Digital Business Ecosystem

Contract n^{o} 507953

WP12: Optimisation

Del 12.2: Optimisation approaches and software optimising the cost function in a static state space

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

Contract number: 507953
Project Acronym: DBE
Title: Digital Business Ecosystem

Deliverable number: D12.2
Due date: 30/12/2004
Delivery Date: 3/12/2004

Short Description: First the necessary requirements are discussed for any formal optimisation method to work. The structure of the system that is needed for such scientifically oriented methods to work is presented. Then a simple optimal recommender is presented. Finally, the neural model briefly mentioned in the previous report is carefully discussed.
Keywords: Recommender, Global optimisation

Partners owing: CVSSP, University of Surrey
Partners contributed: CVSSP, University of Surrey
Made available to: DBE Consortium

Versioning

Version	Date	Author, Organisation
1	3/12/2004	Petrou, Giannoutakis & Thilakawardana
2	13/12/2004	CVSSP, SEPS, Univ of Surrey, UK

Quality check
1st Internal Reviewer: Paolo Dini, LSE
2nd Internal Reviewer: Philippe DeWilde, ICL
3rd Internal Reviewer: Maria Chli, ICL

Contents

D12.2 Optimisation for a single customer and Methodology for analysing a Digital Business Ecosystem

M Petrou D Thilakawardana and K Giannoutakis
CVSSP, School of Electronics and Physical Sciences,
University of Surrey,
Guildford, GU2 7XH, UK

December 8, 2004

Abstract

A digital business ecosystem is treated here as a closed system of SMEs which interact with each other through buyer-seller relationships. As we are interested in using scientific methods to study such a system, we present first the way the system representation has to be standardised. We then present how two methods only briefly mentioned in the previous report (D12.1) have to be converted into optimisation algorithms, starting from the simplest one, namely the recommender. Then we present a model for the fitness of a company or the fitness of a product[1] in the ecosystem. It consists of sets of differential equations applied to a fully connected network of companies.

Key words: Recommender, optimisation

1 Introduction

The digital business ecosystem envisaged in this project is assumed to be a collection of companies that come together in cyberspace, to buy and sell their products and services. A customer is expected to be able to enter the ecosystem and search for the best product that meets her/his requirements. The role of the recommender is to identify which this product is, in an optimal way. This part of the project is the most pragmatic one, that may immediately become part of a usable infrastructure service.

However, there is also another aspect of the project, concerned with the idea that this system of companies may be treated as an ecosystem. An ecosystem is assumed to be closed, or nearly closed, and this offers the chance to model it and observe its behaviour under varying internal (and possibly external, in the case of the semi-closed system) conditions. This part of the project is oriented towards regional analysis studies, as it concerns planners and policy makers rather than customers and sellers. In a sense, we distinguish two scales in which the DBE will operate: The micro-scale, where byers and sellers interact in the most “selfish” way, each trying to get the best deal for themselves, having only a self-centred view of the world, and a macro-scale, where somebody,

[1]In this report we shall use interchangeably the terms “product” and “service”, as both constitute commodities that are either on offer or in demand.

possibly external to the ecosystem, wants to play “God” with it and ask questions of the form “what will happen if....”. The former micro-scale aspect requires the use of a greedy algorithm, where the user’s demand for a product is met in the best possible way, irrespective of the satisfaction of the requirements of other users. Such an algorithm is called “object-centred” because it tries to satisfy a particular set of constraints set by an individual and ignores the state of satisfaction of others. In section 2 we present such a simple algorithm, which is called the recommender, because for a given customer query it identifies the best product in the database to satisfy him/her.

In the macro-scale state, the system is treated as a collection of individuals that interact in certain ways with each other, inhibitory as well as excitatory. A closed system of this form must have some states of existence, and the idea is to study them through models we create. In section 3 of this report we present in detail the way such a system may be modelled using a biologically inspired neural network.

Finally, in section 4 we present our conclusions and plans for the future, and a revised version of the work schedule, on the basis of the understanding we have gained by working on the problem for the past nine months.

2 The recommender

In order to use scientific methodology to solve real world problems, we need an intermediate step which bridges the gap of the diversity and individuality encountered in the real world, with the strict mathematical representation required by the scientific approaches. It is true that for a business ecosystem to attract members, some relaxed rules of representation have to be adopted, where each company is allowed to represent its product or service in the way they consider to be best. However, in order to use an optimisation algorithm borrowed from sciences, a standardised way is required for presenting each offered service or product. We assume that colleagues working on the ontologies and business language aspects of the project will provide the bridge to this gap. So, we assume that the algorithms we develop sit at a top level where all diversities of representation and individualistic aspects of the various members of the DBE have been smoothed out and are hiding behind a layer of standardisation. We assume that each product on offer is stored in the database in a particular way, where products and services of a certain type are all identifiable by the same name, and are all described by the same set of attributes, that take values in standard units. For example, delivery time, may always be described in days, and weight of a product may always be described in kilogrammes, etc.

Once this standardisation has taken place, it is not difficult to imagine that the customer who logs on the system in order to search for a good to buy for him/her, is required to fill an on line questionnaire that describes the product they want in the same standardised way. In order to demonstrate how this might work, we constructed a database of 100 companies, each selling and wanting to buy any number from 2 to 5 products or services. The customer then who logs on the system, is first shown the list of products/services that are on sale, and is asked to type which of these products or services is interested in buying. She then has to type the name of the product as it

appeared in the list. The system replies by telling her that this product is described by "x" number of attributes, and asks her to specify the range of values she will be happy with, for each one of these attributes in turn, specifying also the units that should be used in quantifying each attribute. For example, if the product chosen were a holiday, one of the attributes might be duration, in days. So far, our approach has assumed that all attributes are of numerical nature. For example, we have not catered for attributes like country in which the holiday is desired. This, however, is not difficult to incorporate. Linguistic attributes may be checked for matching and the result presented in a binary form which may be treated as a numerical difference.

Carrying on with the description of the recommender, after the customer has filled in all the attributes of the product they require, she is asked to specify whether she wishes to give a relative importance to the attributes of the product she just described. For example, price may be of most relevance to the required product, or delivery time may be. The customer is asked to score in an absolute scale from 1 to 5 (with 5 being the most important) the importance they give to meeting each one of their requirements. If the customer declines to do that, the system uses default weights of relative significance for each attribute. At the moment our program has a set of arbitrarily chosen default weights. However, in an operational mode, one is expected to use public studies of market questionnaires where for each product people rated what they value most. Alternatively, as the system is operational, scores of previous customers may be used to build up a learned list of relative importance the customers assign to the attributes of each product.

Once this interactive session with the customer is over, the recommender proceeds as follows: First it normalises the scores of relative importance so that they are all converted to weights that sum up to 1. Then it recalls every product in the database with the same name as the requested product, and computes its similarity with the required product.

In this work we use the same notation we defined in the previous report, D12.1. We repeat it here in table 1 for convenience, although not all terms are of relevance here.

Let us assume that a customer wishes to buy product d. In order for the customer to decide which company to contact, the recommender asks about the attributes x_l^d the customer wants product d to have. Let us assume that these are: $x_l^d = \lambda_l$, where $l = 1, 2, ..., L_d$. There may be several companies C_j in the ecosystem that provide similar products $s_{k_p}^j$. The mismatch of any such product with d may be computed as

$$V_{k_pj} \equiv \sum_{l=0}^{L_d} w_l V_{lk_pj} \tag{1}$$

where

$$V_{lk_pj} \equiv \frac{|x_l^{s;k_pj} - x_l^d|}{x_l^d} \tag{2}$$

V_{lk_pj} in equation (2) is the mismatch value of attribute l, between the required product by the customer and product k_p supplied by company j. This is necessary, because attributes are measured in different units and we must use dimensionless numbers. The weights w_l are the scores assigned by the user to each attribute, normalised by the sum of all scores, so that they sum up to 1. Alternatively, default values may be used which should also be expressed as numbers that sum up to 1. The recommender will then identify product $s_{k0}^{j_0}$ for which the mismatch value V_{k_pj} is minimal. At

the end, the recommender will rank all products or services, found in the database, with the same name as the requested product or service, according to increasing value of dissimilarity. It will also list all the attributes of each product, and the company that offers it.

The characteristics of this system are:

1. It can easily incorporate learning of which attributes of each product the customers who log-on give most significance. All one has to do is to store and accumulate the scores the customers give to the various products they request, and at the end average these scores and use them as the default weights next time a customer logs in and does not want to bother and score the significance of the attributes herself.

2. It can be easily modified to incorporate linguistic attributes, treating their match or mismatch as a binary number. For example, if the product on demand is a holiday, and if the customer wants a holiday in Florida, a holiday on offer which is not in Florida, will contribute with 1 to the right-hand-side of equation (1), while a holiday on offer in Florida, will contribute with 0. These values are in the same range as the contributions from matching the other attributes between the requested product and the product on offer.

3. Attributes in the form of dates may also be easily compared. For example, if the customer wants a holiday to start on July 1, 2005, but the holiday on offer starts on June 28, 2005, the difference of these two dates in days may be used to contribute to the mismatch measure. The customer may be asked to specify the shift she is prepared to accept in deviating away from her preferred day of travel, and this may be used to normalise the discrepancy in dates so that they vary in the range $[0, 1]$. For example, if the customer is prepared to tolerate $+2$ days -10 days shift from her preferred date, the above example will lead to a contribution to the overall mismatch value of $3/10 = 0.3$.

4. The recommender uses a normalised simple measure of dissimilarity between the required product and that on offer. This measure is a metric. It can be used in subsequent studies to measure the compatibility between any two products on offer and in demand in the database, when we try to study the states of the ecosystem by modelling the relationships between companies.

3 A biologically inspired self-organising neural network

The purpose of this component of the work is to develop the methodology that will allow one to study the ecosystem under various conditions. In particular, we would like to answer the following question: "Under the assumption that the ecosystem is closed, and static, ie no external influences, which of the companies in it are most likely to prosper and survive, and which are most likely to be suppressed?" This is a situation of competitive existence, where each unit receives excitatory and inhibitory signals from all other units in the system. Such systems exist in biology, and their states are known to oscillate between extremes, rather than to converge to a single steady state. The reason of these oscillations is the asymmetry with which they influence each other. We shall

C	:	The set of all companies in the ecosystem
N	:	The total number of companies in the ecosystem
M	:	The total number of non-empty subsets of C
C_i	:	A particular company in the ecosystem (ie $C_i \in C$)
$D(C_i)$	:	The set of all services/products C_i requires
$S(C_i)$	:	The set of all services/products C_i may supply
d_k^i	:	A particular demand company C_i has (ie $d_k^i \in D(C_i)$)
s_k^i	:	A particular service/product C_i may supply (ie $s_k^i \in S(C_i)$)
$x_l^{d;ki}$	:	A particular characteristic of product/service d_k^i of company C_i needs. Examples: $l = 0$ indicates amount, ie by how much the capacity of the supplier will be reduced if this service is provided $l = 1$ indicates price C_i is willing to pay $l = 2$ indicates time C_i is willing to wait $l = 3$ indicates quality of service C_i requires etc
$x_l^{s;ki}$	:	A particular characteristic of product/service s_k^i company C_i offers. Examples: $l = 0$ indicates the total capacity C_i has to supply service/product s_k^i $l = 1$ indicates price C_i expects to receive $l = 2$ indicates time C_i takes to deliver $l = 3$ indicates quality of service C_i offers etc
$R_2(C_i, C_j)$	:	Direct relationship company C_i has with company C_j
$T_2(C_i, C_j)$	:	Indirect relationship company C_i has with company C_j (via third parties)
R_{ij}	:	Abbreviation of $R_2(C_i, C_j)$
T_{ij}	:	Abbreviation of $T_2(C_i, C_j)$
$R_3(C_i, C_j, C_k)$	:	Direct relationship companies C_i, C_j and C_k have with each other
$T_3(C_i, C_j, C_k)$	:	Indirect relationship companies C_i, C_j and C_k have with each other
$R_n(C_{i_1}, C_{i_2}, \ldots, C_{i_n})$	:	Direct relationship companies $C_i, \ldots, C_{i_n}$ have with each other
$T_n(C_{i_1}, C_{i_2}, \ldots, C_{i_n})$	:	Indirect relationship companies $C_i, \ldots, C_{i_n}$ have with each other
P_t	:	The set of all sub-sets of C consisting of t distinct companies
$R_t = R_t(C_{i_1}, C_{i_2}, \ldots, C_{i_t})$	:	Abbreviation of $R_t(C_{i_1}, C_{i_2}, \ldots, C_{i_t})$ when there is no danger not to know which t companies it is referring to
$T_t = T_t(C_{i_1}, C_{i_2}, \ldots, C_{i_t})$	:	Abbreviation of $T_t(C_{i_1}, C_{i_2}, \ldots, C_{i_t})$ when there is no danger not to know which t companies it is referring to
$C_i \leftarrow C_j$	:	Company C_i buys from company C_j
$p(C_i, C_j)$	:	Probability that company C_i buys from company C_j
N_0	:	Set of indices $\{1, 2, \ldots, N\}$
N_i	:	Set of indices $\{1, 2, \ldots, i-1, i+1, \ldots, N\}$

Table 1: Summary of notation

base our modelling on a neural network that has been proposed recently to model the visual cortex. In this model, every neuron has associated with it a so called membrane potential. The membrane potential changes with time: the stronger it is, the faster it decays. So this membrane potential obeys a differential equation. For example, we know that whatever the potential of a neuron is, in lack of any external stimulus, it will decay with time exponentially:

$$\frac{dy}{dt} = -\tau y \Rightarrow y = y_0 e^{-\tau t} \tag{3}$$

where τ is the time constant of the system, and y_0 is the value of y for the boundary condition $t = 0$. This is a pretty standard equation for decaying systems in Physics. Li in [1, 3] associated with each neuron two cells: one excitatory, and one inhibitory. Each cell obeys an equation like equation (3) with the addition of some extra terms on the right-hand-side in order to model the mutual influences between neurons as well as the external influences to the system. At the end, the fate of a neuron on whether it will be salient in the system within which it operates, or silent, was determined by examining whether the excitatory or the inhibitory potential turned out to be stronger. This way, she was able to take as input the noisy and contradictory signals reaching the neurons from many sensor units, the inhibitory and excitatory signals exchanged between the neurons, and show that after the system was left to relax, the neurons that were prevailing because their membrane potentials were strongly positive, agreed with the observed behaviour of the human visual system, in many aspects. In the following subsection we shall present a model for the ecosystem inspired from these ideas.

3.1 Modelling the competitive co-existence of companies

Let us assume that each company C_i has with it associated a positive variable, y_i, which measures how well the company does and how strong it is, and let us call it the **fitness** variable. This is an abstract quantity with no units associated with it, and it should not be confused with economic indicators like cash flow, volume of transactions etc. If the company is left on its own, in isolation, the value of y_i will decay according to equation (3) because a company of course cannot exist in isolation and with no interactions with other companies/customers. For simplicity, let us assume that for all companies, the decaying constant τ_y is the same.

First we shall produce a simple model for variable y_i. The differential equation obeyed by y_i will have to model the following effects, yielding extra terms that have to be included on the right-hand-side of equation (3):

- The stronger a company is, the more strong it is likely to become. This is a self-excitation term, of the form $J_0 g_y(y_i)$. Self-excitation constant J_0 again is assumed to be the same for all companies. Function $g_y(y_i)$ is a sigmoid function: effects in real life are only linear over a certain scale. They saturate and the benefit we receive by changing the independent variable y_i levels off. On the other hand, before this positive feedback in the strength is triggered, a so called "critical mass" of strength y_i has to be reached. So, function $g_y(y_i)$ may look something like the function shown in figure 1, and it may be modelled as:

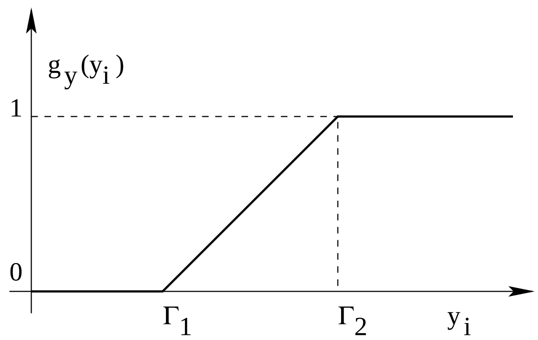

Figure 1: The activation function of the fitness variable

$$g_y(y_i) = \begin{cases} 0 & \text{if } y_i < \Gamma_1, \\ (y_i - \Gamma_1) & \text{if } \Gamma_1 \le y_i \le \Gamma_2 \\ 1 & \text{if } y_i > \Gamma_2 \end{cases} \tag{4}$$

where $[\Gamma_1, \Gamma_2]$ is the range of linearity of the positive gain function.

- A term that models all excitatory signals the company receives from all other companies. First we have to quantify the excitatory signal a company C_i receives from another company C_j. A company will stimulate another company if they demand products that match those the other company sells. So, first we need to quantify the dissimilarity between product s^i_k company C_i sells and product d^j_m company C_j demands. To do that we use a generalisation of equation (1):

$$V_{kimj} \equiv \sum_{l=0}^{L_{d;mj}} w_l V_{lkimj} \tag{5}$$

where $L_{d;mj}$ is the number of attributes used to describe product m wanted by company C_j, and

$$V_{lkimj} \equiv \frac{|x_l^{s;ki} - x_l^{d;mj}|}{x_l^{d;mj}} \tag{6}$$

V_{kimj} measures the dissimilarity between product s^i_k company C_i sells and product d^j_m company C_j demands. If this number is below a certain threshold T_1, we may assume that the products match. The more such products match, the more likely it is that company C_i will receive positive stimulation from company C_j. Let us say, therefore, that we count all pairs of products $\{(s^i_k, d^j_m) | s^i_k \in S(C_i)$ and $d^j_m \in D(C_j)\}$ for which $V_{kimj} \le T_1$ and find them to be E_{ij}. We may define then the excitatory signal C_i receives from C_j as

$$J_{ij} \equiv 1 - e^{-E_{ij}} \tag{7}$$

Note that the higher E_{ij} is, the more J_{ij} will tend to 1, while when $E_{ij} = 0$, ie when no products match, $J_{ij} = 0$ too. Also note that the excitatory signal company C_j sends to C_i is not the same as the excitatory signal C_i sends to C_j. In other words $E_{ij} \neq E_{ji}$, as E_{ji} will count the pairs of products that are less dissimilar than T_1 from the set $\{(s^j_k, d^i_m) | s^j_k \in S(C_j)$ and $d^i_m \in D(C_i)\}$.

In addition, we must also realise that a company C_j will stimulate company C_i only if C_j is healthy and strong itself. So, the excitatory signal J_{ij} must be modulated by $g_y(y_j)$ to account

for that. A company that is very weak will probably not create much volume of trading. Finally, we must sum up all such positive influences C_i receives from all other companies in the ecosystem. So, the term we must add on the right-hand-side of (3) should be:

$$\sum_{j \in C, j \neq i} J_{ij} g_y(y_j) \tag{8}$$

- A term that models all inhibitory signals the company receives from all other companies. First we have to quantify the inhibitory signal a company C_i receives from another company C_j. A company will inhibit another company if both companies sell similar or identical products. So, first we need to quantify the dissimilarity between product s_k^i company C_i sells and product s_m^j company C_j sells. To do that we use a generalisation of equation (1):

$$U_{kimj} \equiv \sum_{l=0}^{L_{s;mj}} w_l U_{lkimj} \tag{9}$$

where $L_{s;mj}$ is the number of attributes used to describe product m sold by company C_j, and

$$U_{lkimj} \equiv \frac{|x_l^{s;ki} - x_l^{s;mj}|}{x_l^{s;ki}} \tag{10}$$

U_{kimj} measures the dissimilarity between product s_k^i company C_i sells and product s_m^j company C_j also sells. If this number is below a certain threshold T_2, we may assume that the products match. The more such products match, the more likely it is that company C_i will receive inhibitory signals from company C_j. Let us say, therefore, that we count all pairs of products $\{(s_k^i, s_m^j) | s_k^i \in S(C_i) \text{ and } s_m^j \in S(C_j)\}$ for which $U_{kimj} \leq T_2$ and find them to be F_{ij}. We may define then the inhibitory signal C_i receives from C_j as

$$W_{ij} \equiv 1 - e^{-F_{ij}} \tag{11}$$

Note that the higher F_{ij} is, the more W_{ij} will tend to 1, while as $F_{ij} \to 0$, $W_{ij} \to 0$ too. We note that $F_{ij} = F_{ji}$, as F_{ji} will count the pairs of products that are less dissimilar than T_2 from the set $\{(s_k^i, s_m^j) | s_k^i \in S(C_j) \text{ and } s_m^j \in S(C_i)\}$.

In addition, we must also realise that a company C_j will inhibit company C_i only if C_j is healthy and strong itself. So, the inhibitory signal W_{ij} must be modulated by $g_y(y_j)$ to account for that. A company that is very weak will probably not create much volume of trading. Finally, we must sum up all such negative influences C_i receives from all other companies in the ecosystem. So, the term we must add on the right-hand-side of (3) should be:

$$-\sum_{j \in C, j \neq i} W_{ij} g_y(y_j) \tag{12}$$

- We may also include a term which may be external input to the company, like total volume of transactions originating outside the DBE, or something like that, properly scaled to be a dimensionless number. Let us call this I_i.

- Finally, we may add a term that expresses the background input, eg the general economic climate, and it is the same for all companies in the ecosystem. Let us call it I_0.

If we put all the above together, we come up with the following differential equation that has to be obeyed by the fitness variable of company C_i:

$$\frac{dy_i}{dt} = -\tau_y y_i + J_0 g_y(y_i) + \sum_{j \in C, j \neq i} J_{ij} g_y(y_j) - \sum_{j \in C, j \neq i} W_{ij} g_y(y_j) + I_i + I_0 \tag{13}$$

This is a set of coupled differential equations concerning all companies in the ecosystem. If we solve it, we may be able to see the combination of values of the fitness variables that will tell us which companies will dominate the ecosystem.

3.2 Modelling the competitive co-existence of services

The model we presented in the previous section deals with each company as a whole. Here we are going to refine it further, by dealing with each product or service offered by each company and associating with it a **fitness** variable y_{ik}. Variable y_{ik} will obey for each product on offer from each company a differential equation which is a refined version of (13):

$$\frac{dy_{ik}}{dt} = -\tau_y y_{ik} + J_0 g_y(y_{ik}) + \sum_{j \in C, j \neq i} \sum_{m \in D(C_j)} J_{kimj} g_y(y_{jm}) - \sum_{j \in C} \sum_{m \in S(C_j), m \neq k} W_{kimj} g_y(y_{jm}) + I_{ik} + I_0 \tag{14}$$

There are a few differences between equations (13) and (14) which we must discuss:

- The first sum on the right-hand-side of (13) has been replaced by a double sum in (14) over all other companies and all other products they demand. Variable J_{ij} has been replaced by J_{kimj} which expresses the compatibility between product k on offer by C_i and product m on demand from C_j. This may be quantified as

$$J_{kimj} \equiv e^{-V_{kimj}} \tag{15}$$

  where V_{kimj} is defined by (5). Note that as the dissimilarity between two products becomes high, J_{kimj} tends to zero, while it tends to 1 as the products become more similar.

- The second sum on the right-hand-side of (13) has been replaced by a double sum in (14) over all companies and all other products they sell. Variable W_{ij} has been replaced by W_{kimj} which expresses the compatibility between product k sold by C_i and product m sold by C_j. This may be quantified as

$$W_{kimj} \equiv e^{-U_{kimj}} \tag{16}$$

  where U_{kimj} is defined by (9). Note that as the dissimilarity between two products becomes high, W_{kimj} tends to zero, while it tends to 1 as the products become more similar. Note also, that we allow products within the same company to suppress other products if they are sufficiently similar.

- I_i was replaced by I_{ik} which quantifies the stimulation or suppression this particular product gets from outside the ecosystem.

Equations (14) constitute a set of coupled differential equations that if solved, they will give for each product a fitness value. The products with the highest fitness value will be those that will survive and dominate the ecosystem.

3.3 Implementation issues

The systems of differential equations we developed in the previous two sections may be solved as difference equations applied to the nodes of a fully connected network, the values of which are updated in an iterative scheme. Equation (13) may be written as

$$y_{i;new} - y_{i;old} = -\tau_y y_{i;old} + J_0 g_y(y_{i;old}) + \left\{ \sum_{j \in C, j \neq i} J_{ij} g_y(y_j) - \sum_{j \in C, j \neq i} W_{ij} g_y(y_j) \right\}_{old} + I_i + I_0 \tag{17}$$

The values of y_i are initialised to be all equal to 1 at the first step. After each update cycle, we may remove from the system the companies the fitness value of which is below a certain threshold T_3. In a more sophisticated level of investigation, we may allow the introduction of new companies (or new products) with a certain rate, giving them as starting fitness value the average fitness of all other companies.

There are important reasons of why this system of equations cannot be solved as a dynamic system within the Hamiltonian framework, because it does have a stable fixed point and it does not correspond to the minimisation of an objective function. A discussion on this point may be found in [2].

The model may be investigated for various values of its fixed parameters, in order to observe the behaviour of the system under different conditions. One refinement we shall investigate concerns the modelling of the mutual inhibition of two companies: At the moment we model this taking into consideration only the products both companies try to sell. This is fine in an open environment where the supply is infinite. However, in a closed environment when the supply is finite, two companies may exchange inhibitory signals even when they simply want to buy the same product or service. In this case we shall have to modify the calculation of term W_{ij} to rely also on the common products two companies seek to purchase.

In a similar way, equations (14) may be investigated.

4 Conclusions and Future plans

When the demand of an individual customer is to be satisfied, the optimisation problem is separable: each customer wishes to get the best for themselves, independent of how this may affect the other customers. The model we propose to use for the recommender is a simple model that allows the identification of the best product in the database that receives a demand expressed by the customer. Our model may incorporate product attributes expressed in numerical or in linguistic terms. It may also incorporate learning of which attributes of each product customers consider as most important.

The recommender software has been implemented to work with a simple database of 100 companies we developed. However, in order to be fully operational it has to be implemented in conjunction with a true database of companies.

We also developed a model that is currently being implemented for the identification of the fittest companies, in its simplest form, and the identification of the fittest products in the ecosystem, in its advanced form. This model allows one to experiment with its parameters, in particular, it includes parameters that allow the ecosystem to be influenced by external factors and thus allows the user to investigate the response of the system to changing some of those factors. The database we use to develop our model is artificially constructed by us. So, many of the variables used are in their generic form. A real database consisting of real companies will give us the chance to re-implement this model in a more realistic way and perform real experiments.

So, our immediate plans for the future are to collaborate with one of our Computer Science or Business partners to use the models we have been discussing in both deliverables D12.1 and D12.2 in conjunction with a real database of services, in order to produce some joint journal papers. The distribution of this report among the partners will help set us such a collaboration.

Further, for the next stage of the project, we propose to radically revise the work-packages we had defined before the project started. First of all, we know now that the use of Markov Random Fields (MRF) is not appropriate, because MRFs assume symmetric interactions between companies, and this is not the case as shown in section 3. Also, the concepts of Renormalization group transform which are based on the equivalence of MRFs with Gibbs distributions and the accelerated optimisation of complex systems are of no relevance. Instead, the ideas of self-organising networks, and the use of Langevin equation (as discussed in the previous report D12.1), are much more relevant. In our future work we plan to concentrate in completing the implementation of these systems and investigating their behaviour for various parameter settings. The collaboration with a business partner would help to this task enormously.

References

[1] Z. Li, (1999). Visual segmentation by contextual influences via intra-cortical interactions in the primary visual cortex. Networks: Comput. Neural Syst. 10, 187–212.

[2] Z. Li and P. Dayan, (1999). Computational differences between asymmetrical and symmetrical networks. Networks: Neural Syst. 10, 59–77.

[3] Z. Li, (2001). Computational design and nonlinear dynamics of a recurrent network model of the primary visual cortex. Neural Computation, 13/8, 1749–1780.