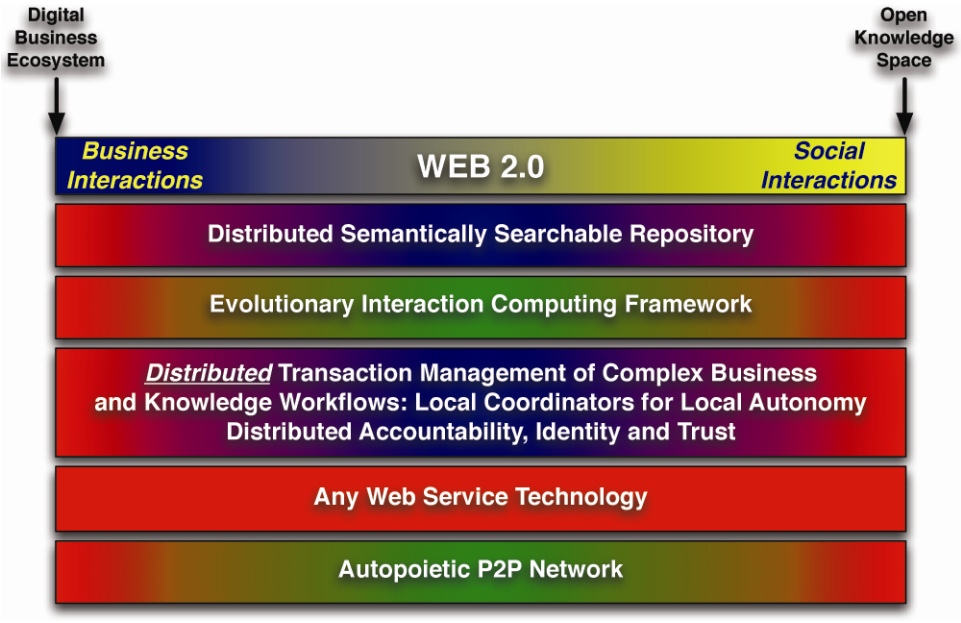
In this respect, SMEs do not suffer from centralised authority, and their flatter organisational structure offers flexibility for changing strategies in a short time, but the challenge is to enable them to interact through flexible and dynamic networks that can support business interactions. It is important to mention that any fragmentation of the ICT network can stop the organisational advantages of such dynamic, distributed and loosely organised global production networks. These are precisely the concerns addressed by digital ecosystems research. So, the flexibility and resilience of digital ecosystems viewed as system integrators of business, social and ICT networks would seem to enable the economy to benefit from the greater flexibility of SMEs without suffering from their greater vulnerability to market fluctuations and with lower transaction costs than experienced through 'state-of-the-art' centralised servers.

In short, in a DBE SMEs have the opportunity to do the following:

- maintain local autonomy
- avoid dependence on a centralised server (provided by a large enterprise which is then likely to promote its own model or standard)
- access a customised infrastructure that can cope with the dynamics of their local environment and adapt to changes quickly
- avoid network fragmentation and loss of partners because of hacker attacks, server failure, etc.
- benefit from the diversity of their environment and avoid the detrimental effects of unexpected peaks in traffic (a risk for centralised super-peers). The dynamic and unpredictable character of the SME environment means that traffic propagation cannot be predicted easily and static super-peers cannot cope with unpredictable peaks in traffic loads without a wasteful investment in resources to ensure against unpredictable peaks and congestion.

As a continuation of the DBE work, the Open Philosophies for Associative Autopoietic Digital Ecosystems (OPAALS) project is investigating an integrated architecture for business interactions and content sharing that is consistent with the social science arguments and technical principles outlined above. Figure 4 shows a high-level view of the open-standards architecture being developed and put into effect in the OPAALS project (www.opaals.org).

Figure 4 *Technology* vision of digital ecosystems being pursued in the OPAALS project (see online version for colours)



4.1 The role of government

Lessig (2006) argues that as economies change from manufacturing to service and now knowledge economies, legislative frameworks are unable to keep up with these developments and that there is a need for new forms of market intervention. For example, the anti-trust proceedings in Europe against Microsoft have taken 10 years before requiring this player to pay a fine of 340m Euro for its unfair market dominance. During this time the online media industry has undergone a revolution with Google, My Space, Facebook, Skype, e-Bay, You Tube, and others emerging and reaching dominant market positions in various 'sectors' of the web. Although it could be argued these developments signify healthy innovation in line with Schumpeterian forces of creative destruction in 'Fast-Forward' mode as large companies overtake each other in a frenzy of market expansion consistent with the emerging Knowledge Economy, it can also be argued that there are high risks for latecomers.

Historical evidence suggests that early entrants such as those referred above seek to claim as much territory or market share as possible through a variety of lock-in strategies, and then charge economic rents to the latecomers. Historically, this led to government intervention to foster minimum standards of fairness on competition in the marketplace. In interactive web developments and P2P networks, social networks are enabling trust and reputation mechanisms to support content-sharing spaces. But in the B2B space SMEs are disadvantaged by features of a digital divide because of their small scale and the absence of incentives for coordinated action that would enable them to reach efficiency and competitiveness.

Alternatives to the present system of economic rents arising from technical knowledge asymmetries created by complex patenting and licensing systems which were put in place since the beginning of the PC revolution are difficult to develop. This is partly because the norms and precedents on proprietary standards have become so familiar: we are so used to digital technologies being owned by others that we do not notice to what extent how we work, think and communicate is influenced by how these technologies are designed and regulated. The temptation is to regard the market arrangements for digital technologies and services as a given.

In digital ecosystems research, technology is regarded as an extension of our language, and language as the medium of social construction. In this context, the structure and development of B2B e-commerce and e-business markets are not predetermined. This opens the possibility of raising the question about the best economic model that will be most advantageous for SMEs. In short, digital ecosystems research asks whether it is proper for someone, anyone, to own our language, and, therefore, a key aspect of constructing knowledge economies. In so far as the answer is that open standards offer an alternative to the centralised server architectures that are available in the market, there is a case for government support for such an alternative.

Government support for open standards is an example of an indirect and light-touch way to strengthen the chances of success of new entrants in a market characterised by the dominance of large firms, thereby encouraging greater innovation potential. In the digital technology sector, there are some proprietary platforms through which SMEs must conduct all their online business transactions. Today in Europe this is the only entry point to the knowledge economy for millions of SMEs (20 million in the EU25). Clearly, there is a need to look at how this market environment operates and the extent to which SMEs are being disadvantaged.

5 Conclusion

The aim of the digital ecosystems approach is to develop structural, architectural and regulatory measures that can enable new entry in the face of the advantages enjoyed by dominant large firms that can exploit substantial economies of scale and scope in the B2B e-business marketplace. The digital ecosystems approach offers a means for protecting open innovation environments and for enhancing the potential for greater inclusion of SMEs in the emerging knowledge economy with the expected benefit that these firms will contribute more effectively to a dynamic marketplace and to sustainable economic growth. In the setting of intensified globalisation, we believe that this approach can support global production networks as a more inclusive and participatory industrial organisation model, creating positive effects for innovation and economic growth.

Acknowledgements

The authors gratefully acknowledge valuable feedback at various stages in the writing of this paper from Mr. Allan Mayo of the Department of Business Enterprise and Regulatory Reform of the UK Government. Mr. Francesco Nachira of the EC offered suggestions during the drafting of Figure 3, which were taken on board. Finally, the authors are very grateful to Dr. Olga Memedovic of the United Nations Industrial

Development Organisation who, as guest editor of this special issue, provided a great deal of encouragement and detailed feedback on all aspects of this paper. The research discussed in this paper was supported by the EC through the DBE project (FP6-507953) and the OPAALS project (FP6-034824).

References

Allee, V. (2000, July–August) 'Reconfiguring the value network', *Journal of Business Strategy*, Vol. 21, No. 4.

Allee, V. (2002) 'A value network approach for modeling and measuring intangibles', *The Transparent Enterprise: The Value of Intangibles Conference*, Madrid, 25–26 November.

Andersen, T.J. and Segars, A.H. (2001) 'The impact of IT on decision structure and firm performance: evidence from the textile and apparel industry', *Information & Management*, Vol. 39, No. 2, pp.85–100.

Benkler, Y. (2004) 'Sharing nicely: on shareable goods and the emergence of sharing as a modality of economic production', *Yale Law Journal*, Vol. 114, pp.273–358.

Bettis, R.A. and Hitt, M.A. (1995) 'The new competitive landscape', *Strategic Management Journal* (Special Issue: Technological Transformation and the New Competitive Landscape), Vol. 16, pp.7–19.

Binger, A. (2003) 'Global public goods and potential mechanisms for financing availability', Background paper prepared for *the Fifth Session of the Committee for Development Policy meeting*, 7–11 April 2003.

Burgelman, R.A. (1988) 'Strategy making as a social learning process: the case of internal corporate venturing', *Interfaces*, Vol. 18, No. 3, pp.74–85.

D'Aveni, R. (1994) *Hypercompetition*, The Free Press, New York.

European Commission (2005) *Information Society Benchmarking Report 2005*, eEurope 2005 Action Plan.

European Commission (2007a) *What is 'e-Business?'*, e-Business Watch, Luxembourg.

European Commission (2007b) *The European e-Business Report*, e-Business Watch, Luxembourg.

Eurostat (2007) *Community Survey on ICT Usage and E-commerce in Enterprises*, Brussels.

Fulk, J. and DeSanctis, G. (1995) 'Electronic communication and changing organizational forms', *Organization Science*, Vol. 6, No. 4, pp.337–349.

Galbraith, J.R. (1977) *Organization design*, Addison-Wesley.

Galbraith, J.R. (1994) *Competing with Flexible Lateral Organizations*, 2nd ed., Addison-Wesley.

Galbraith, J.R. (1995) *Designing Organizations: An Executive Briefing on Strategy, Structure, and Process*, Jossey-Bass.

Granovetter, M. (1985) 'Economic action and social structure: the problem of embeddedness', *American Journal of Sociology*, Vol. 91, No. 3, pp.481–510.

Huber, G.P. (1990) 'A theory of the effects of advanced information technologies on organizational design, intelligence, and decision making', *The Academy of Management Review*, Vol. 15, No. 1, pp.47–71.

Kaul, I., Grunberg, I. and Stern, M.A. (1999) *Global Public Goods – International Cooperation in the 21st Century*, Oxford University Press, Oxford, New York.

Lessig, L. (2006) *Code: Version 2.0*, Basic Books, New York.

Mansell, R. and Steinmueller, W.E. (2000) *Mobilizing the Information Society: Strategies for Growth and Opportunity*, Oxford University Press.

Mintzberg, H. (1978) 'Patterns in strategy formation', *Management Science*, Vol. 24, No. 9, pp.934–948.

Mintzberg, H. (1992) *Structure in Fives: Designing Effective Organizations*, Prentice Hall.

OECD (2002) *Electronic Commerce Business Impacts Project (EBIP)*, Graham Vickery and Vladimir López-Bassols, OECD/STI, Information Economy Unit.

OECD (2004a) *ICT, eBusiness and SMEs*, Working Party on the Information Economy, Paris.

OECD (2004b) *Recommendation of the Council on Broadband Development*, Statement by the OECD Information, Computer and Communications Policy Committee.

Papazoglou, M. (2003) 'Service-oriented computing: concepts, characteristics and directions', *Proceedings WISE'03*, IEEE, pp.3–12.

Papazoglou, M., Traverso, P., Dustdar, S., et al. (2006) 'Service-oriented computing roadmap', *Dagstuhl Seminar Proceedings 05462, Service-Orienrted Computing (SOC)*, pp.1–29.

Peppard, J. and Rylander, A. (2006) 'From value chain to value network: insights for mobile operators', *European Journal of Management*, Vol. 24, Nos. 2/3, pp.128–141.

Razavi, A., Moschoyiannis, S. and Krause, P. (2007a) 'A coordination model for distributed transactions in digital ecosystems', *IEEE Digital Ecosystems and Technologies* (*IEEE-DEST'07*).

Razavi, A., Moschoyiannis, S. and Krause, P. (2007b) *Preliminary architecture for P2P network focusing on hierarchical virtual super-peers, birth and growth models OPAALS D3.1*. Available online at: http://files.opaals.org/OPAALS/Year_1_Deliverables/WP03/

Razavi, A., Moschoyiannis, S. and Krause, P. (2007c) *Report on Formal Analysis of Autopoietic P2P Network, Together with Predictions of Performance*, OPAALS D3.2. Available online at: http://files.opaals.org/OPAALS/Year_1_Deliverables/WP03/

Sassen, S. (2006) 'Constructing the digital object of study', *Presentation given at the Claudio Ciborra Conference*, Department of Information Systems, LSE, February.

Thomas, L.G. (1996) 'Dynamic resourcefulness and the hypercompetitive shift', *Organization Science*, Vol. 7, No. 3, pp.221–242.

UNCTAD (2007) *Information Economy Report 2007-2008. Science and technology for development: The new paradigm of ICT,* United Nations, 35pp.

US Census Bureau (2007) *E-Stats: E-commerce 2005*, US Department of Commerce, Washington DC.

US Census Bureau (2008) *US Census Bureau News – Quarterly retail E-commerce sales 4th Quarter 2007*, US Department of Commerce, Washington DC.

WS-BusinessActivity (2004) Available online at: http://schemas.xmlsoap.org/ws/2004/10/wsba/

Notes

1 Available online at: http://creativecommons.org/

2 Available online at: www.openmoney.org

3 From Ovum study for the UK DTI (2005). Available online at: www.ovum.com

4 Available online at: www.digital-ecosystems.org

5 Including proprietary Electronic Data Interchange (EDI).

6 The principal agent problem in economics arises under conditions of incomplete and asymmetric information when a 'principal' hires an 'agent'. Various mechanisms may be used to try to align the interests of the agent with those of the principal, causing the agent to be 'locked in' to the principal.

7 In computer science, 'stateless' refers to a system or protocol that does not keep a persistent state between transactions. A stateless server is a server that treats each request as an independent transaction that is unrelated to any previous request. 'Stateful' is the opposite of stateless. Available online at: http://en.wikipedia.org/wiki/Stateless_server