3

Section Three

Digital Ecosystem Technology

European Commission
Information Society and Media

Section Three

1 Digital Ecosystems Technology and Distributed Nature of Information

Miguel Vidal
TechIDEAS
http://techideas.es/
miguel.vidal@techideas.es

Jesús E. Gabaldón
TechIDEAS
http://techideas.es/
jesus.gabaldon@techideas.es

Introduction

Since the Internet (IP network) inception in 1969, it has been assumed that servers (and therefore services) are unmovable. The IP network was designed as a static network, with the ability of dynamic routing. It is precisely its ability of dynamic routing (only possible if IP nodes are fixed in place) which is the key to its success. The IP network is self-reconfigurable by construction (i.e. highly adaptable).

Thirty years later we have a great quantity of networks in which the nodes that form them (potential servers) are moving constantly (or changing its IP address by means of DHCP). In spite of this, we have tried to maintain the paradigm which gave so good results: the server and service are bound to an IP address.

If it is assumed, though, that users change their location, that devices are portable (laptops, PDAs, etc), that networks do no longer need fixed infrastructures (there are wireless networks like Bluetooth, GPRS, UMTS, 802.11a/b/g/n, WiMax, etc), why not suppose that servers can change their location (say their IP addresses) without the need to interrupt the service they provide?

Coming back to our problem: if the mobile device changes its IP address, is the service it provides the same? Obviously so. It is not so, however, at the network level. The idea must be, therefore, to get rid of the transport level, so as to add mobility to the service.

How to tackle disappearing services?

To provide usable services, in spite of the mobility of the service which supports them, we need an infrastructure that relates the service (which we perceive as fixed) with the device that provides it (which we suppose intrinsically mobile) making it appear as fixed when in fact it is moving. In this way, what remains unmovable is the service provided to the network, not the physical location from which the service is provided.

The obvious solution is to have a fixed data structure (and, therefore, known by all) that keeps the relation between the unmovable element (the service) and the mobile element (the device that supports it).

However, the use of a centralized solution generates the appearance of a weak point, a single point of failure, within the very service repository. When we want to create a system with the ability to move, we need to keep in mind that it implies failure management. Something that moves may disappear, and therefore, we need to design mechanisms that enable to detect and recover from the disappearance of the service.

If we wish to avoid a single point of failure, we have to solve the problem without the use of a central data repository that matches the binding between the IP and the service. Is it feasible?

What is clear is that we have to change paradigm in order to face this new challenges. The approach we propose in the project is just to change the point of view. Instead of looking for 100% of information with a 99.9% probability (i.e. the up time of the system) we propose to have 99.9% of the information with a 100% probability (100% of system up time). This is the way in which nature works, and in which information is stored in natural environments.

Nature of information

Information is an essentially distributed resource in nature. It is not fully contained in central units, but rather distributed across a huge number of parts or elements, which are far spread out in space and time. Consequently, natural systems can hardly ever cope with all the information available. Nevertheless, such a limitation does not prevent natural organisms and ecosystems from evolving; on the contrary, it fosters competition and the “struggle for life”.

On the other hand, most computer databases and networks are still built upon a strongly centralized hierarchy. Centralized systems do work well for most purposes on a small scale. As size increases, distributed systems outperform centralized ones, but the management of decentralized networks results in new challenging difficulties that we are just beginning to address. Here we will give some hints and describe the main characteristics of the new paradigm of computer communities and network information systems, and their advantages and drawbacks in computer sciences.

Fundamentally, there is no direct exchange of information between computers in centralized networks. Therefore, all traffic passes through a server: e.g. Public Switched Telephone Networks (PSTN) are strongly centralised communication networks. This type of networks has driven human society to Internet, and will probably allow us to go much further. However, other technologies like IP Telephony, File Sharing (P2P) and its related applications have shown an unparalleled development during the last years. Indeed, since its onset, this breakthrough has been achieved by young and small, but highly skilled companies.

One of the main limitations of centralized networks is bandwidth. Since all data pass through central servers, these are under strong pressure. Consequently, providing sufficient bandwidth for millions of users would demand onerous investments in infrastructure and technological skills. However, distributed and decentralized networks (which we shall from now on name Computer Communities) do not have servers but nodes, and these can play both roles.

Computer communities assume most of the structural and functional characteristics of the natural communities and ecosystems. They include essentially: self-organizing, self-healing, self-protecting and self-optimizing. Self-healing is the capacity to recover or rebuild a functional and optimized topology after one or several node disconnections. This is a complex behaviour emerging as a side effect of the local activity of certain nodes. Nevertheless, most man-made systems like computer networks are designed to self-protect against external perturbations and internal errors.

The main goal of distributed networks is to use a large number of nodes with variable connectivity in unified form in order to minimize central organization. This is linked to four reasons:

1. to avoid having a single point of failure,
2. due to resource limitations,
3. for a more efficient use of distributed information and resources,
4. and finally, to increase the performance and stability of the system.

In addition, most of the centralized computer networks suffer from the structural and functional limitations of closed systems. They were not designed to growth indefinitely and boundlessly, therefore the growth rate of a computer network strictly follows the law of diminishing returns, which means that the scalability cost is not linear but exponential in size. On the contrary, scalability costs remain linear in distributed networks. The larger the number of nodes in the network the greater the robustness and stability, as well as the total transport rate, service performance and efficiency. There are thus strong reasons for developing such type of networks.

In practice, centralized networks cannot grow indefinitely. Bandwidth is a fundamental resource for such networks that easily becomes limiting. As such, it is not possible to ensure an infinite bandwidth supply and, therefore, growth will progressively be reduced to zero for economic or technological reasons. Actually, the growth of centralized networks can only be carried out by human intervention. This is because the growth of such networks is achieved by a physical (hardware) upgrade of the system or addition. Human action still remains fundamental in most network maintenance and upgrading services. Human intervention is, however, one of the main tasks which one seeks to minimize, or even avoid, for the future.

Technology

The following chapters address the technology developed within the Digital Business Ecosystem project in order to build a highly resilient distributed infrastructure enabling the construction of an ecosystem of digital components. This infrastructure is a self-healing and self-managed system that avoids the need for a DBE corporation which would be in charge of infrastructure maintenance.

2 Ecosystem Oriented Architecture (EOA) vs SOA

Pierfranco Ferronato
Soluta
http://www.soluta.com
pferronato@soluta.net

Introduction

As James Moore [1] pointed out, a Business Ecosystem is based on a dynamic interaction of organizations which evolve over time in terms of capabilities and roles. To this extent this section will describe why the Service Oriented Architecture (SOA) is not adequate to face such challenges which are unique in the context of a Digital Ecosystem (DE). The author will highlight such differences and describe the features of a new architectural style, called the Ecosystem Oriented Architecture (EOA).

The chapter will explore the fact that an Ecosystem Oriented Architecture is not a "sort of SOA", nor is it just a "bigger SOA". A DE employ a broad set of digital components, such as: software services, business services, knowledge, representations of the economy, etc. Software services being just one of them. To explain the differences between EOA and SOA, this article will focus on software services. All the components interact together, forming a digital ecosystem. A whole set of new problems are to be addressed, namely responsive alignment with the business, decentralization, ownership of a distributed knowledge base, self-organisation and self-healing. EOA is a new mindset in decentralized architectures for Digital Ecosystems.

A Digital Ecosystem implementation needs to support a particular dynamic scenario where dynamic business service aggregations and evolutions are key. Neither are B2B market place solutions able to adequately tackle such challenges. DE has to "... exploit the dynamic interaction (with cooperation and competition) of several players in order to produce systemic results in terms of innovation and economic development"[2].

It is indeed true that the recent achievements in Business to Business (B2B) implementations are enabling enterprises to accelerate the dynamic of business, however these solutions are still limited because Service Oriented Architectures (SOA), the prime supporting architectural style of B2B, has been conceived for supporting a single value chain, in a single business domain and usually between a static set of participants; in fact it is often the implementation of a single-organization supply chain. We shall rather name the current implemented scenarios as "business to Business" (b2B) to enhance the fact that the structure is not "democratic": in these cases there is a single Master in the supply chain, which is often the company that owns the chain. This company which is usually a large organization, can impose it's standard to its suppliers and customers, and the 'small' players have to accept the specifications; it's not a peer based model, as the name B2B do inspires, the two 'b' have to be different: this is key. Such b2B environments are thought as being an example of an across enterprise implementation where interoperability is tackled and successfully implemented but -on the other hand- this is an oversimplification, since in reality supply chains do intersect and overlap.

B2B solutions are rarely applied outside the boundary of an enterprise and it is a challenging project: it is cumbersome, and especially complex to maintain. Digital Ecosystems are to be implemented applying a new perspective in Software Architecture that has to overcome the limitations of SOA: an Ecosystem Oriented Architecture (EOA). We intend to pin-point the fact that DE specific features and issues cannot be properly addressed by SOA; there is a need to define a different architectural style that specifically tackles DE requirements from both the functional and structural viewpoint. Applying SOA when dealing with DE implementations overlooks the problems.

SOA has been conceived in the context of intra enterprise systems: in essence, the assumption is that any aspect either functional or structural is managed (or manageable) via a central governance entity. The infrastructure is under control and managed via a single department unit: network appliances like routers, firewalls, cables, routing and topology are planned and managed centrally. In addition, also the functional specifications of the SOA are planned in advance either in joint meetings between parties or defined by a single central authority. The WSDL representing the common technical contract for service invocation are defined up front and are to be used by all the partners in order for the value chain implementation to be effective: this is the environment in which SOA was born and where it is actually used most of the time. SOA is an architectural style that evolved from EAI, RPC and CORBA where the focus was on Applications, Procedures, Objects; focus on services was added later but still with an "intra enterprise" mindset (Figure 1 below).

An SOA implementation is often conceived, funded and implemented by an organization with the sole goal of supporting and increasing its business, as a consequence this drives the entire environment which is single-party centred and does not follow the competition/evolution core feature of a DE.

In an enabling ICT-based infrastructure aiming at supporting the economic activity of networks of business clusters (or business ecosystems) fostering systemic synergies with special focus to SMEs. DE scenarios are changing the rules, because the focus is moving from "intra enterprise" to "across enterprises" (inter community) and soon "across communities". Using SOA for implementing a DE, that requires enlarging the participants in a broader spectrum, supporting a wider set of functional models, running over the Internet, spanning a WAN, is underestimating the problem. As a matter of fact, reading the literature[3], and from the author's experience, it is evident that dynamism and flexibility are key for running a Business.

In a digital ecosystem the value chains are overlapping, they are not partitioned but intersect each other;

- the social and business network topologies are not hierarchical [4];
- a single functional reference model cannot be implemented;
- there is no single point of management from both the business and structural viewpoints.

Taking the previous premises into account, the final goal should be the integration of the services offered by each of these SMEs, without involving extra investments in items not related to their businesses (such as information systems). Therefore, the system should be operated automatically without human intervention. In other words, the system that supports integration of the aforementioned services should have self-organising capabilities.

On the other hand, it should be decided what would be the minimum infrastructure required to allow the presence on the Internet (that is: great portals, operators, ISPs, etc) of these businesses without the need for great investments, or great resources. To reach this goal, two clear premises were identified: minimum hardware, and zero maintenance. The need for maintenance and administration should be eliminated, wiping off the greatest source of cost. This now opens the challenge of zero-administration, which requires the development of software technology that provided self-organization mechanisms. Along with these elements, the system should bring us the possibility to publish the presence of a given business (identified as a service) from the moment it's connected to the DBE network, and the ability to detect its disconnection to eliminate the service from all the contents in which it was referenced.

Functional Reference Model

Digital Ecosystems cross business domains and different value chains, for this reason they are characterized by not having a single functional reference model. Since it is not feasible to define up front all the required functional models, which are intricate, complex and continuously changing, the ecosystems participants need to be free to define, publish and use any models that they consider adequate for their business.

As an example, a book distributor or reseller might create a model that represents their application interface to allow consumers to search, browser, order and buy books. This model could be published and implemented by their service component. Other competitors in the ecosystem will probably do the same in autonomy and this will end up with a set of different APIs that would burden the effort of a bookstore when required to automate the order process; for each supplier/distributor a different technical adapter is required. This constraint would slow down the rate of adoption and lock stores on a single supplier because of the effort required to align the software again. This would represent the dead end of the ecosystem; without fast business alignment, there will be no evolution.

One rather ingenuous approach to overcome this issue is to have all the book distributors sit around a table in an association defining "The" reference model for the book store sector. From direct experience of the author(2), this is a method that does not scale for a long time and, assuming that the participants are able to converge to a suitable model, there will soon be other "competing standards" (notice the oxymoron) that would again create interoperability problems.

Also, maintaining the specification would be very time consuming and in the end it would not be possible to keep it aligned with the business requirements: new features driven by the end users or marketing would incur the risk of being left behind, waiting for the new specification to emerge or -even worse- of being implemented diverging from the standard. As a consequence, the expected well ordered mechanism would soon break.

This scenario is a gross over-simplification of the models what might be found in a DE, especially considering cross value and supply chains. The overall map of models would be so complex and articulated that managing them would be impossible. As a comparison, we can recall the Internet map(3)[5] and its topology; no-one can have full control of it. It emerges rather autonomously from complex usage mechanisms that have been investigated only in recent times. Even maintaining the functional models of a complex ERP project, with well-defined boundaries and dependencies, can be very difficult and impossible for a single party; changes and updates are often tough tasks to accomplish. In a business ecosystem this effort cannot be addressed at all, and a new mindset and approach in this sense is required, and the SOA approach is hence inadequate. In addition, assuming that an ecosystem can be managed is a contradiction in terms. The keyword is "self-regulation", "self-adaptation"[6] and the EOA has to implement the required instruments for this to happen, it is useless to fight and oppose the dynamic nature of a DE, it is better to support it.

The way to go about then assumes the inability to control the reference models; we might assert that there is no reference model at all, and take all the required architectural decisions to support it and let the ecosystem converge, dependant on time, in a model. What is fundamental to assume when defining the architecture of a DE is to recall that it is a highly dynamic environment where the IT related frictions and inertias needs to be reduced to the minimum. This is the prime condition that will allow an ecosystem to self-converge and adapt.

The architecture needs a mechanism to allow participants to:

- publish any model;
- investigate which is the most adequate to their needs;
- adopt it (and change it) in a totally free and uncontrolled space (regulatory and restrictive features shall only be added as a means to avoid hacking or spamming the environment).

A structured and highly connected repository has to manage the models, their dependencies and their association with implementing services. As an example: if the book distributor could inspect the ecosystem (specifically using a model repository), it could detect that there is a functional model for the book sector that is adopted by 75% of bookstores and another one less adopted (hence less connected) but closer to its technical needs and more straightforward to implement due to the better alignment with their back-end systems. The distributor has the chance to decide whether to adopt the most connected model, hence facilitating the migration and adoption by bookstores, or to stick to the easy way with an obvious drawback regarding the level of adoption. In this scenario it is evident that bookstores (the service consumers) on the other hand will try to reduce the number of different models in order to lower their integration efforts and favour the quality of the service offered. The balance between the symmetric aspects is the basis for competition and evolution.

Model repository

In SOA, UDDI is the catalogue of services and service models. They are mixed with binding information, there is no separation between the technical specification and the functional one, and in addition the service end-point is also written in the service specification. Such structure is a consequence of the fact that UDDI has been conceived as a static catalogue of intranet services(4); it is clearly a consequence of the fact that it descends from classical RPC approaches. UDDI is essentially a catalogue of programmatic resources.

For example: two different book distributors might use the same technical specification of the service (e.g. WSDL) but have different kinds of discount policies, different return policies, different quantity discounts or serve different regions. The WSDL is a technical specification that exposes the service protocol that in turns implements the business service. What has to be modelled and delivered is the business service rather then the mediator to the service. In an SOA the need to model the business specification is not a prime need because there is no economical transaction involved. SOA is often implemented, in the author's experience, in a context where the associated business transaction costs are null (zero). Nevertheless, the writer is aware of some SOA implementations (rather tough though) in which an invocation implies an effective business transaction, i.e. some "money exchange". But also in these cases the participants and the services involved have been defined up-front -statically- and the business models are known in advance: there is no dynamic discovery or negotiation and for this reason -under these assumptions- SOA works fine: in DE on the other hand it would not scale. Reference documentation about UDDI mentions "Companies can establish a structured and standardized way to describe and discover services"(5), but a DE is not a structured or standardized environment.

In a DE, the model repository needs to manage business models instead of programmatic specifications. OMG's XMI is the prime choice for encoding models because it is a platform-independent specification; it supports meta-modelling, model dependency, merging, inclusion, inheritance and versioning. XMI is able to represent semantically rich model specifications, where WSDL is not. Services in DE need to make use of more complex specifications, the definition of software interfaces is not sufficient: there is the need to express the underlying business model. The plain interface specification is not relevant in the context of an ecosystem where services need to be explored automatically via recommendation agents: having computable business models is essential.

In addition, the functionalities provided by the repository need to support an enormous amount of unstructured and related information. The users, either a software component or a human being, must be able to navigate the intricacy of models and their dependencies in order to identify those that are most useful and adequate. In this sense the repository needs to provide intelligent and semantically aware research and recommendation tools[7].

It is also essential is to decouple the service model catalogue from the actual service instance catalogue: "The service registry".

Service Registry

The service registry contains the references to actual services published in a DE associated with the technical and business models. Each entry includes self-contained information about the service (called Service Manifest[8]), made of:

1. service business models;
2. technical specification (i.e. Service APIs);
3. business data;
4. service end-point.

The first type is essentially the business specification (it might be a reference to an entry in the model repository, this is an implementation aspect which is not relevant in this context). The second is the technical specification of the service. The third is information specific to the service instance, for example the name of the published service or the location of the service; in general this information is associated with the business model. The fourth is programmatic information needed to actually invoke the service, for example – it is an over simplification – the IP address and the protocol used.

Whatever way this registry is implemented, the essential aspect is that is has to be extremely dynamic and bind to the actual published service. In SOA it is a great frustration to try to invoke services from information found in the UDDI just to discover that they are not available. The real issue in these cases is that the requesting service is not able to provide the reason for the failure: is it due to the fact that it has been discontinued or because there are some temporary technical issues? In an intranet SOA implementation, the architect has the ability to put all the efforts in order to have a high availability of service: in the Internet this cannot be assured. As a solution, the service entry in the registry needs to be bound with the actual remote published service so that it provides up-to-date status information; since it is too administratively intense to manually keep it aligned, a lease base mechanism is a good technical approach, like SUN's Jini(6) framework dynamic lease management or the FADA framework (7).

As for the model repository, the service registry needs to be MOF8 compliant in order to ease the issues related to model interoperability.

The model repository and service registry represent a single point of failure (SPoF) for the DE architecture and this can jeopardize the entire ecosystem. This issue is addressed via a decentralized architecture, described in section 5 "Single Point of Failure" Chapter.

Basic Services

An architecture for DE needs to consider a set of basic business services to support the ecosystems and facilitate the rapid and correct interaction between business services. A DE without a proper set of basic services is unlikely to be sustainable: the goal is to improve the level of adoption by easing the participants' effort in publishing and integrating services. It is fundamental for example to execute a negotiation process before actually consuming the service (which is not required in an SOA implementation, as mentioned above) essentially because a service invocation in a DE is a business service consumption. For the same reasons, services such are reputation and trust are as fundamental in a DE.

The following services are needed essential to facilitate the bootstrapping phase in a DE:

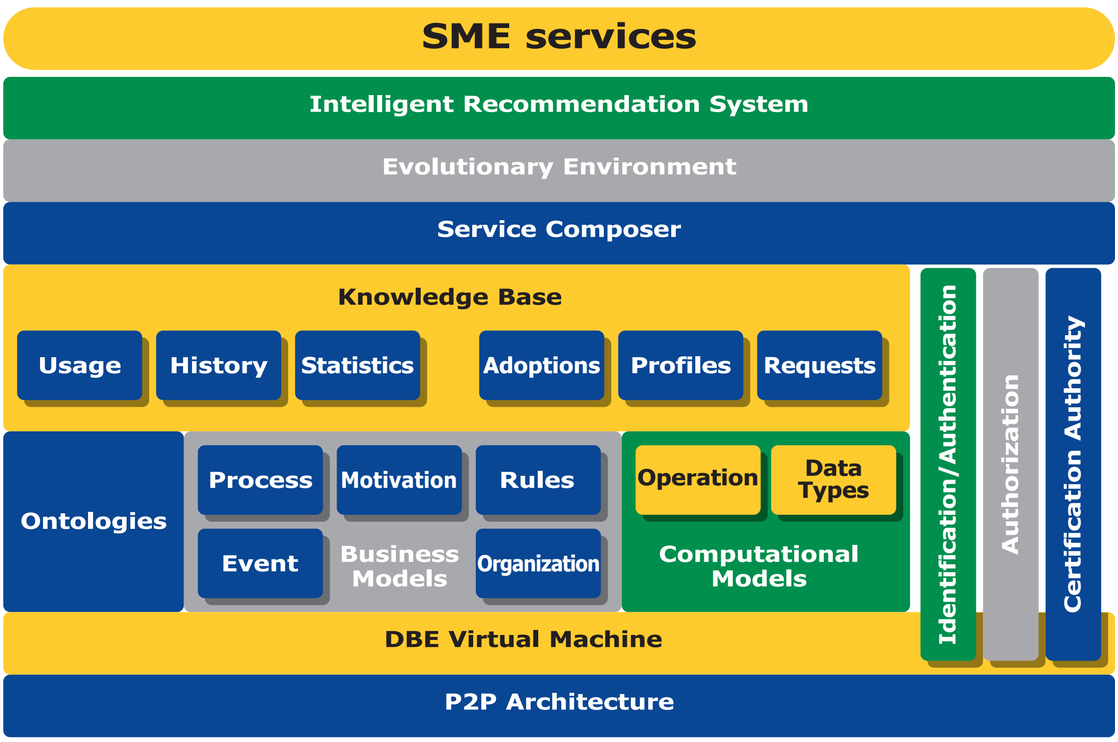
- Payment
- Business Contract & Negotiations
- Information Carriers
- Billing
- Trust
- Reputation
- Legal compatibility

It is however important to underline that all these services do not specifically need to be implemented up front. It is important to support them for example by defining their models in the repository and providing an adequate infrastructure for their implementation: it might be up to participants and organizations to implement them. But some, like the accounting service, need to be supported by the core infrastructure of a DE because it has to adequately intercept the inter services messages.

One of the most significant services required in a DE is support for negotiations. In SOA, in those rare case in which it is implemented in across-enterprise B2B environments, negotiation takes place outside of the IT systems, often through real meetings; in SOA implementations, only the service execution is supported together with a poor search mechanism. In DE, following the definition given at the beginning, the ecosystem is such only if the integration mechanisms are fast and automated. As a matter of fact DE had to replicate in an e-environment what happens in the real world environment.

Fig. 1

An example of the service stack in the Digital Business Ecosystem project (DBE)



In addition there is the need to reconsider other services, although in a different perspective:
- Service Discovery
- Reliability-guaranteed delivery
- Security
- Long running Transactions
- XML Firewall.

Single Point of Failure

The Service registry is a key element for SOA; it is used at run time for service discovery and invocation, for this reason it represents a single point of failure for the entire architecture. If the registry is not available, the services will not be reachable.

This is a key issue also in SOA, for this reason UDDI version 3 has introduced replication schema for cluster of registries that provides high availability feature[10]. It supports both clustering and mirroring, however replications are based on the complete mirroring of nodes; in addition the replication policy is to be accurately planned by an administrator and implemented beforehand. But for a DE, given the complexity and intricacy of the infrastructure, the very frequent changes and the absence of any “root” node, this solution is not adequate.

In DE, the registry is even more critical because service IP addresses change very often, while in a classical SOA all the services are published in static IPs and change quite seldom: caching IPs would not work for long[11].

Setting up a single central fail-safe and highly redundant registry server would be very expensive and would not even guarantee service continuity in case of natural disaster. The alternative solution is to exploit decentralized approaches, i.e. a topology and replication schema that does not make the DE dependant on a single node but rather on a collaborative set of peer nodes (more on this in the next section “Scale free networks”). Instead of a controlled cluster of nodes, there is the need to advocate the use of peer-to-peer networks as the routing infrastructure that improves routing resilience to node failure and attacks on service registries[16]. Such a network of nodes needs to be self healing and self adaptable to the ever changing nature of the requests and traffic: there should not be an administrator. Such kind of solutions would be resilient to node failures and would not loose information under critical circumstances. Nodes within this network interact in rich and complex ways, greatly stressing traditional approaches to name service, routing, information replication and links.

In such types of networks, data replication within nodes takes place intelligently: entries migrate automatically in relation to requests, moving data toward nodes that started the request. In this way, as in typical caching mechanisms, information is copied from the closest nodes so as to increase the probability that sequential requests get fulfilled in less time. It is relevant to notice that “close” in this context is relative to speed and not to geographical distance, since often in Internet hub nodes 100 km apart are faster to ping then local servers. Moreover, such a copying mechanism replicates redundant information among nodes so as to increase tolerance in case of nodes failure. As a matter of fact the new Italian Health Care System is adopting such a decentralized architecture for the Patient Health Record registry[12].

Avoiding having single points of failure for an EOA is essential. Beside the technical non marginal aspect of having a more reliable system, the DE will not suffer from the “big brother syndrome”. With a decentralized P2P based architecture the knowledge which is held in by the model repository and the service registry is not managed by a single institution which could tamper with it at the expense of the community by imposing unwanted control. A DE is self-regulated and self adaptable by definition[13] and a central institution with the potential power to control the environment from a technical and functional point of view could hinder the entire process of adoption and sustainability. Consider for example what would happen in case the organization hosting the service registry decided to shut it down. Such possibility would impede the adoption of the DE.

DE founds its entire sustainability and existence on knowledge about models and services. Participants in the DE are providing and using models while actively participating and being part of a business community, they are hence scared about loosing models. The owners of DE knowledge need to be the community itself, to this extent a peer-to-peer network (see next section “Scale Free Network) is a good approach because it is democratic; it provides participants with the possibility to offer resources to host part of this knowledge.

The significant drawback is the implementation: such a peer-to-peer infrastructure needs to be self-healing and self-adaptable. But there are already some frameworks and tools that support the enhancement of the properties of Scale Free networks.

Scale Free Networks

Most of the solutions in SOA, like the cluster of UDDI registries, are based on hierarchical structures because this is the way humans proceed in order to deal with complexity, i.e. in order to create comprehensible models. But as a matter of fact, the social and business networks in the real world are not hierarchical at all: this is essentially the reason why information models become more and more unmanageable with the increase in complexity. The more the IT systems push in the direction of being aligned with the business, the more the IT becomes unmanageable. Below a certain degree of complexity, any model can be reduced to a hierarchy that represents a good approximation, but with the increase in complexity it becomes impossible to stick to a hierarchy because reality is not as simply structured: it is based on different models and topologies: Scale Free networks[15].

The scale free networks are well described in the literature[14], we do not intend to describe it in this paper; what we state is that since scale free networks are the topology at the basis of business and social networks[15], a proper EOA has to support it and define appropriate mechanisms in order to let it emerge in a self organized way without human intervention.

In order for a Scale Free Network to emerge, it is necessary to support connectivity, proximity and preference[16]; it is dangerous and it represents a risk in the architecture to over-impose an unnatural topology. The advantage of a Scale Free Network is well described in the literature, essentially it is tolerant to a random failure of nodes and the properties of a "small world" allow efficient searches[17][18].

The author envisages a service registry and a model repository implementation that take advantage of such kind of networks essentially because this is the way they exist in the real world and supporting this vision will help align the ecosystem with the business -as is required.

Technologies are already available and they make use of concepts like the Tuple Space or the Distributed Hash Table, for example Sun's Jini™ Network Technology[10], FADA[11], Bamboo[12], Cord[13] and others; there are also commercial implementations like GigaSpaces©[14]. P2P architecture can help, even if they can be used to infringe copyright: there is no need to be prejudiced, a technology is not bad per se, but it depends on the way it is used. The Digital Business Ecosystem (DBE)[15] has made a significant step forward in this direction.

Conclusion

Service Oriented Architectures (SOA) do not scale nor address the new challenges addressed by the architectures for Digital Ecosystems. The author envisions a new architectural style, called the Ecosystem Oriented Architecture (EOA). Three levels of service specifications are to be identified and addressed[20]:

- service models: a catalogue of business and computational models to be reused;
- service implementation: a catalogue of services descriptions (Service Manifest) implementing some models together with their data;
- service instances: service name and endpoint to actually invoke and consume a service.

In DE it is essential to have a repository of models separated from the registry of services[20]. The model repository needs a whole set of discovery features and supports XMI in order to implement model driven capabilities like dependency, versioning, merging and inheritance. Services need to be described also from the business viewpoint: the computational specification is not sufficient in DE because services are not known in advance and the discovery process needs to be smarter and based on business specifications.

The service registry needs to overcome the static limitation of UDD-like services and be dynamically bound to actual published services. In the near future a lot of mobile services are expected and these devices are going to make use of dynamic IPs, enhancing SOA based approaches is not enough. The service instances are to be resolved at run-time via a sort of DNS service.

Given the nature of a DE, the architecture needs to avoid single points of failure, the best approach envisioned is to make use of P2P technology to implement a decentralized data storage system (as opposed to the SOA centralized or distributed approach).

Basic services need to be implemented and defined up front in order to sustain the ecosystem, such as negotiation, information carriers, payments, accounting, billing and others. While SOA essentially supports only the service execution phase, a DE has to support the entire business service life-cycle including service selection (as opposed to service search), negotiation, agreement, contract specification, consumption and delivery.

In any aspect, either functional, structural or topological we have to reflect the real ecosystem in the DE: after over 40 years we realize that we are still applying the Conway[16] law that states "Organizations which design systems are constrained to produce designs which are copies of the communication structures of these organizations"[21], i.e. any piece of software reflects the organizational structure that produced it, and a DE is no different.

References

[1] Dr. James F. Moore, "The Death of Competition", Collins, May 21 1997, ISBN 0887308503
[2] F. Nachira et alii., EC Discussion paper, Bruxelles, September 2002
[3] P. Dini et al, "Towards Business Cases and User-Oriented Services in Digital Business Ecosystems", Conclusions FP7 Workshop on Needs and Requirements of Regions, Brussels, 18 April 2005
[4] A. L. Barabási, H. Jeonga, Z. Néda, E. Ravasz, A. Schubert and T. Vicsek, "Evolution of the social network of scientific collaborations", 16 January 2002.
[5] Milena Mihail, Christos Papadimitriou and Amin Saberi, "On certain connectivity properties of the internet topology", Journal of Computer and System Sciences, Volume 72, Issue 2, March 2006, Pages 239-251
[6] F. Nachira, M. Le Louarn, "ICT supporting businesses and industry", "Innovation ecosystems", FP7 – General Orientations, February 2006.
[7] Paolo Dini, Tuija Helokunnas, Pierfranco Ferronato, Angelo Corallo, Neil Rathbone, "Towards a semantically rich business modelling language for the automatic composition of web services", e-Business Research Centre (eBRC) 2004
[8] P. Ferronato, "Technical Architecture of the Digital Business Ecosystem Project", MDA Technical Forum, Tokyo, 20th, 21st October 2004
[9] T. Heistracher, T. Kurz, C. Masuch, P. Ferronato, M. Vidal, A. Corallo, P. Dini, and G. Briscoe, "Pervasive service architecture for a digital business ecosystem", In ECOOP First International Workshop on Coordination and Adaptation Techniques for Software Entities (WCAT04), Oslo, June 2004.
[10] "UDDI Version 3 Features List", OASIS, 2004, http://uddi.org/pubs/uddi_v3_features.htm
[11] M. Vidal, P. Hernandez, "Building a Digital Ecosystem", SUN, Java Conference 2005
[12] Pierfranco Ferronato, Stefano Lotti, Daniela Berardi, "Titolo: Strategia architetturale per la Sanità Elettronica", Innovazione Italia, Dipartimento per l'Innovazione e le Tecnologie, 31/03/2006 14.17
[13] P. Dini et al, "The Digital Ecosystems Research Vision: 2010 and Beyond",
[14] Albert-Laszlo Barabasi, Duncan J. Watts et al, "The Structure and Dynamics of Networks", ISBN: 0691113572, Princeton University Press (April 17, 2006)
[15] Albert-Laszlo Barabasi, Reka Albert, Hawoong Jeong, "Mean-field theory for scale-free random networks", Department of Physics, University of Notre-Dame, Notre-Dame, IN 46556, USA
[16] Albert-Laszlo Barabasi, "Linked: How Everything Is Connected to Everything Else and What It Means", ISBN: 0452284392, Plume; Reissue edition (April 29, 2003)
[17] Nancy A. Lynch, "Distributed Algorithms", Elsevier Science & Technology Books, 1996.
[18] Fletcher, George, Hardik Sheth, "Peer-to-Peer Networks: Topological Properties and Search Performance", Indiana University, 2003, http://www.cs.indiana.edu/~gefletch/papers/l597report.pdf
[19] P. Ferronato, "WP21 - DBE Architecture Requirements Del 21.1 - Preliminary design, usage scenarios, and architecture requirements", D.B.E. Digital Business Ecosystem Contract n° 507953, October 2004, http://www.digital-ecosystem.org/Members/aenglishx/linkstofiles/deliverables/Del_21.1_DBE_Architecture_Requirements.pdf/download
[20] P. Ferronato, "WP 21: DBE Architecture Requirements, Del 21.2: Architecture Scope Document", D.B.E. Digital Business Ecosystem Contract n° 507953, October 2004, "http://www.digital-ecosystem.org/Members/aenglishx/linkstofiles/deliverables/Del_21.2_DBE_Core Architecture Scoping Document_Release C.pdf/download"
[21] Melvin Conway, an issue of Datamation, April 1968

3 DE Services in Ecosystem Oriented Architecture

Paul Malone
Waterford Institute of Technology,
Telecommunications Software & Systems Group (TSSG)
http://www.tssg.org
pmalone@tssg.org

Introduction

This paper introduces the concept of Digital Ecosystem Services deployed in an Ecosystem Oriented Architecture (EOA). As described in Chapter 3.1, the EOA concept is based on a peer-to-peer architecture that allows the digital ecosystem to be effectively pervasive and decentralised. There is no single point of failure and the ecosystem itself is owned by participant SMEs and not by a governing body or organisation. The knowledge and services are spread across the supporting nodes through a peer-to-peer self-healing architecture. Services are described in both the technical and business point of view hence allowing semantic and business types of search and discovery.

This paper will describe the requirements of services deployed in an EOA and how the DBE project [1] have realised these requirements for SME service deployment.

Digital Ecosystem Services

A conceptual framework for DBE services emerged from the work done by Soluta.net on DBE architecture requirements. This framework is similarly applicable to DE services in an EOA. This can be seen below in Figure 1. Every DE service is specified using a set of formal languages that aims at defining the business models as well as the technical interface in a platform implementation way. Thanks to the model driven approach taken, business models can be transformed and mapped into platform specific models without specific user interventions. The family of languages adopted defines the service's DNA that fully specify the service and the ability to evolve and adapt. Each structural component is decentralised. Such an approach potentially allows the ecosystem to be self-healing and survive technical and network failures.

The technical architecture eases the entire integration and adoption process by providing an infrastructure at both sides of the pipe in a consumer-provider point of view. Services already residing in legacy systems can quickly become DE enabled, thanks to the decoupling approach provided by EOA and the MDA (Model Driven Architecture [2]).

A Service Composer allows for browsing of the collected published services and the creation of a workflow that can be executed and published as an atomic service. The chained nature of such new service will be completely transparent to the consumer application or user.

Given the meta-service and meta-modelling approach followed, there is not a single model or service that cannot be replaced or enhanced. No pre-defined or immutable behaviour is coded in an EOA. [3]

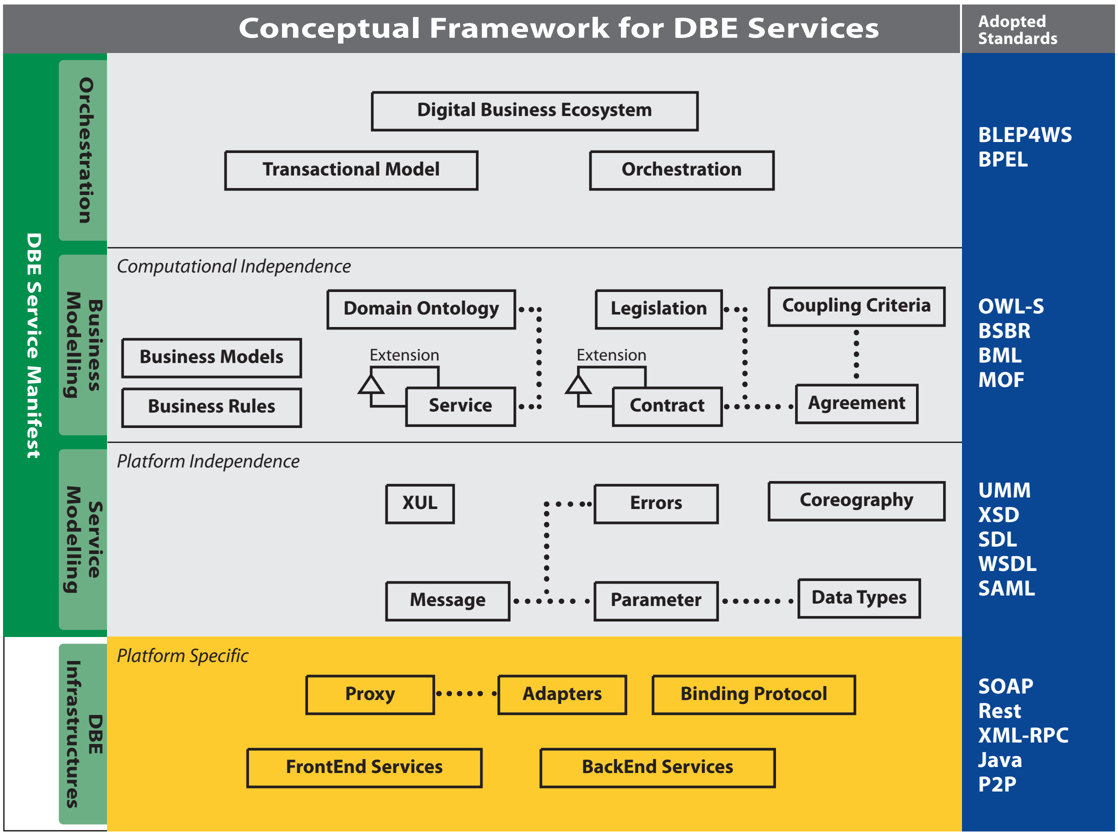


Fig. 1
DBE Services Conceptual Framework (Source [3])

The Service Factory
Creating and Maintaining the Services

The starting point for a participant in a digital ecosystem is to model and create a DE Service through a Service Factory. A Service Factory is a set of tools to aid the developer in the creation of DE Services, through the association of those services with appropriate models and deploy those services in the Execution Environment. The Service Factory was realised in DBE through the open source DBEStudio Project [4].

The DBEStudio is an Integrated Development Environment (IDE) for the Digital Business Ecosystem (DBE). It was developed using the Eclipse framework [5] and includes a set of eclipse plug-ins that allow business services to be analysed, and corresponding software services to be defined, developed and deployed. When the DBEStudio is launched it is configured by the user to connect to the core DBE Services via a URL denoting the location of a running Execution Environment.
Descriptions of the core plug-ins are provided below.

BML Editor

The BML(Business Modelling Language [6] Editor plug-in is a visual modelling tool and provides a UML-like Graphical User Interface (GUI), similar to that of well known UML editors. The tool supports the modelling tasks and stores the created models in the DBE Knowledge Base (KB) deployed in the Servent. The current version of the editor supports both the semantic description of the services offered by an SME and the business model of the particular

SME. The former provides to the user the ability to create service models according to semantic service language metamodel and the latter to create business models based on the BML metamodel. Both metamodels are described using OMG's MOF 1.4 [7].

BML Data Editor

The BML data editor takes MDA M1 level BML models (created with the BML Editor) and allows the developer to populate these in order to create M0 level instances of those models. These M0 models then represent the business data associated with actual running instances of service business models.

Ontology Analysis Tool

The Ontology Analysis Tool plug-in provides a visual environment based on a UML-like graphical user interface that enables business analysts to deploy domain specific ontologies in order to describe the business requirements of SMEs in the context of the DBE project. The ontology definitions is based on the Ontology Definition Metamodel (ODM) compatible with the OWL, can be represented using XMI (XML Metadata Interchange [8]) technologies and can be stored either locally (in the local file system) or into the DBE Knowledge Base using the JMI (Java Metadata Interface) standard.

Service Exporter

The Service Exporter plug-in enables a user to export a DBE project and deploy it as a DBE service to a Servent. Using a set of wizards the user can add/edit their deployment information. The tool creates a DBE Archive (DAR) file, which contains a particular structure for deployment within a Servent. This plug-in is also integrated with the Metering Wizard to allow users to add metering information at deployment time.

Metering Wizard

The DBE metering wizard is run as an optional element of the Service Exporter plug-in. This wizard allows for the selection of parameters upon which the filters installed in the servent can extract usage data. The SME deploying the service can select methods and parameters of those methods that require metering. This usage data can then be used by OSS (Operation Support Systems) type services installed in the Execution Environment. In DBE, open source accounting services have been implemented and deployed and make use of this usage data in applying charges for services usage as well as providing billing information.

Manual Composer Tool

BPEL Editor

The DBE Composer Tool is a BPEL (Business Process Execution Language [9]) editor to allow for the creation of composed services for execution in the DBE ExE. The design of the BPEL Manual Composer tool centres on a graphical editor and a composition wizard for this composition language. This editor is the core component as it allows the user to graphically design the composed service as a workflow process, while the wizard uses simple rules to help a user to select services and create model structures. The implementation of both the editor and the wizard fully support the BPEL meta-model. The design of the editor provides a 3-view editor where each view has a more abstracted representation of the BPEL model. The intention is to provide two levels of graphical abstraction and granularity to suit both a semi-technical user and a BPEL developer, where the wizard and the graphical editor attempt to address the needs of both user types respectively.

SDL Editor

The SDL (Service Description Language [10]) editor allows SMEs to define their services from a technical point of view. The editor provides a graphical means of defining service interfaces and expresses those interfaces via XML instances of the SDL schema

SDL2Java Compiler

This plug-in takes the XML SDL instance created by the SDL editor and generates a set of Java packages and interfaces which need to be implemented by the developer in order for the service to be deployed successfully in the SBE Execution Environment.

Service Manifest Composer

The Service Manifest composer is responsible for the creation of the Service Manifest (SM) [11]. The Service Manifest acts as an advertisement for a deployed service and contains both business and technical models of the service instance. The Service Manifest Composer is responsible for the creation of this from the models created from the other plugins. This SM is deployed with the service and inserted into the Semantic Registry (Service Registry). See the next section for details.

Execution Environment Deploying the Services

As described in the chapter on EOA, an execution environment for DEs must provide certain functionalities; a suitable peer-to-peer network, a service container, a service composition engine, a model repository and a service registry.

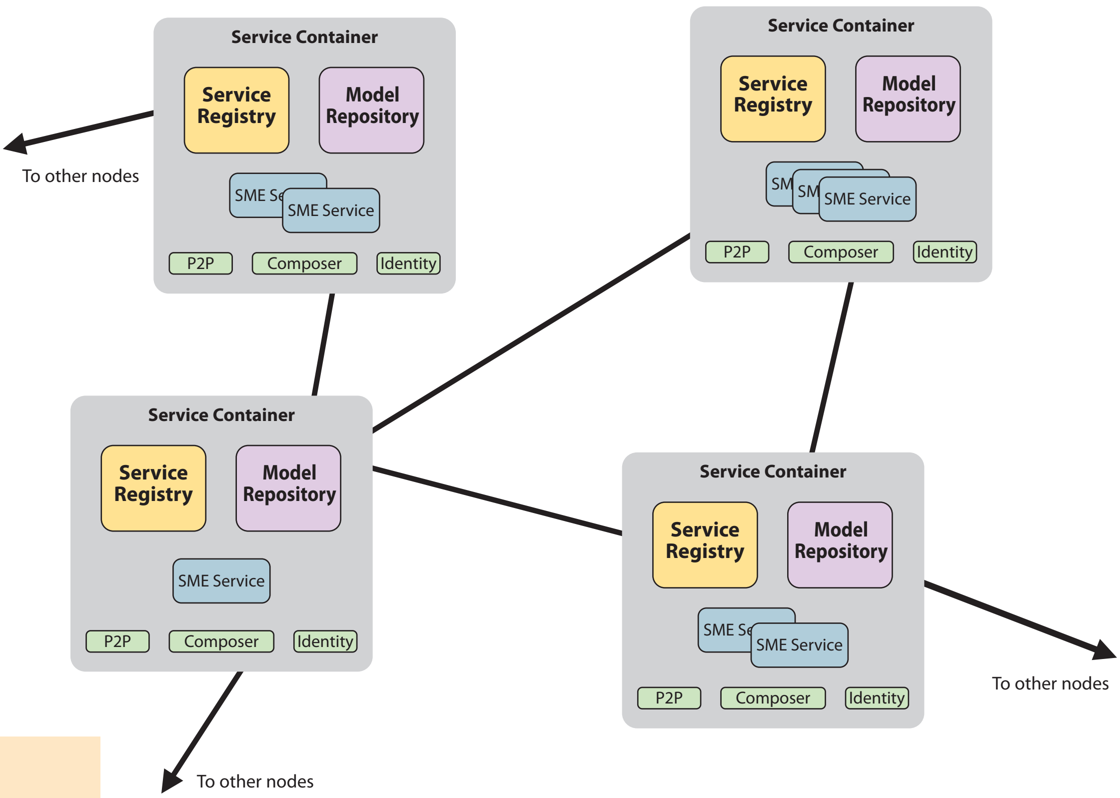


Fig. 2
Distributed Infrastructure of EOA

The distributed infrastructure of the EOA can be seen in Figure 2. The nodes are connected via the peer-to-peer overlay network. Each node contains the core functional components required for the EOA to be successful. Each node also hosts services offered by SMEs. The services have the ability to migrate, providing high availability of service provision in the event of a node failure or downtime.

Each of the supporting core components is briefly discussed below and the DBE implementations also referenced. Detailed descriptions of these implementations are available in the chapter on Distributed Infrastructural Services.

Service Composer

The composer provides the ability to orchestrate and manage the execution of composed service chains. The DBE implementation integrated the open-source workflow engine ActiveBPEL [12] and extended it with a custom invoke handler to deal with invocations. This provides the advantages of a standards-based workflow description and execution with the added bonus of DBE peer-to-peer interaction.

Model Repository

The model repository is a business model container. Services deployed in the ecosystem are associated with one or more of these models. The preferred encoding option of models in the registry is XMI (XML Metadata Interchange)[8] (see chapter on Ecosystem Oriented Architectures). The DBE implementation, developed by the Technical University of Crete, is called the Knowledge Base and provides distributed persistence satisfying the OMG MDA [2] approach taken in DBE.

Service Registry

The service registry is a repository for references to deployed services in a digital ecosystem. Each entry is associated with one deployed service and contains information on business models, technical specifications, business data and the service end point. The DBE implementation of this component, developed by the Technical University of Crete, is the Semantic Registry, which is used to store a Service Manifest [11] per deployed service.

Peer-to-Peer Overlay Network

A peer-to-peer overlay network is essential in providing a digital ecosystem with the assurance of no single point of failure and robust distributed knowledge and service provision through a distributed set of collaborative nodes. This approach improves routing resilience to node failure. A suitable implementation needs to be self healing and autonomically adaptable to the changing nature of the requests and the traffic.

FADA (Federated Autonomous Directory Architecture) [13] was the initial peer-to-peer implementation deployed in DBE. FADA emerged from the European project Fetish. Trinity College Dublin also developed a DHT (Distributed Hash Table) peer-to-peer implementation based on their peer-to-peer architecture design for DBE [14]. More details of both these implementations are available in the chapter on Distributed Infrastructural Services.

Identity

A fully distributed identity management system is essential for providing trust among the participants in digital ecosystems. Identity constitutes one of the basic building blocks for providing accountability functionality to B2B transactions. Services need to be associated with an identity of the service provider and service consumers need also to be identified for accounting and access control purposes. However creating a decentralised, robust and trustworthy identity management system with no dependencies on third party certificate authorities is a challenging proposition.

In the DBE project, Trinity College Dublin developed a core identity component overlayed on top of the DHT implementation. This contitutes a decentralised solution that provides the redundancy and management features inherent in the DHT. The system has the ability to verify keys associated with service invocations.

Conclusion

Successful digital ecosystem service deployment in EOAs requires a set of mechanisms for the definition of business and technical models, the creation of service interfaces based on these models and robust decentralised service hosting. These mechanisms have been described in this paper, together with descriptions of how these mechanisms have been realised in the DBE project.

Acknowledgements

John Kennedy, Intel Ireland. Pierfranco Ferronato, Soluta.net. Dominik Dahlem, Trinity College Dublin. Fotis Kazasis, Technical University of Crete. Juanjo Aparacio, TechIdeas.

References

[1] DBE Project, *Digital Business Ecosystem*, http://www.digital-ecosyten.org (22.3.2007)
[2] MDA: *Model Driven Architecture*, http://www.omg.org/mda/ (22.3.2007)
[3] DBE Deliverable D21.3: *DBE Architecture High Level Specification*, available at http://www.digital-ecosystem.org (22.3.2007)
[4] DBE Studio: http://dbestudio.sourceforge.net (22.3.2007)
[5] Eclipse Framework, http://www.eclipse.org (22.3.2007)
[6] DBE Deliverable D15.3: *DBE Business Modelling Language, 2nd Release*, available at http://www.digital-ecosystem.org (22.3.2007)
[7] OMG MOF: Meta-Object Facility (MOF™), version 1.4, http://www.omg.org/technology/documents/formal/mof.htm (22.3.2007)
[8] XMI: *XML Metadata Interchange*, version 2.1, http://www.omg.org/technology/documents/formal/xmi.htm (22.3.2007)
[9] Microsoft, IBM, Siebel Systems, BEA, SAP. *Business Process Execution Language for Web Service*, version 1.1. May 2003. http://www-128.ibm.com/developerworks/library/specification/ws-bpel/ (22.3.2007)
[10] DBE Deliverable D16.1, *Service description models and language definition, Part 1 -SDL Definition*, available at http://www.digital-ecosystem.org (22.3.2007)
[11] P.Ferronato, *Technical Architecture of the Digital Business Ecosystem Project*, MDA Technical Forum, Tokyo, 20th, 21st October 2004
[12] Active Endpoints, *The ActiveBPEL Open Source BPEL Engine*, http://www.activebpel.org/index.html (22.3.2007)
[13] FADA, *Federated Autonomous Directory Architecture*, http://fada.sourceforge.net (22.3.2007)
[14] DBE Deliverable D24.3: *DBE Peer-to-Peer Architecture Design*, available at http://www.digital-ecosystem.org (22.3.2007)

4 Modelling Languages, BML, SBVR

Angelo Corallo
e-Business Management Section,
Scuola Superiore ISUFI
University of Salento
http://www.ebms.unile.it
angelo.corallo@ebms.unile.it

Virginia Cisternino
e-Business Management Section,
Scuola Superiore ISUFI
University of Salento
http://www.ebms.unile.it
virginia.cisternino@ebms.unile.it

Introduction

The fast growth and the diffusion of the World Wide Web, its effect on trade, economic transactions and communication systems, allowed and boosted the diffusion of the network society. Radical changes in the geography of costs, production and human and social capital affected the space-time of business altering the traditional business frontiers (Castells, 1996).

These phenomena allowed companies and organisations to grow globally, reacting to the climate of uncertainty through the decentralisation of production and innovation. Shifting activities to the electronic dimension allowed an increase in efficiency and the reduction of coordination costs. At the same time, organisations geographically distributed start to interact and trade globally, enhancing the growth of networks of distributed companies. Transient networks and organisations as portfolios constitute new organisational forms, which exhibit unmatched connectivity, flexibility and adaptability. Competing in this context poses a number of integration problems that enterprises have to tackle, such as the integration with customers, among suppliers or between design and manufacturing sites. As a consequence, integration ceases to be only a matter of interconnecting physical and software applications; more than this, it starts to be considered in a wider perspective, that concerns the overall business and all its aspects. This means that, in order to enable a different use of enterprise and network resources to better achieve strategic objectives, a global business integration is needed and considered as a key factor for successful enterprises.

The most interesting consideration arising from these concepts is related to modelling issues. Things to be integrated and coordinated need to be clearly defined and expressed: the more a model is effective, the easier is the definition of communication, coordination, control and exchange mechanisms. Thus, enterprise modelling is clearly a prerequisite for enterprise integration.

An *enterprise model* is a computational representation of the structure, activities, processes, information, resources, people, behaviour, goals and constraints (Vernadat, 1996) of a business, government, or other organisations. It can be both descriptive and definitional - spanning what is and what should be. The role of an enterprise model is to achieve model-driven enterprise design, analysis and operation. The finality is to make explicit facts and knowledge that add value to the enterprise or can be shared by business applications and users. Besides an effective enterprise integration, the main purposes of business modelling are to support the analysis of an enterprise and, more specifically, to represent and understand how the enterprise works, to capitalize acquired knowledge and know-how for later reuse, to design and redesign a part of the enterprise, to simulate the behaviour of the enterprise, to make better decisions or to control, coordinate and monitor some parts of the enterprise. Enterprise modelling techniques and associated visual languages are very important and useful to support new approaches to enterprise business transformation and improvement, developing smart businesses and new networked organisations.

Approaches and methodologies

Enterprise modelling was born in the United States at the beginning of the 80's and emerged through large Computer Integrated Manufacturing projects. In the mid-80's, Europe launched several projects on enterprise modelling giving birth to several enterprise modelling languages. As a result, in the 90's many commercial tools dealing with enterprise modelling or business process modelling appeared on the marketplace, as well as a myriad of workflow systems, each one with its own modelling environment. This intensive production of tools has led to a Tower of Babel situation in which the many tools, while offering powerful but different functionalities and semantics, are unable to interoperate and can hardly or not at all communicate and exchange models. Currently, enterprise modelling is a wide and complex domain containing many different methodologies, languages, tools and techniques, often developed in different context for different scope. Such languages could be roughly divided among the approaches developed mainly from a business perspective and the languages and methodologies related to design and development of IT applications.

In the stream of the knowledge modelling related to the business area, there are many languages and frameworks devote to model specific characteristics of enterprise. Among them, the *Zackman Framework* (Zachman, 1987) and the *Generalised Enterprise Reference Architecture and Methodology* (GERAM) (Bernus et al., 1997) represent the most general approaches; more operational frameworks are the *Integrated Enterprise Modelling Method* (IEM) (Spur et al., 1996), the *Integrated DEFinition methodology* (IDEF) (NIST, 1981), the *Architecture of integrated Information Systems* (ARIS) (Scheer, 1992), the *Process Specification Language* (PSL) [8], the *Workflow Process Definition Language* (WPDL) (WMC, 2002), the *Business Process Modelling Language* (BPML) (BPMI, 2002), and the *Business Rules* (BRG, 2000) approach.

Among the standards related with software design and development there are the *UN/CEFACT Modelling Methodology* (UMM) (UN/CEFACT, 2003) and the *Rosetta Net* (RosettaNetl, 1998). In the same group, there are some industry initiatives and de-facto standards, such as those promoted by the Object Management Group: the *Model Driven Architecture* (MDA) (Frankel, 2003), the *Unified Modelling Language* (UML) (OMG, 2003), the *Meta Object Facility* (MOF) (OMG, 2003a) and the *Semantic of Business Vocabulary and Business Rules* (SBVR) (OMG, 2006). Other relevant standards in metadata definition and exchange are related to the work of W3C. Among them there are the *eXtensible Markup Language* (XML) (Bray et al., 1998), the *Resource Description Framework* (RDF) (Lassila et al., 1999) and the *Ontology Web Language* (OWL) (McGuinness et al., 2004).

The Business Modelling Language

The Business Modelling Language (BML) has been created as the business language for the DBE project. Its main aim is thus to create a general framework enabling business people to represent the business knowledge related to DBE services and to the enterprise that stands behind such services, in order to allow communication mechanisms based on semantically rich information models.

One of the most interesting characteristics of BML is that it has been designed in order to bridge the gap between business and technology perspectives. If on one side BML allows to express business concepts, that is the actual concepts, actions and events that business people have to deal with as they run their businesses, independently by technological aspects, on the other side it grants a rigorous mapping to formal logics, to make business knowledge accessible to software.

Another important aspect, is related to the software production methodology and to the effort in realizing mechanisms for software development based on models. The BML framework is aimed at supporting business analysts to express in a formal and well defined way all the knowledge necessary to represent a customer company.

Defining the BML framework has implied the definition of the BML syntax (how information is expressed) and the BML semantics (how obtaining a shared meaning). In order to decouple these two fundamental issues, it has been chosen to define a general architecture based on a meta-modelling approach. In this way, the information model, that could change during the evolution of the project, is related only to the meta-model content, without implication on the general architecture of the framework.

The BML general architecture

The BML architecture has been developed applying a meta-modelling approach. Such an approach refers to the Meta-Object Facility (MOF), a standard defined by the Object Management Group (OMG, 2003a), that aims at guaranteeing a universal approach to describe modelling constructs. It is based on a multi-layered architecture, where each level is defined through the constructs defined in the upper one. This design choice has allowed to exploit many advantageous features for the semantics definition, since every item defined in whatever level could benefit from the semantics inherited from upper level models, in terms both of definition and relationships. This approach also grants coherence among different sub-domains since, once a general meta-model has been defined, it is possible to define different coherent domain models, used to provide a shared meaning to the businesses of a given industry or community. Moreover, this design choice permits to obtain interoperability and interchange capabilities through the MOF mapping technologies. As an example, MOF XMI (OMG, 2002) encoding can be used to transfer models conforming to the BML meta-model as XML documents and to transform the BML meta-model itself into an XML document, for interchange between MOF-compliant repositories. This allows to translate BML artefacts in formal language expressions that could be interpreted and processed by software. The choice of MOF is also related to the objective of supporting simple and smart software development processes. MOF is the cornerstone of the Model Driven Architecture (Frankel, 2003), one the most interesting approaches in software development methodology. Starting from this basic assumption, the BML architecture has been designed as shown in Figure 1.

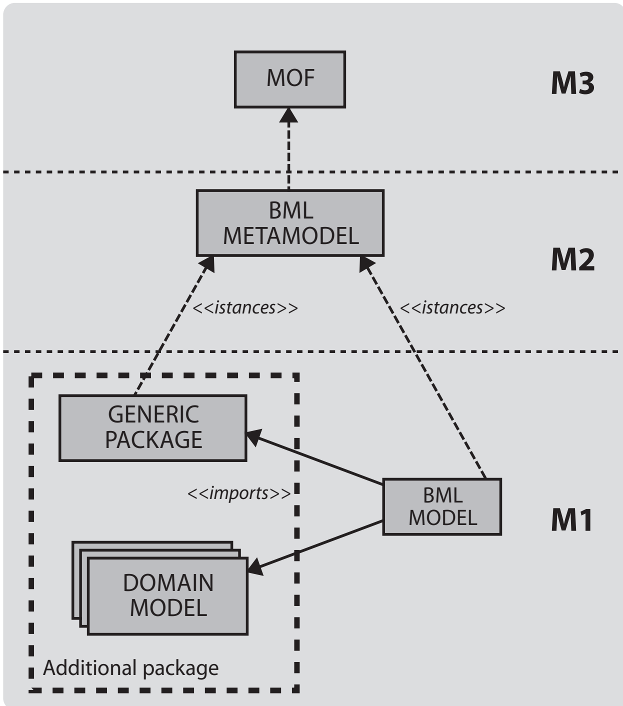


Fig. 1
The BML general architecture

The M3 level contains the MOF constructs, which define the set of elements used to define meta models (e.g. Class, Attribute, Association). The M2 level contains the BML meta-model, that is the semantics necessary to create business and domain models in the DBE project. More specifically the BML meta-model, provides several packages that constitutes the BML information model and contains all the primitives necessary to define a business. Such primitives are abstract and independent from specific business domain, in order to be instantiated in whatever business domain. The information at M2 level is considered long lifecycle information and requires complex competencies in both business and modelling field; this implies that it can be modified only by DBE experts. This MOF-based meta model has been indicated as *BML 1.0*. In the M1 level, it has been developed a *Generic Package*, containing cross-domain concepts and constructs (e.g. the concepts of price or customer). Moreover, this level will contain several business *Domain Models*, a repository of knowledge shared within a specific business domain and developed by domain experts or a given community, and the *BML Models*, developed by SMEs to describe themselves and their services. Such models can be created instantiating concepts from a domain model, from the Generic Package and directly from the BML meta-model.

The BML Semantics

The definition of the BML semantics is based on two main key elements. From one side, it has been decided to align it with the main standards for business and e-business modelling. First of all, the BML meta-model is provided with a packages structure compliant to the OMG *Architecture of Business Modelling* framework [223]; moreover, the specific content of the packages has been developed using as input information coming from other e-business standard, such as UMM (UN/CEFACT, 2003) and ebXML (Lassila et al., 1999). On the other side, from a theoretical perspective, the meta-model has been built on the basis of the Zachman Framework for Enterprise Architecture (Zachman, 1987).

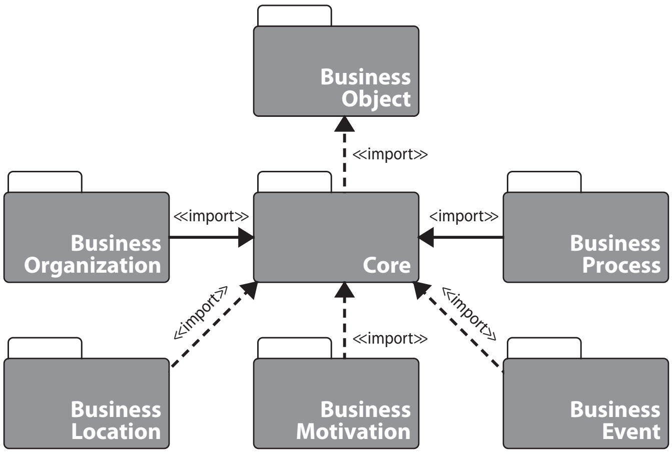


Fig. 2
The BML packages structure

The resulting meta-model is split up into seven packages, as shown in Figure 2: *Core*; *BusinessOrganization*; *BusinessProcess*; *BusinessMotivation*; *BusinessLocation*; *BusinessEvent*; *BusinessObject*. These packages are relatively independent of one another and allow to obtain the needed range of expressivity and flexibility.

In more details, the *Core* package contains the basic classes generic enough to be defined as a separate set within the BML meta-model and extended by the other six packages. Many of the element of this package are defined using the ebXML core component library. The *BusinessOrganization* package aims at describing the whole organization, specifying the entities involved in the business, their resources and how they can interact. The *BusinessProcess* package is used to define the behavioural elements of an organization. It contains all the meta-concepts needed to describe how the organization actually performs its business. One of the purpose of this package is the description of the dependencies existing between partner processes, modelling the business actions and objects that create, consume and exchange business information. The BusinessProcess package encompasses two main areas, closely related each other: an agreement area, concerning parties' engagements created by collaborative activity, and a behavioural area, related to the organizational working activities to perform a specific business. The *BusinessMotivation* package aims at describing the elements an organization analyses and settles in order to make choices and define its action. The *BusinessEvent* package contains the meta-model for describing the events able to influence the business behaviours, including rules about their temporal ordering or partial ordering in the business activity cycle. The *BusinessLocation* package contains the meta-model for describing geographic locations, business sites, geographic areas, volumes, and perimeters, political subdivisions and boundaries, and logical connections between them. Finally, the *BusinessObject* package contains the business data types constructors. It allows the modeller to create the types needed in order to model a particular domain or business, without references to their operational features, that are aspects typical of programming language.

Semantics for Business Vocabulary and Business Rules

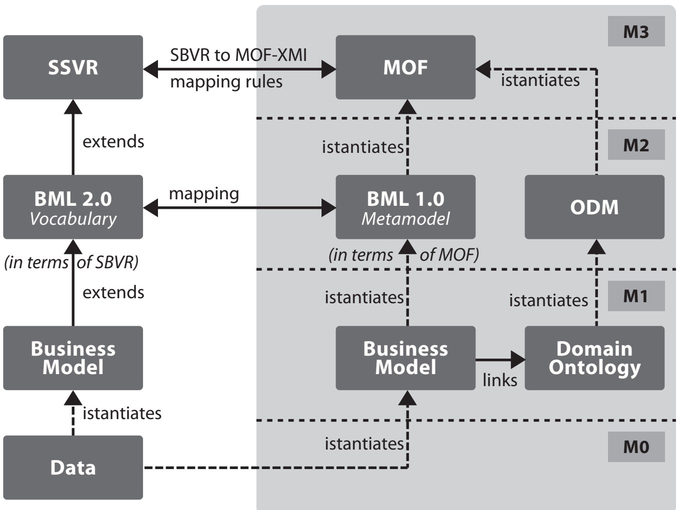


Fig. 3
The SBVR-based BML architecture

During the BML inception and first design phases, while there was wide consensus about using MOF and a meta-modelling strategy, there were many open issue about the concrete syntax to be adopted in the framework. In this first phase, the ontology based modelling was the main candidate to cover this role. Starting from June 2004, the BML team started to have interaction with the OMG Business Enterprise Integration Domain Task Force (BEI DTF) members and to exchange ideas and solutions with them. Starting from shared objectives, this interaction led to a convergence toward the concrete syntax proposed by the BEI DTF. In the wider context of the Architecture of the Business Modelling, BEI DTF produced an OMG Request for Proposal in 2003 (OMG, 2003) named *Business Semantics of Business Rules* (BSBR) with the aim of creating a concrete syntax enabling business people to model their own domain in a natural language and granting an effective MOF and XMI mapping. The adopted submission was the *Semantics of Business Vocabulary and Business Rules* (SVBR) (OMG, 2006), that is based on business rules and vocabularies definition through a Structured English notation.

Even though, this represented a very ambitious and risky objective, it was decided to explore the possibility to adopt this innovative standard, obtaining a new version of the meta-model (named as *BML 2.0*). Figure 3 describes this new perspective in the BML architecture. As en example, the business concept *product*, defined through BML 1.0as a MOF class, using BML 2.0 is described as a BML Vocabulary entry (Table 1):

Table 1

Product	
Definition	**business item** used for describing tangible things or substances produced by natural process or manufacturer
Definition	the **product** allows describing what an organization offers to its customers and **partners**
Genera Concept	**business item**
Example	room or food

Conclusion

Some open issues concerning BML, its adoption and development will be faced in the next future. First of all, the practical application of BML will give the opportunity to test the architecture stability, the meta-model completeness and the concrete advantages for business community in adopting the approach. This application is currently based on the BML 1.0. The immaturity of the SBVR standard, and in particular the lack of specific tools, are the main reasons behind this choice. In this perspective, the most interesting future development is related to SBVR and to the realisation of a business modeller. Such a tool is an SBVR editor that will allow business people to create their own model, through the definition of their own vocabulary and rules. At the moment, the editor exists in a prototypal form,the *SBeaVeR* project (ISUFI, 2006), that should be developed from DBE project partners in collaboration with open source communities

References

[1] M. Castells, *The rise of the network society*, Blackwell Publishers ltd. Oxford, 1996.
[2] F. B. Vernadat, Enterprise modelling and integration: principles and applications, Chapman & Hall, 1996.
[3] J. A. Zachman, "A Framework for Information Systems Architecture," IBM Systems Journal, Vol. 26, No. 3, 1987.
[4] P. Bernus and L. Nemes, "The contribution of GERAM to consensus in the area of enterprise integration", in Proceedings of ICEIMT'97, Springer, 1997.
[5] G. Spur, K. Mertins, and R. Jochem, *Integrated Enterprise Modelling*, Beuth Verlag GmbH, Berlin, 1996.
[6] NIST, National Institute of Standards & Technology, *IDEF0 method report*, http://www.idef.com/Complete Reports/idef0, 1981.
[7] A. W. Scheer, *Architecture of Integrated Information Systems - Bases forCompany Modeling*, Berlin, 2nd edition, 1992.
[8] NIST, National Institute of Standards & Technology, PSL website, http://www.mel.nist.gov/psl, 2002.
[9] WMC, The Workflow Management Coalition, http://www.wfmc.org, 2002.
[10] BPMI, The Business Process Management Initiative, http: //www.BPMI.org, 2002.
[11] BRG, The Business Rules Group, *Defining Business Rules: What Are They Really?*, July, 2000
[12] UN/CEFACT, United Nations Centre for Trade Facilitation and Electronic Business, *UN/CEFACT Modelling Methodology (UMM) User Guide*, 2003
[13] RosettaNet, *RosettaNet Standards*, http://www.rosettanet.org/, 1998
[14] D. S. Frankel, *Model Driven Architecture – Applying MDA to Enterprise Computing*, Wiley Publishing inc., 2003
[15] OMG, Object Management Group, *Unified Modelling Language Specification, 2003*
[16] OMG, Object Management Group, *MetaObjectFacility (MOF) Specification v1.4*, 2003a
[17] OMG, Object Management Group, *Semantics of Business Vocabulary and Business Rules Specification*, March, 2006
[18] T. Bray, J.Paoli, C.M Sperberg-McQueen, *Extensible Markup Language (XML) 1.0*, World Wide Web Consortium, 1998
[19] O. Lassila and R. R. Swick, *Resource Description Framework Specification*, World Wide Web Consortium, 1999
[20] D.L. McGuinness and F. van Harmelen, *OWL Web Ontology Language Overview*, World Wide Web Consortium, 2004
[21] OMG, Object Management Group, *XML Metadata Interchange (XMI) Specification v2.0*, 2002
[22] S. Hendryx, Hendryx and Associates: Architecture of Business Modeling (2003)
[23] OMG, Object Management Group, *Business Semantics of Business Rules – Request For Proposal*, 2003
[24] ISUFI, *The SBeaVeR Project*, http://sbeaver.sourceforge.net, 2006

5 Digital Ecosystem Topology: Information in Nature

Jesús E. Gabaldón
TechIDEAS
http://techideas.es/
jesus.gabaldon@techideas.es

Juanjo Aparicio
TechIDEAS
http://techideas.es/
juanjo.aparicio@techideas.es

Power Law Distributions in Nature

Power laws describe frequent scaling trends found in nature, in areas as disparate as physics, biology, sociology, economy and semiotics. Among many other examples, Vidondo et al. (1997) described the body size spectra of seston (materials in suspension) as a Pareto (or Bradford) type distribution, which is itself also a power law commonly used in economy for statistically describing the allocation of wealth among individuals or the exponentially diminishing returns.

Power law distributions show long right-tailed (skewed) distributions in linear axis plots, but they are often described by log-log plots and, therefore, they appear as rather linear distributions with a characteristic slope coefficient. However, power laws should not to be confused with log-normal distributions which show a linear distribution in short range loglog plots; power laws show consistently linear trends over 3 or more orders of magnitude.

Power laws can describe the energy or information transport between hierarchical levels of a system, whether natural (ecosystems) or artificial (man-made). The transport coefficient, proportional to the slope of the distribution function in log-log plots, can be representative of the stability, fitness and efficient performance of a system against the environmental conditions. They can be also used for assessing the equilibrium status or succession stage of ecosystems and natural communities (see also Rinaldo et al., 2002).

Scale-Free Networks and the Barabási-Albert Model

Scale free networks are a category of power law distribution networks. Scale free graphs or networks have been proposed as generic, yet universal models of network topologies that exhibit power law distributions in the connectivity of network nodes (Li et al., 2005).

However, despite its popularity, there does not exist a precise definition of scale-free networks ("a common property of many large networks is that the vertex connectivities follow a scale-free power-law distribution"; Barabási & Albert, 1999), either in the context of complex systems or in computer sciences. Scale-free graphs are, essentially, more an empirical evidence than a rigorous definition. Yet, Margalef stated in 1991 that one cannot expect "reasonable answers from models with constant interaction parameters". This is, in mature, stable and efficient systems, the relationships between the components of a system are usually flexible, and show a complex topological structure that systems analysts try to grasp, understand and reproduce by means of generative models (see below).

The Barabási-Albert (BA) was a simple algorithm defined basically in terms of growth and preferential attachment. Growth tends to increase as new nodes are added over time, and preferential attachment describes the ability of a node to grab new links. Proportionally, the higher the number of links of a node the more likely is to receive new links. The non-null preferential attachment probability of a node can follow a linear function.

The network begins with at least 2 connected nodes, and new nodes are added to the network, one at each time step. Each node connects itself to an existing node with a certain probability. This probability is proportional to the number of links that the node already has. Such strategy is analogous to autocatalytic reaction kinetics, which are present in many biochemical pathways.

Information Management

The nature of a system's complexity and behaviour is closely related to its internal structure. In the case of computer networks, as well as other complex systems, topology plays a fundamental role in their apparent behaviour. Topology may be crucial for the system's environmental fitness and stability : fault tolerance, self-healing, self-configuration, selfoptimization, etc. All of them are properties of mature systems.

On the other hand, complex systems are essentially heterogeneous, and show differentiated functional behaviour and hierarchical roles. Complex behaviour depends (or relies) on both the system's topology and a robust hierarchy. Furthermore, topology and hierarchy are closely coupled and, therefore should not be separately considered.

In an open universe in which the network topology may change quickly as nodes join or leave the community, new approaches have to be developed in order to efficiently address search and management tasks. In such a case, the use of global algorithms is no longer possible. Computers will have to learn to manage themselves in this unbounded and highly dynamic environment. This is something which each of the organisms that populate this planet continuously face from birth to death. Therefore, ecosystems and natural communities can be used to help illustrate and improve our understanding of man-made systems like computer networks.

Yet ecosystems show an enviable robustness against both internal instabilities and (external) environmental perturbations and never collapse ultimately. Network servers can still fail, and they do eventually, with increasing economic and social costs, among other significant drawbacks. However, to better comprehend how robust ecosystems actually are, we must fully understand the mechanisms at a level where interaction between individual organisms occur.

The first step to enable the emergence of intelligent behaviour, is to let the nodes interact freely with each other, in order to memorize or recall other nodes and their interaction experiences. This implies that they must have some sort of memory. But essentially nodes must be willing to interact with each other, and to exchange information. The information that nodes will attempt to disseminate is essentially about their own characteristics and properties, but especially about the services they have to offer to others. And each node remembers at least some of the received information input from their neighboring nodes for a limited period of time. Intelligent behaviour is, however, just a byproduct or apparent property of this type of autonomous communication.

Building Distributed Networks

The ubiquitous presence in nature of a specific form such as scale-free is often interpreted as a signature of universal underlying generating mechanisms. One can classify the complexity of a system according to different criteria. This implies some sort of fundamental knowledge about it. But on the other hand, the lack of knowledge about the emergence and development of complexity and hierarchies is still evident and profound.

There is also a clear relationship between the efficiency of a system (or environmental fitness) and its internal structure. But there is still a significant lack of knowledge about how hierarchy and complexity emerge and develop. Mature systems show internal gradients of information and complexity. This feature can be understood in terms of an increasing number of hierarchical levels. However, in the case of distributed computer networks one can force the topology to adopt the internal structure of complex scale free systems. This makes the systems more robust against single point failures, also improving the efficiency of information transport.

On the other hand, when dealing with distributed networks one has to assume that these have no boundaries or well-defined limits. In other words, a user's computer will always directly interact with another user's computer, not with a central server. And this implies that the other computers will continuously join and leave the network and, therefore, their information, services, etc., will not always be accessible. The total information available will always be much greater than that which is accessible through central servers. Information distributed networks will be available in vasts amounts, although not accessible in their entirety, since they are not exhaustive. Central servers have never been complete either, although they intentionally perform that way.

Generative Algorithms

The lack of precision in the scale-free definition evidence a still partial knowledge about the underlying processes that drive to the appearance of well structured systems. However, there is a clear correlation between the overall system's efficiency and its scale-free topology. Therefore, in spite of this limited knowledge about the generative processes that pull systems towards a mature and efficient stage, we attempt to reproduce such structures with the aim of improving the performance of complex processes or systems.

Preferential attachment is often the main generative mechanism upon which many power law distributions develop, but it is not the only mechanism that is able to produce scale-free distribution graphs or networks. The main objective is to scale the proportion of highly connected nodes ("hubs") and less connected ones.

Hubs play different roles (among the most relevant):
1. Holding the network together, thus preventing fragmentation.
2. Re-organizing the network structure or topology according to environmental feedbacks, promoting the development of functional hierarchical levels.

The functional presence of hubs is fundamental for the complex behaviour of the system. They are responsible of its robustness, but they are also its "Achilles' heel". The disappearance of a single populated hub may imply fragmentation, which increases the difficulty of incident recovery mechanisms in distributed computer networks.

The main hypotheses of distributed networks (self-healing, self-organizing, self-optimizing, etc.) rely on the functional activity of highly connected nodes at different hierarchical levels, hence the importance of the coupling between the environmental response of the system and the activity of these hubs.

Relevance to the digital ecosystem problem

As mentioned above, the topology of a distributed system is critical in order to guarantee the system performs well under favorable conditions, and to guarantee that it performs at all under unfavorable conditions, such as random failures and attacks.

Given the resilience shown by biological networks (of cells, of individuals, ...) and their self-healing capabilities, it is extremely desirable to incorporate these traits into the design of a digital ecosystem.

In particular, the search problem is one of the first encounters with the radical difference between distributed and centralized systems. In a centralized system searching for a particular item (data, service endpoint, ...) is a problem bounded in time. It is also decidable: the central authority either has or it doesn't have the requested item, and its reponse will tell which case we're in front of.

In a fully distributed system (that is, lacking a central authority) there can be no guarantees as to whether a given item exists or it doesn't exist in the network. The item of interest may lie in a node of the network that is not accessible anymore.

It is interesting to note that the distributed system is not less resilient than the centralized one. In the centralized case, the response tells us without a doubt whether the item exists or not, but if the central authority stops being accessible, the whole system falls down. In the distributed case, the failure of a node renders that node unusable and the items it provides unaccessible, but the system as a whole is still working. Centralized systems can not say the same.

If the probability that a certain node in the network may crash is *p*, in the centralized case the whole system works with probability *p*. But in the distributed case, the probability that all items are accessible is:

p^n times where n is the number of nodes in the network

Given that p is usually less than 1, the probability of all the items being accessible at once is smaller for a distributed system than for a centralized system. However, the probability that at least one node is accessible in the distributed system is:

$1 - (1 - p^n)$ times where n is the number of node in the network

This value is greater than 1-*p*, the probability that a centralized system doesn't fail, if $p \in [0,1]$, which is true for a probability value.

But, in order for the system as a whole to be able to keep running in the face of node failure, the system must be able to find a way to reorganize to work around failed nodes, or otherwise be extremely resilient to those failures.

Scale free networks provide an opportunity to avoid system failures as a result of individual node failures. Given their properties, outlined above, they can also search in an efficient way for nodes that can take the roles of the failed nodes, thereby exhibiting self-healing capabilities that don't degrade the performance of the system.

The power-law tail of the degree distribution of scale-free networks means that there is a decreasing number of nodes as the degree of those nodes grows. That is, there are very few nodes with very high degrees. It is those nodes that interconnect most other nodes of the network. The power-law distribution also tells us that these nodes don't connect exclusively with each other. Therefore they are considered *hubs* in the network.

Given that the existence of these hubs is known, it is possible to create strategies that improve the performance of a search for items in the distributed network.

It must be noted that the search problem mentioned above is a particular instance of the more general 1-to-many communications problem.

It can be seen that the potential of scale free networks with respect to digital ecosystems can not be ignored.

References

Barabási, A.-L. & R. Albert. (1999). Emergence of scaling in random networks. *Science, 286*: 509-512.

Bollobás, B. & O. Riordan. (2003). Robustness and vulnerability of scale-free random graphs. *Internet Math.,* 1: 1-35.

Li, L., D. Alderson, R. Tanaka, J.C. Doyle and W. Willinger. (2005). *Towards a theory of scale-free graphs: definition, properties, and implications (extended version).* Technical report: CIT-CDS-04-006, Engineering & Applied Sciences Division.

MacPherson E. & C. M. Duarte. (1994). Patterns in species richness, size, and latitudinal range of East Atlantic fishes. *Ecography*, 17: 242-248.

Margalef, R. (1991). *Teoría de los sistemas ecológicos*. Publicacions de la Universitat de Barcelona. Spain.

Rinaldo, A., A. Maritan, K.K. Cavender-Bares and S.W. Chisholm. (2002). *Cross-scale ecological dynamics and microbial size spectra in marine ecosystems.* Proc. R. Soc. Lond. B., 269: 2051-2059.

Vidondo, B., Y.T. Prairie, J.M. Blanco and C.M. Duarte (1997). Some aspects of the analysis of size spectra in aquatic ecology. *Limnology & Oceanography*, 42(1): 184-192.

6 Distributed Infrastructural Services

John Kennedy
Intel Ireland
IT Research - IT Innovation Centre
http://www.intel.ie
John.M.Kennedy@intel.com

The Digital Business Ecosystem (DBE) can be described as being a peer-to-peer semantically-aware service-oriented architecture (SOA) for Small and Medium sized Enterprises (SMEs). Architecturally, the DBE has been divided into an Execution Environment (ExE) to host services, a Service Factory to design and develop services, and an Evolutionary Environment (EvE) to help optimise the system. All three of these blocks have extensive requirements for distributed functionality. These needs are met by the DBE's distributed infrastructural services. This paper introduces the key distributed infrastructural services of the DBE, highlighting some of the key technical challenges that they tackle, and the features they deliver.

Introduction

The DBE project software has been architected [1] into three core systems: a distributed runtime environment known as the Execution Environment – the ExE; an environment for designing and implementing services known as the Service Factory; and an autonomous, distributed optimisation system known as the Evolutionary Environment - the EvE.

Whilst the ExE and EvE are both fundamentally distributed in nature, the Service Factory also includes important distributed components: these, for example, allow constructed SME services to benefit from shared ontologies and business models.

The ExE [2] – the peer-to-peer network of DBE runtimes – hosts all of the services of the DBE. It has three main components: the servent, which can be considered to be a DBE service container; the locally accessible core components, which implement fundamental, low level DBE functionality; and finally network addressable services, which are either infrastructural or SME specific. Infrastructural services deliver critical functionality such as the ability to locate specific SME services on the peer-to-peer network [the Semantic Registry], the ability to store arbitrary content in a distributed, replicated space [the Distributed Storage System], and the ability to browse the DBE network using a web-like interface [the DBE Portal].

The servent, hosted on the SourceForge project Swallow, actually provides a container for service adapters. This name stems from the fact that they can provide a gateway to existing services, though nothing in the design or implementation prevents them from being full-fledged, self-contained services. The adapters are deployed onto the servent. The servent creates the endpoints necessary for clients to be able to consume those services, and publishes their existence.

The publication of the existence of service adapters consists of the creation of a data item which contains the URL of the endpoint of the service, the models the service conforms to, and additional, user-defined tags. These data items are then registered in FADA.

The servent also serves as the consumer of those services. That means a client application uses the servent to locate and run service adapters that may be deployed in a remote servent. The exact location of the service adapter is unknown to the client. The servent acting as a client of services uses FADA to search for the service the client wants to run, gets the endpoint data and performs the necessary communications with the remote servent, which then executes the service adapter.

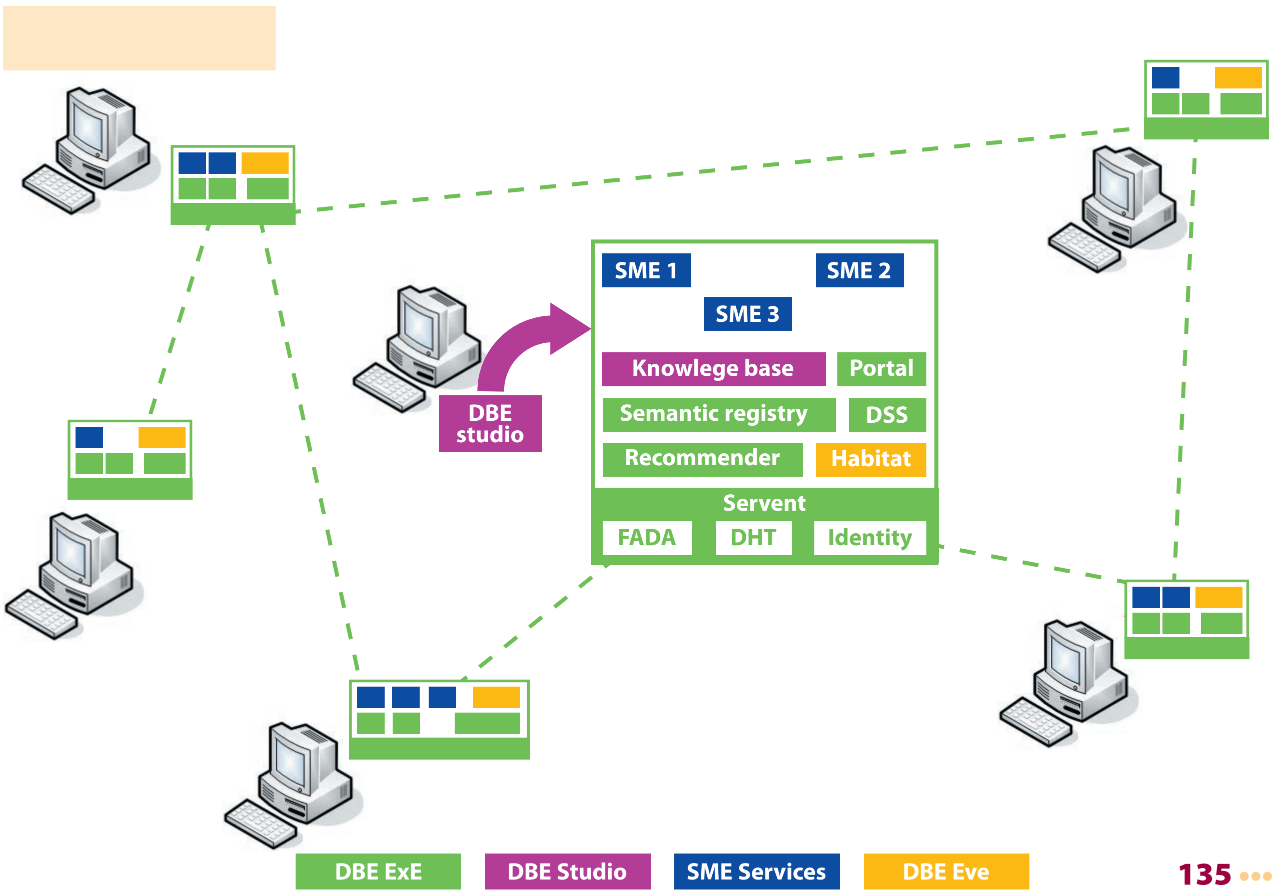
The service adapters are created with the Service Factory, after the service modeling phases have been performed. The Service Factory also deploys the created service adapters onto a user-specified servent.

From an end-user point of view the Service Factory is centred on the Eclipse* based DBE Studio [3] development environment. It includes many editors and tools to support the modelling, implementation and deployment of SME services. However, to allow ontologies and models to be reused, a distributed model repository known as the Knowledge Base has been implemented. Rather than require a dedicated infrastructure, the Knowledge Base is also hosted by the ExE as a distributed infrastructural service.

The EvE [4] – the optimisation engine of the DBE – is designed to monitor the consumption of services on the DBE and over time pre-position pointers to services onto nodes from which they are more likely to be consumed. Thus, queries will produce more useful results faster. It has also been designed to support the automatic composition of services, referencing existing atomic (or indeed composed) services inside entirely new workflows and deploying these new workflows as new services to help satisfy potential user needs. The EvE has been realised by the implementation of Habitats in which pointers to real SME services can be examined and manipulated. These Habitats are another distributed infrastructural service hosted by the ExE.

The key distributed infrastructural services of the DBE are illustrated in Figure 1. The ultimate purpose of these services is to support the deployment of the SME services, the raison d'être of the DBE.

Fig. 1
Distributed Infrastructural Services of the DBE



Although implemented as Core Components rather than pure DBE services as such, the peer-to-peer networking and identity layers of the DBE also deliver fundamental, distributed functionality to the DBE and so are also considered in this paper. These distributed infrastructural services and components are now described in further detail.

Infrastructural Services

Knowledge Base

The Knowledge Base [5] service, implemented by the Technical University of Crete (TUC), provides the distributed storage facilities that enable the Object Management Group Model Driven Architecture* (OMG MDA*) inspired design of the DBE [6], [7]. In it both XMI-formatted metamodels (models that describe models) and models can be persisted, queried and retrieved.

Due to the distributed and dynamic nature of the SME-based network, the Knowledge Base replicates data following a primary/secondary asynchronous model in which one node is always the primary node for a particular piece of content. Should it fail, a secondary node becomes the primary. Content is replicated from primary to secondary nodes asynchronously.

To improve the efficiency of queries, content is stored on nodes that already include semantically similar data. This is achieved by comparing the ontologies and semantics referred to inside the models. Initial queries from a node are propagated to all nearby neighbours, but nodes store information regarding which nodes the results came from and over time a comprehensive set of routing information is built up to help direct future queries more efficiently.

The actual raw data in the Knowledge Base is persisted in a native XML database, Oracle Berkeley DB XML [8], which can be installed to run on the MacOS X*, Windows*, Linux* and Solaris* operating systems. To support arbitrary updating of content, a versioning system has been implemented that can accommodate both the distributed nature of the Knowledge Base and the replication scheme.

The Knowledge Base service is used by the DBE Service Factory to save and share Ontologies, Business Modelling Language (BML) models, Semantic Service Language (SSL) models and Service Description Language (SDL) models. These models are typically used at design time when creating or modifying a service inside the DBE Studio. The Knowledge Base can, however, be used to store arbitrary XMI-formatted data in the DBE, and the ExE, for example, uses the Knowledge Base to store User Profile information.

Semantic Registry

Whilst the Knowledge Base service largely stores models, the Semantic Registry [5] service, also developed by TUC, is used to store published Service Manifests. A Service Manifest is an XML document that completely describes an individual DBE service. It can be considered to be an advertisement for a service on the DBE. It typically includes copies of the BML, SSL, SDL and BML data for the service, as well as additional configuration information.

Due to the similarity with requirements of the Knowledge Base in terms of data format, distribution, redundancy and performance, the Semantic Registry service shares many implementation components (and features) of the Knowledge Base service.

In the DBE, the Semantic Registry now supports the Service Manifest 2.0 specification. Service Manifests are published into the Semantic Registry Service by the DBE Studio, and are mapped to actual service proxies by the Peer-to-Peer layer.

The Semantic Registry is essentially the distributed service-directory for the DBE ExE. The contents of the Semantic Registry are accessed whenever a user searches for a service in the DBE, for example when using the Query Formulator / Semantic Discovery Tool incorporated in the DBE Portal.

Distributed Storage System

The Distributed Storage System [9], implemented by Intel, delivers a generic distributed storage capability to the DBE. Essentially it allows arbitrary content to be persisted onto the DBE peer-to-peer network, and generates an identifier by which the content can later be retrieved from any node on the network. For redundancy, the content is replicated. To avoid the distributed system overfilling with content, all content must be assigned a time-to-live by the entity storing the content.

This time-to-live can be reset by the same entity that stored it. After expiry of the time-to-live, the content will be automatically purged. Content may be secured by encrypting it before persisting.

By default, the DSS uses the DBE DHT core component (introduced later in this paper) to index the location of individual blocks of content. However, the index connector can be easily swapped out should alternatives be required, e.g. for local testing purposes. Alternative indexing mechanisms including indexing using the local disk store, a centralised PostgreSQL* database and a dedicated DSS Indexing DBE service have been implemented.

In terms of storage, the DSS persists content onto the local hard disk of the machines on which the DSS service is instantiated. However, the storage layer is also designed to be swappable, and alternative storage layers could be developed in the future to allow content to be saved in a database, or in another internet-based storage system, for example.

For performance reasons, blocks of content of an excessive size are partitioned into smaller blocks before storing. When retrieving content, the blocks are all copied to the node generating the request, these replicas then being available for future data requests.

To cope with the disappearance of nodes, background processes are used to monitor the quantity of duplicates of the blocks, and replicate them should this number get dangerously low.

For applications that require content to be given particular identifiers, e.g. filenames, a namespace can be overlaid on top of the DSS. For example, for file system functionality a dedicated file system service can (and has) been implemented which uses the DSS to persist the actual content.

DBE Portal

The DBE Portal [10] is a core service, also implemented by Intel, which provides a user-friendly HTML interface to the DBE. Typically, each SME has one portal hosted on their servent. This Portal consists of a completely arbitrary website representing the SME's business. It includes links pointing to the DBE services which that SME has deployed, as well as the ability to search for arbitrary DBE services. DBE Portals can also link to local DBE administration interfaces allowing basic servent configuration and functionality to be administered via the web.

To allow Portals to themselves be searched for, the Portal includes self-registration functionality which automatically publishes the existence of the Portal service within the DBE's Semantic Registry. Ultimately, this enables a peer-to-peer network of DBE Portals to be formed. If the IP address of the SME is static, or if they have registered an internet domain name, their DBE Portal can also be accessed directly over the internet using this address.

Recommender

The Recommender service [5], also implemented by TUC, is an autonomous system that uses preconfigured user profile information to identify the best-matching Service Manifests published on the Semantic Registry that may be of interest.

This ranked list of recommendations can be returned when explicitly requested by a querying application, or alternatively it can dynamically notify client applications when an update to the recommendation list is made. Thus, for example, a user whose profile explains that they are interested in low-cost flights could be automatically alerted when a new low-cost flight booking service is published in the Semantic Registry.

Habitat

The EvE is implemented in the Habitat service, designed by Imperial College London / Heriot-Watt [11] and implemented by Salzburg Technical University / London School of Economics [12] and Intel [13]. Although designed to support features such as autonomous service composition, the initial implementation uses neural networks to identify services that closely match those that have already been invoked. By clustering pointers to similar services, the DBE will be able to give better results faster in response to user queries.

Fig. 2

Screenshots from the DBE Portal template

Every time a service is deployed on the DBE, a reference to the service is inserted in the "Local Service Pool" of the local Habitat service. The Distributed Intelligence System then applies Neural Network algorithms (implemented using the open-source JOONE* engine [14]) to compare the deployed service to other deployed services it has references to.

If a similar service is identified, the usage history of the similar service is examined, and the newly deployed service pointer is migrated to the Habitat services on the nodes where the similar service was consumed.

In this way, pointers to similar services get migrated towards nodes that have consumed similar services. Should the user make a similar request in the future, then a query tool or recommendation algorithm that interrogates the Habitat service will be able to identify these potentially more useful services faster.

Core Components

FADA

FADA [15] stands for Federated Autonomous Directory Architecture (http://fada.sourceforge.net). An open-source project that originated in the European Commission funded Fetish project and now maintained by TechIdeas (http://www.techideas.es), FADA was the first peer-to-peer infrastructure embedded into the DBE.

FADA nodes find each other either using broadcasting on a LAN or via manual configuration and provide a location to store and retrieve proxies to services.

If the FADA node does not have the requested proxy, it queries its neighbours and they their neighbours until the requested proxy (or proxies) are found. To avoid indefinite queries, maximum query times and number of hops can be specified.

As machines are switched on and off, and the services on them become available then disappear, it is important to prevent the directory of proxies from filling up with proxies to services that no longer exist. FADA achieves this by using a lease mechanism. The node with the service registers the proxy to this service in FADA for a certain, relatively short, amount of time. Before this lease expires, the node re-registers the proxy. If for some reason the node does not re-register the proxy, the lease expires and FADA removes the proxy from the system.

Whilst DBE uses FADA as a registry for service proxies, FADA also provides searching facilities whereby service proxies can be assigned tags known as “entries”. FADA can then be queried to return not just proxies to specific services, but proxies to all services that have been assigned certain entries too.

Distributed Hash Table

The Distributed Hash Table (DHT) is the realisation by Trinity College Dublin (TCD) of one of the peer-to-peer overlay networks described in their DBE P2P architecture design [16]. Built on top of the open source Bamboo project [17] which is maintained by University of California Berkeley in association with Intel Research Berkeley, the DHT essentially provides for arbitrary distributed hash tables to be layered on top of the DBE network. These tables can store multiple values for each hash entry.

DHTs employ highly efficient lookup algorithms to locate the values for a particular entry, and each query is typically routed through no more than O(log2 N) peers. The DHT has redundancy built-in, with entries replicated onto a configurable number of logical neighbours. Just as in FADA, stale data is purged by means of setting a time-to-live. Although the DHT cannot support alternative search mechanisms like FADA, it does guarantee that if an entry exists, it will be found.

The DHT provides the ability for content to be removed as well as added. To prevent unwanted deletion from the table, the node storing the entry can provide a key, which must be provided if the entry is to be edited.

Identity

The Identity core component, also implemented by TCD, provides a customised overlay on top of the DHT that allows identity certificates to be stored, and various related operations to be invoked. By building on the DHT, the Identity system automatically becomes decentralised and inherits the redundancy and autonomous management features of the DHT. In particular, certificate revocations are automatic, thanks to the time-to-live functionality of the underlying DHT.

As well as providing a distributed key store, the Identity core component provides additional functionality including the ability to verify keys. All incoming DBE calls to the servent can be intercepted and any identities associated with the call can be automatically verified. Depending on the servent configuration, calls with no identity associated with them at all can either be halted or allowed to pass through.

The Identity core component is currently based on the web-of-trust model. However, it has been architected in such a way that alternative algorithms can be implemented and enabled via configuration parameters.

Conclusion

The Execution Environment, Service Factory and Evolutionary Environment of the DBE all rely on distributed infrastructural services including the Knowledge Base, Semantic Registry, Distributed Storage System, DBE Portal, Recommender and Habitat. Additionally, distributed core components provide fundamental peer-to-peer functionality that connect DBE nodes to each other and provide for service proxy lookup and identity management. These distributed infrastructural services provide key functionality without which the DBE could not function.

Acknowledgements

Nikos Pappas, Fotis Kazasis, Technical University of Crete; Andrew Edmonds, David McKitterick, Intel Ireland; Claudius Masuch, Salzburg Technical University/London School of Economics; Juanjo Aparico, TechIdeas; Dominik Dahlem, Trinity College Dublin.

References

1 DBE Deliverable D21.2: Architecture Scope Document. P. Ferronato – Soluta.NET, available at http://www.digital-ecosystem.org

2 DBE ExE: http://swallow.sourceforge.net

3 DBE Studio: http://dbestudio.sourceforge.net

4 DBE EvE: http://evenet.sourceforge.net

5 DBE Deliverable D14.5: Final P2P implementation of the DBE Knowledge Base and SR, N. Pappas, G. Kotopoulos, Y. Kotopoulos – Technical University of Crete, available at http://www.digital-ecosystem.org.

6 Object Management Group (OMG): http://www.omg.org

7 Model Driven Architecture* (MDA*): http://www.omg.org/mda/

8 Oracle Berkeley DB XML: http://www.oracle.com/database/berkeley-db/xml/index.html

9 DBE Deliverable D24.7: Distributed Storage System, J. Kennedy, R. Lee – Intel, available at http://www.digital-ecosystem.org.

10 DBE Deliverable D26.7: DBE Portal Specification, A. Edmonds – Intel, available at http://www.digital-ecosystem.org.

11 DBE Deliverable D6.4: Intelligence, Learning, and Neural Networks in Distributed Agent Systems, G. Briscoe, P. De Wilde – Imperial College London, available at http://www.digital-ecosystem.org.

12 DBE Deliverable D9.2: Report on Evolutionary and Distributed Fitness Environment, T. Kurz, G. Marcon, H. Okada, T. Heistracher – Salzburg Technical University, A. Passani – Censis, available at http://www.digital-ecosystem.org

13 DBE Deliverable D6.7: Distributed Intelligence System, D. McKitterick – Intel, available at http://www.digital-ecosystem.org.

14 JOONE: http://www.jooneworld.com/

15 FADA: http://fada.sourceforge.net/

16 DBE Deliverable D24.3: DBE Peer-to-Peer Architecture Design – Trinity College Dublin, available at http://www.digital-ecosystem.org

17 Bamboo: http://bamboo-dht.org/

* Other names and brands may be claimed as the property of others.

7 A Trusted Negotiation Environment for Digital Ecosystem

Luigi Telesca
CREATE-NET
http://www.create-net.org
luigi.telesca@create-net.org

Hristo Koshutansky
E.T.S.I. Informatica, University of Malaga
http://www.lcc.uma.es
hristo@lcc.uma.es; hristo@koshutanski.net

Digital Ecosystems have emerged as a novel paradigm to support the endless evolution of Networked Organizations. Existing tools and platforms enabling business collaboration and contracting are often developed and owned by large companies and big market bodies and, hence, centrally controlled and not fully trusted by small and medium size enterprises (SMEs). Thus, there is a pressing need for a trusted and affordable distributed technological environment supporting the creation of Virtual Organizations with common business goals and facilitating the creation, stability and improvement of business ecosystem performance on a more reduced time frame. The chapter proposes a multidiscliplinary trust framework based on current security technologies and reputation mechanisms. A novel research concept of evolutionary trust is identified that reflects the constantly evolving social institutional relations.

Introduction

Now-a-days organizations live in a highly competitive business environment where the availability of low cost broadband services change the way companies operate and behave in the global market. The recursive use of transitory structures based on alliances, partnerships and collaboration is required to overcome local market limitations and to pursue global opportunities. Businesses need a converging and trusted technological environment through which they can cooperate and create alliances to pursue business opportunities and growth.

Current negotiation platforms, such as Business-to-Business electronic marketplaces and Internet trading platforms have been developed in order to help the formation of virtual organization. Current solutions are centrally managed, normally developed and owned by large companies or big market integrators, therefore they are not fully trusted by SMEs and/or too expensive and hence not widely used by European SMEs today.

A new generation of distributed platforms and services are required to support the evolutionary and dynamic networked organizations overcoming the above-mentioned problems. Flexible technologies are needed to enable enterprises to efficiently cooperate in the digital world towards the creation of Digital Ecosystems (DEs).

The DE concept has emerged worldwide as an innovative approach to support the adoption and development of ICT. Inspired from a business ecosystem, a digital ecosystem is a self-organizing digital infrastructure aimed at creating a digital environment for networked organizations. However, current DE technologies lack of suitable models for addressing properties of trust and identity management.

Inspired by the technological nature of DEs, this chapter will provide a comprehensive framework underpinning reliable and trusted ecosystems communications. The objective of the framework is to provide SMEs with a trusted and secure technological environment underpinning their business growth.

Chapter Contribution

This chapter defines the basic components of a generic negotiation environment that supports trusted and decentralized business contracting for DEs. Decentralization is faced by negotiations. Trusted negotiations are built on top of reputation models and existing security technologies. A new notion of evolutionary trust is introduced based on learning, reputation and social institutional trust.

Chapter Outline

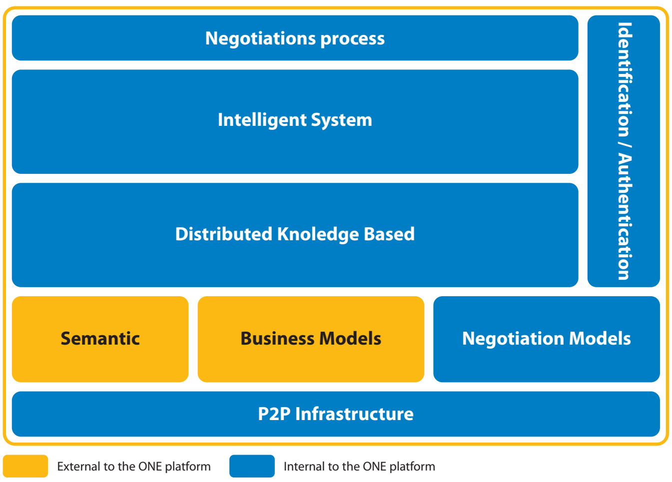
The chapter has the following structure. Section 2 introduces the concept of negotiation for decentralized business contracting as inspired by an ongoing EU project, called ONE. Next, Section 3 defines the role of reputation models for DE communication and the intuition for evolutionary trust. Section 4 overviews current security technology and models underpinning a reliable and trusted negotiation environment. Section 5 concludes the paper and outlines future research directions.

Negotiation for Decentralized Contracting

As already defined in [16] "Negotiation is a process involving dealing and communication among two or more parties, who have different concurring (non conflicting) objectives, which intend to reach a reasonable compromise and a mutually accepted agreement on a given matter and commit to a course of action".

Business-to-Business electronic marketplaces ("B2B e-markets") have been developed to facilitate business transaction trough a common environment and supporting tools for all the parties involved in a negotiation process. Anyway current B2B e-markets are usually centralized, therefore not fully trusted by, and not able to deal with multi-party and multi-issue negotiations.

Fig. 1
ONE Architecture



In order to cope with business complexity a new approach has been defined to cope with distributed resources/services and decentralized aspects in Digital Ecosystems. The ONE[1] (Open Negotiation Environment) Project will develop an open environment supporting tactical negotiation and agreement processes among participants. This environment will support the creation of Virtual Organizations with common business goals and facilitate the creation, stability and improvement of the ecosystem performance on a more reduced time frame. At the same time it will have no central governance cockpit or console for managing negotiation models and ongoing processes in order to avoid the "big brother" effect, which will put at risk the level of adoption.

The ONE platform will dynamically support the negotiation owner through automatic learning techniques applied to the goal of learning the best negotiation strategies in a multi-agent environment. The hybrid [4] and conversational personal negotiation recommender will support a user and will carry out operations on his behalf with some degree of independence and autonomy. It will compute the recommended actions exploiting a distributed knowledge base, which expands the personal knowledge base of an actor, and makes possible to speed-up the policy learning process, exploiting experiences gathered not only by the supported user but also by a community of trusted partners [15]. ONE will support a model of collaboration and trust based on the idea of "collaborative multi-agent systems", where agents work and learn with other trusted agents and develop collaborative learning schemes.

With this intent we defined the concept of trusted negotiations in an ecosystem environment by:

- ► Securely identify partners in negotiations – provide privacy, anonymity and accountability during (part of) negotiations;
- ► Assess trustworthiness of partners based on their past behavior and your (agent's) own experience – provide proper reputation models supporting cross-domain reputation assessments;
- ► Facilitate trust relationship establishment across distributed ecosystems (e-communities) reflecting the constantly evolving business requirements over time.

Reputation Models for Trusted Ecosystems

Close to business contracting is reputation. Reputation assessments have a direct influence on a negotiation process and are strongly linked to the results of bilateral and multilateral contracting. Assessing in a measurable way the trustworthiness of partners in inter-ecosystem communication becomes a key issue for a trusted negotiation environment.

There are two main approaches to reputation referred in the context of agents. With the first approach, agents use trust models to reason about the reliability or honesty of their counterparts. With the second approach, agents calculate the amount of trust they can place in their interacting partners where the likelihood for an agent to be selected as an interaction partner depends on the calculated level of trust. Either of the trust models aims at guiding agents to decide on how, when and who to interact with.

To face the decentralization nature of the ecosystem environment peer-to-peer reputation mechanisms will be provided. Users of services own the best knowledge about the behavior of services based on their own experience. This experience can be translated and expressed as reputation statements [1,14,18].

In some commercial scenarios peer-to-peer mechanisms are not suitable or easily accepted and so the concept of trusted rating agencies[2] has to be provided. Here, partners use trusted agencies to reason on trustworthiness of other partners (service providers). On the other side, service providers subscribe to rating agencies to be included in their list of recommended services.

Evolutionary Trust

The key feature of an ecosystem is its evolution in state and time in order to adapt and respond to new conditions without being slowed down by human related factors. In this sense, an ecosystem should be empowered with a model for decentralized cross-domain trust relationship establishment.

To face decentralized trust establishment we have to look at how to facilitate joining to an online community. Current security models supporting IT digital businesses are concentrated on establishing trust between entities already in the network. But what occur when a new organization is joining an ecosystem? And what happen when an organization already active in one ecosystem is taking a role in another (new) ecosystem?

We need to borrow the concept of institutional trust [11] and analyze the collective behaviour of users when they deal with digital institutions. Institutions, professional or associations, public administration, to name a few, can provide trust to newcomers and affect their behaviour when communicating with other partners.

An ecosystem-driven system should provide additional learning mechanisms based on institutional trust. Trustfulness in one or more institutions (partners) can be initially obtained by examining institutional trust existing between those institutions and the known institutions by the partner. This will create an independent and evolutionary platform capable to adapt and evolve on the basis of the evolution in institutional trust.

1) www.one-project.eu

2) See for example http://www.dotcom-monitor.com

Therefore we identify a new research challenge: combining learning mechanisms with reputation and social institutional trust. As already mentioned in Section 2, some of the learning techniques can be found in [4,15]. The possible synergies will open new research topics complementing the concept of trust as advocated in computer security literature.

Security Technology for Trusted Ecosystems

This section provides an architecture and overview of security models and standards underpinning a reliable and trusted negotiation environment.

A trusted negotiation environment will provide authentication, integrity and confidentiality as basic security primitives. Existing cryptographic algorithms and protocols will be used and employed to achieve it. On top of them a set of APIs will be provided, generic and user friendly as well as design independent from the underlying cryptographic algorithms. The APIs will be easy to use and adopt while providing new algorithms to be plugged in the future.

Digital identities represent individuals' sensitive information and are used when individuals introduce with each other. Identity management becomes a bottleneck when negotiations cross different administrative domains.

There are a number of industrial approaches offering identity management solutions such as OASIS SAML[3], Liberty Alliance[4] and WS-Federation[5].

The key idea behind those is enabling a multilateral federation of partners sharing the same domain (circle) of trust. Each federation supports multiple identity providers and within a federation (circle of trust) a user may traverse all involve partners' services with a single authentication.

However, a proper identity management model that scales to the DE nature should go beyond a federation-based concept and rather provide:

- user-centric identity management: each entity will be the sole holder of its identity information,
- peer-to-peer or a hybrid (partially hierachical/federated) model of trust relationships between identity providers (authorities),
- brokering trust of identities and authentication information between different DEs.

Identity management goes hand-by-hand with privacy protection [12]. Pseudonyms are used to identify parties when negotiating with different ecosystem domains. Pseudonyms can be used to achieve different levels of anonymity. By shifting the creation and management of identities and pseudonyms to the end-entity, the model will benefit improved privacy protection (decentralized identity storage) and accountability: allowing users to remain anonymous while giving service providers strong guarantees about the users' accountability. Close to our needs is the work in [13].

Computer security trust has emerged as a major security issues over the last years[6]. The notion of trust management has vast meaning and definition as depending on the particular context. Referring to the settings of a trusted negotiation environment, we focus the notion of trust to the notion of distributed access control and decentralized access rights establishment.

The basic approach to distributed access control, underlying current systems and models, is the capability-based access control (see [5] for a comprehensive survey): rely on one's capabilities to take access decisions. The term credential has become widely used for expressing digital access rights (capabilities) and credential-based access control management has grown as the proper model for enforcing authorization requirements in a distributed setting [2,9,3,8].

A trusted ecosystem environment will approach decentralized access rights establishment via bilateral negotiations, also called automated trust negotiation [17,8]. Some of the related projects in this field are TrustBuilder[7] and iAccess[8].

3) OASIS Security Assertion Markup Language: http://www.oasis-open.org/committees/security

4) http://www.projectliberty.org

5) http://www.ibm.com/developerworks/webservices/library/ws-fed

6) See the iTrust Working Group at http://www.itrust.uoc.gr

7) http://cdr.cs.uiuc.edu/trustbuilder

8) http://www.interactiveaccess.org

Figure 2 shows the multidisciplinary framework underpinning a trusted negotiation environment. The right side column represents possible technology platforms suitable for ecosystem service execution management. The left side column represents the trusted negotiation environment that SMEs use to perform their business goals.

The horizontal layers comprise the whole range of security and trust issues discussed so far where higher the layer is, closer to the business-level management it is. As all the layers might be interconnected (e.g., security standards and protocols with evolutionary trust layer) for one or another security aspect so the dashed lines are used to symbolically distinguish each of them.

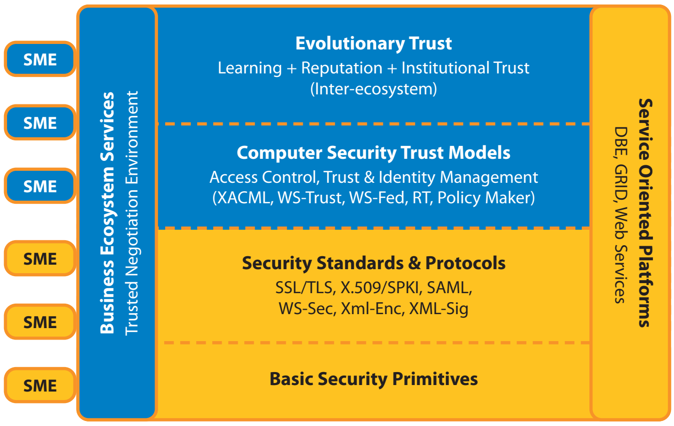


Fig. 2
A Multidisciplinary Framework for a Trusted Negotiation Environment

Conclusions and Future Work

The new generation of technologies underlying current business contracting platforms lack of suitable models for supporting reliable and trusted negotiations. Inspired by this, the chapter advocates a generic framework that underpins a trusted negotiation environment. Drafting the future research directions, the first step is to provide the trusted DE environment with those security technologies that are flexible and affordable by most of SMEs. The aim is to form a comprehensive open technical platform, we call it security middleware, that puts in practice the existing standards and protocols and, at the same time, provides easy adoption and extension of new technologies.

On top of the security middleware, the second main step is to provide a proper model for the concept of evolutionary trust. Again the aim is to incorporate and reuse (where possible) existing reputation mechanisms and institutional trust models into one comprehensive reputation framework affordable and easyly adopted by SMEs.

The final objective of the work is to provide SME's businesses with a trusted and affordable technological environment through which they can create tactical and strategic alliances and pursue business opportunities and growth.

Acknowledgements

This work was partly supported by the projects: 034744 EU-INFSO-IST ONE, 034824 EU-INFSO-IST OPAALS, 038978 EU-MarieCurie-EIF-iAccess.

References

[1] Abdul-Rahman A. and Hailes S. Supporting Trust in Virtual Communities, In Proceedings of the 33rd Hawaii International Conference on System Sciences. 2000. IEEE press.

[2] Bertino E., Catania B., Ferrari E., and Perlasca P. A logical framework for reasoning about access control models. In Proceedings of the Sixth ACM Symposium on Access Control Models and Technologies (SACMAT), 2001. ACM Press, pp. 41–52.

[3] Bonatti P., and Samarati P. A unified framework for regulating access and information release on the Web. Journal of Computer Security, 10 (3), 2002. 241–272.

[4] Burke R., Hybrid Recommender Systems: Survey and Experiments, User Modelling and User-Adapted Interaction, 12 (4), 2002. pp. 331-370.

[5] De Capitani di Vimercati S., and Samarati P. Access control: Policies, models, and mechanism. In R. Focardi and F. Gorrieri (eds.) Foundations of Security Analysis and Design - Tutorial Lectures, vol. 2171 of LNCS, 2001. Springer-Verlag Press.

[6] Guttman R. H., Moukas A., and Maes P. Agents as mediators in electronic commerce. Electronic Markets, 8(1), 1998.

[7] Kersten G.E. and Lo G. Aspire: an integrated negotiation support system and software agents for e-business negotiation. International Journal of Internet and Enterprise Management, 1(3), 2003.

[8] Koshutanski H. and Massacci F. Interactive Access Control with Trust Negotiation for Autonomic Communication. In D. Khadraoui and F. Herrmann (eds.), Advances of Enterprise Information Technology Security, 2007. IGI press.
[9] Li N. and Mitchell J.C. RT: A role-based trust-management framework. In Proceedings of DISCEX III Conference, pp. 201–212, 2003. IEEE press.
[10] Maes P., Guttman R.H., and Moukas A.G. Agents that buy and sell. Communications of the ACM, 42(3), 1999. pp. 81-91.
[11] Pavlou P. A., Tan Y.-H. and Gefen D. The transitional role of institutional trust in online interorganizational relationships. In Proceedings of the 36th Hawaii International Conference on System Sciences. 2003. IEEE Press.
[12] Pfitzmann, A. Anonymity, Unlinkability, Unobservability, Pseudonymity, and Identity Management – A Consolidated Proposal for Terminology. Tech. Report. 2005. Available from http://dud.inf.tudresden.de/~pfitza/Anon_Terminology_v0.22.pdf
[13] Poursalidis V. and Nikolaou, C. A new User-centric Identity Management Infrastructure (IMI) for Federated Systems. In Proceedings of 3rd International Conference onTrust, Privacy & Security in Digital Business (TrustBus), 2006. Springer-Verlag Press
[14] Sabater J. and Sierra C. REGRET: a reputation model for gregarious societies. In Proceedings of the 1st International Joint Conference on Autonomous Agents and Multi-Agents Systems, 2002
[15] Tan M. Multi-agent reinforcement learning: Independent vs. cooperative learning. In Huhns, M. N. and Singh, M. P. (eds), Readings in Agents, pp. 487-494. Morgan Kaufmann, San Francisco, CA, USA, 1997.
[16] Telesca L., Finnegan J., Ferronato P., Malone P., Ricci F., Stanoevska-Slabeva K. Open Negotiation Environment: An Open Source Self-Learning Decentralised Negotiation Framework for Digital Ecosystems. In Proceedings of the 1st IEEE DEST Conference, 2007.
[17] Winslett M. An Introduction to Trust Negotiation. In Proceedings of iTrust Conference, 2003. pp. 275-283.
[18] Witkowski M., Aritikis A. and Pitt J. Experiments in building experiential trust in a society of objective-trust based agents. Trust in Cyper-societies, 2001, pp. 111-132.

8 Social Network Simulation and Self-Organisation

Thomas Kurz
Salzburg University of Applied Sciences
http://www.fh-salzburg.ac.at
thomas.kurz@fh-salzburg.ac.at

Antonella Passani
CENSIS
http://www.censis.it
antonellapassani@gmail.com

Thomas J. Heistracher
Salzburg University of Applied Sciences
http://www.fh-salzburg.ac.at
thomas.heistracher@fh-salzburg.ac.at

Approach for Cooperation

Developing the concept of Digital Business Ecosystem (DBE), and implement it at local level trough regional engagement, integrate manifold areas of interest, namely Business, Computing, Science and Socio Economics (see Heistracher, Kurz, Marcon and Masuch, 2006). Consequently it represents a high challenge in communication and collaboration, different research agenda, different vocabularies and languages need to be compared, converged, translated. The concept of Digital Business Ecosystem, in fact, becomes part of different conceptual frameworks that, by the way, need to be constantly interlinked and efficiently connected.

The present article will describe a path of interdisciplinary collaboration that took place in the second year of the project and in which computer science played a pivotal role. Specifically, in the following, we present an effective means of collaboration by introducing a simulation framework called Evolutionary Environment Simulator (EVESIM) (Kurz et al, 2006) Two the main input for the collaboration above the EVESIM: the evolutionary algorithms developed in the science domain and the territorial social network that arose from social science field research. In principle, the development of evolutionary algorithms and the analysis of social networks could be performed independently, thereby, however, excluding any potential of mutual benefits. In this aspect, the EVESIM can be considered a kind of 'middleware' between the Natural Science and Social Science domain[1].

The EVESIM is the software simulation framework, which facilitates the communication between the Natural and Social Science "applications" that possibly base on similar meta-concepts. That does not mean that EVESIM solves all

1) According to Linthicum (2003), middleware is a software that facilitates the communication between (two) applications.

issues of communication but it is a starting point of how different areas of science can effectively collaborate and take advantage of each other. We discuss in the following the issues in the context of Social Science and Natural Science and, preliminarily, we describe in details the Evolutionary Environment Simulator itself.

Role of the Evolutionary Environment and EVESIM

The name Evolutionary Environment Simulator comes from the initial intention to set up a simulator of the so-called Evolutionary Environment in the DBE project (Heistracher et al, 2004). The Evolutionary Environment is a network of DBE nodes and services which enable the self-organisation of the DBE network and provide a test bed for various research topics like natural language business modelling (OMG, 2006), evolutionary algorithms (Colin, 2002) and distributed intelligence (Briscoe and De Wilde, 2006). For more information on the Evolutionary Environment see also (Masuch, 2006).

Although the name Evolutionary Environment Simulator results from this particular Evolutionary Environment, the intention of the EVESIM is not only to simulate the behaviour of the Evolutionary Environment, but also to provide partners from Natural Science, Social Science, Business and Computing a framework to collaborate and test their findings together. During the ongoing collaboration in the past, the EVESIM emerged to be a generic framework for simulating self-organisation and SME networks for a broad audience from different research domains.

The approach of choice for communication and collaboration was to meet the needs of the different partners and to avoid influencing their very particular way of working as long as possible. Therefore, generic interfaces had to be found and a couple of transformation modules, import and export capabilities had to be added.

Specifically for Natural Science stakeholders, a plug in mechanism was developed to use both the evolutionary algorithms developed especially according to the EVESIM model and the evolutionary algorithms with binary representations. Through a transformation module from binary representation to the representation of SMEs and services according to the EVESIM model, additional optimisation algorithms can be added and evaluated in their usage in a DBE. More details about the model used in the EVESIM can be found in subsection 1.7.3. Furthermore, an XML-based import mechanism enables importing real-world business network data during runtime.

Specifically for Social Science stakeholders, the EVESIM provides import capabilities for Comma Separated Files (CSV). That enables non-technically experienced people to export data from any spreadsheet software for subsequent import into the EVESIM. Moreover, the configuration of actors along seven predefined 'social variables' influences the behaviour and set-up of the agents in the simulation. These variables are described in the following.

Natural Science

To imitate Digital Business Ecosystems the real-world behaviour has to be simulated which is achieved by using evolutionary algorithms, well known from the study of life as explained in section 1.1 "Natural Science Paradigms". Evolutionary algorithms are used to find an optimum solution for different types of problems. In the case of the EVESIM, the challenge is to find the best-fitting service for a specific task of a SME. Thus by using evolutionary algorithms the self-organizing features of natural ecosystems are utilized to simulate and enhance business networks.

Furthermore, it is possible to check the effects of different social and business parameters onto the ecosystem. To achieve this, the individual SMEs in the ecosystem are simulated by independent software agents[2]. These agents can interact and individually adapt to the changing business needs. The possibility to adapt dynamically to a changing ecosystem in a self-organizing way is the major advantage of utilizing biological approaches in the Digital Business Ecosystem. Therefore evolutionary algorithms are the fundamental optimisation mechanism of the EVESIM.

As was mentioned in section 1.1.5, it is hard to predict how a real-world ecosystem will evolve. This is true for a simulated ecosystem as well. But by utilising a simulator it is possible to find out key parameters influencing the evolution of an ecosystem. One of these key parameters is the critical mass of participants that is needed to get the ecosystem work as detailed in (Kurz and Heistracher, 2007). As research on evolutionary algorithms, for example, is often done on random high-scale networks (Colin, 2002) the availability of real-world data from Social Science would be highly beneficial to make simulations more 'close to reality '.

2) In the context of EVESIM, the terms agents, SMEs and actors are used interchangeably.

The input of social science in this sense is mainly correlated to the concept of *social capital*; intended in its broad sense of relational and business territorial networks. That of social capital is, in fact, one of the theoretical approach social science researchers choose for interpreting the DBE community building process. From this specific point of view the simulator can be understood as an instrument for visualize, in a dynamic way, ongoing process and as a tool for validate different hypothesis on the capacity of DBE to boost territorial social capital by improving the level and the quality of collaboration among SMEs and other local actors.

Social Science

Researches carried on by social scientists in the DBE consortium have been focused not on technology itself – considered as an independent factor of business attitude - but on the correlation between technological innovation and existing social relations. A key question was represented by the possibility for DBE to reinforce already existing business and social relationships and/or create new links among local players in this way contributing to improving the territorial social capital, i.e. the level and quality of collaborations among local players. The main methodology used for exploring this research's topic has been that of Social Network Analysis (SNA) The EVESIM come into play after the first network analysis research, as an useful tool for improving results visualisation and multivariable analysis.

Before describing the concrete convergence between social science research and computer science domain trough EVESIM, it seams interesting to briefly introduce the theoretical framework upon which the Social Network Analysis has been based. In fact, it generate by on of the main goal of the DBE project, i.e. to sustain European SMEs by offering them a process and a technological solution for clustering.

When analysing results from a range of different researches, it emerges clearly that the capacity to collaborate and take advantage of social capital is a decisive factor in the diffusion of innovation within a given local production system and in its SMEs. SMEs collaboration and cluster is a well know catchphrase in the innovation debate, however, the latest research carried out by Censis indicates the pressing need to abandon the use of slogans and focus, instead, on the various levels of collaboration, highlighting which models they give rise to and which benefits they can bring to companies implementing them. An approach of this type makes it possible to analyse the concept of collaboration more systematically, highlighting the way in which SMEs are still too often involved in so-called 'limited-horizon collaborations' that are implemented through the use of shared services, through participation in trade fairs and by accessing shared credit services. We use the term 'limited-horizon collaboration' to underline how this type of initiative - even when formalised and persistent over time - does not face up to the problem of company development in project terms. This model can guarantees economic benefits in the short term but should not be considered suitable as a facilitator for product or process innovation. DBE has been seeing as an instrument for open up new collaborative process, with a wider horizon.

The advantages of collaboration, in fact, increase in proportion to two factors:
- The centrality of the corporate functions engaged: what is being collaborated on?
- The heterogeneous complexity of the network: who is the collaboration between?

In other words, the advantages for companies increase as they move from collaboration on support functions to collaboration on strategic functions (R&D, marketing, internationalisation, and so forth) and as they open up their networks to university, research centres, intermediate actors as Chambers of Commerce and Development agencies an so on. DBE – thanks to its flexible architecture – can easily adapt to different territorial characteristics and include different local actors accordingly to their missions and SMEs real needs and by so doing could become a collaboration facilitator. In order to evaluate in which grade this is not only possible in theory but also already observable in practise, Censis carried out two different surveys on existing networks and present territorial social capital using network analysis methodology[3].

The role of simulator, here, is that of visualizing and making dynamic data that are normally only static. The simulator has been used in order to visualize the growth of the already existing territorial networks during the process of SMEs recruitment. It make possible to picture those networks on which DBE can rely on, individuate missing links, and give in signs to the SMEs recruitment strategy adopted. Evaluating the networks in terms of social capital is essential for at least two reasons:

---

3) An initial definition of social capital is required here in order to understand the rapid conversion from social capital to networks. In accordance with Bourdieu, we may define social capital as "the sum of resources, actual and virtual, that accrue to an individual or a group by virtue of possessing a durable network [...] of mutual acquaintance and recognition" (Bourdieu, 1980:22).

1. The networks, being relational infrastructures between actors, are, invariably, a useful way of defining the context in which those actors operate, and describe – at the same time - the actor's characteristics.
2. Describing how the network is composed can help the consortium to understand which are the most important actors that should be included in the DBE in order to make the ecosystem grow and reach the critical mass needed to be self-sustaining.

An important element when studying territorial networks is that of group characteristics. In this regard, the research explored various possible types of contacts that can be considered as different types of collaboration. Possible relationship were as follows: personal contact; participation in associations or institutional bodies; participation in projects; sharing of resources; information exchange; and no contact, meaning "I am aware of their existence but have no contact with them"[4].

By diversifying the types of contact, we were able to conduct important research into:

- Formal contact vs informal contact
- Intensive relationships, i.e. highly focused collaboration projects vs extensive collaboration (sharing of information and/or resources)
- Presence or absence of subgroups and types of subgroup: associations, working groups, clusters

Thanks to network analysis first and thank to the simulator in a second step, all those information take the form of relational networks. Interviewees were given the opportunity to provide more than one answer for each relationship, meaning that SMEs representatives may indicate different types of contacts for the same actor. Overlaps of this nature, when they occur, are very interesting because they can function as a tool with which to measure network density.

Indeed, as Portes has stated, "an intrinsic characteristic of social capital is that it is relational. Whereas economic capital is in people's bank accounts and human capital is inside their heads, social capital inheres in the structure of their relationships. To possess social capital, a person must be related to others, and it is these others, not himself, who are the actual source of his or her advantage" (Portes,1998). In short, social capital exists only when it is shared. But is not simply a matter of the extent to which people are connected to others, but the nature of those links. Social capital benefits grow together with the grow of network density. While social capital is relational, its influence is most profound when the interaction occurs between heterogeneous clusters, as we have mentioned the "who is the collaboration between?" is a key question. From an economic perspective, several recent studies conducted as part of the World Bank's Local Level Institutions Study (Grootaert and Narayan, 2000) confirm the importance of heterogeneity in group membership and economic outcomes. From another prospective, Florida also confirmed that the dimension of diversity is strongly connected to the innovation level of a given group or region. In these studies, the capacity of a group to include a high level of diversity comes across as crucial, since a high "level of tolerance", as the author puts it, makes is easier for that group to innovate and, consequently, become more competitive. Making further reference to the metaphor of the ecosystem, it may be said that biodiversity is one of the most important conditions for sustaining the life of the system. In light of this, we introduced the question of diversity. We asked participants to grade the level of diversity in their workplaces, in order only, at this stage, to help us build up a snapshot of SMEs from this particular perspective. The interviewees were asked to consider a variety of factors such as differences in levels of education, wealth, social status, gender and ethnicity, age group, party/political affiliation or religious beliefs and length of residency. In addition to the internal level of diversity described above, the level of network diversity (i.e. the number of actors with which SMEs interact and the 'nature' of those actors) is also important.

All the above-mentioned network characteristics have been introduced in the simulator and constitute what we called Territorial Social Capital.

In recent years, some scholars have proposed an additional conceptual classification. Called "linking" social capital (Woolcock, 2001), this dimension refers to a given individual's ties to people in positions of authority, such as representatives of institutions, public (police, political parties) and private (banks) alike. Whereas the operation of bridging social capital is, as the metaphor implies, essentially horizontal (that is to say, it connects individuals of more or less equal social standing), linking social capital is more vertical, connecting individuals to key political (and other) resources and economic institutions - in other words, across power differentials. Importantly, it is not the mere presence of these institutions (schools, banks, insurance agencies) that constitutes linking social capital, but rather the nature and extent of social ties between such different actors. Defined as such, access to linking social capital is demonstrably central to producing economic wealth.

4) The social network analysis has been based upon face-to-face interview to Regional Catalysts and engaged SMEs.

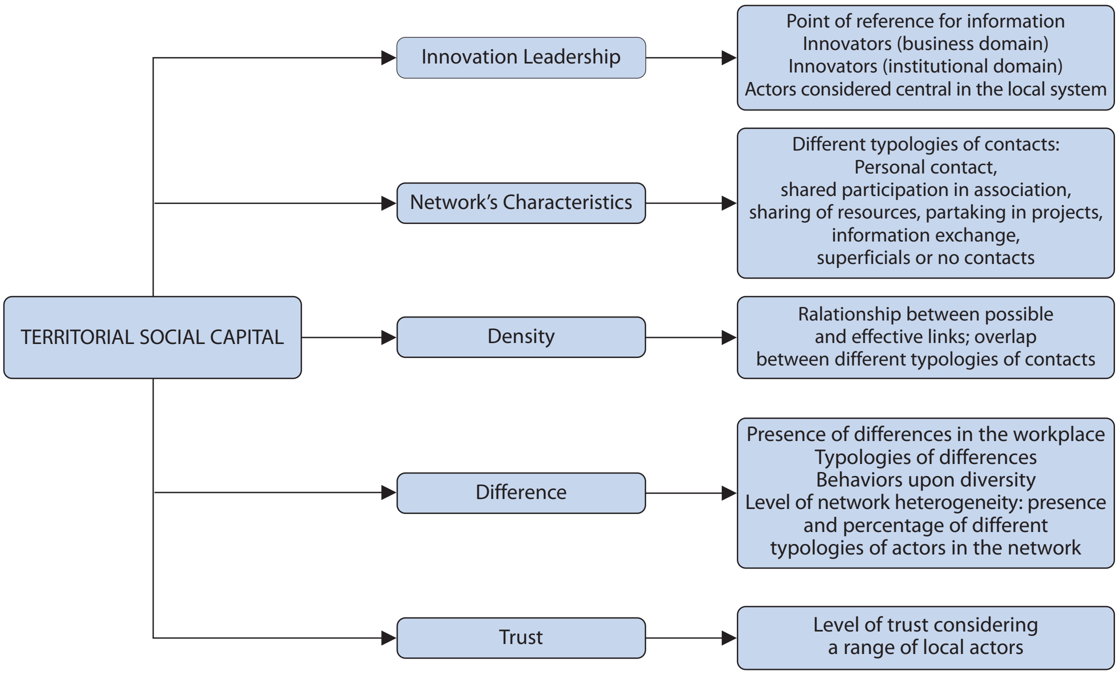


Fig. 1

Territorial Social Capital Definition
Source: Censis, 2006

The survey also explores respondents' subjective perceptions of the trustworthiness of key institutions that shape their lives as a crucial dimension in the potential for collaboration, and this is closely related to the concept of linking social capital discussed above: reciprocal trust is a precondition for collaboration and is the 'glue' that makes it possible to engage with the risks and benefits of long-term projects such as DBE.

Social Networks present and future dimensions

As we already mentioned, Censis carried on a first survey in the associated regions about RC and Driver SMEs' relational networks. Data gathered in Aragon have been the starting point for the collaboration with computer science specialists and the simulator adjustment to social science needs. In this first survey 7 typologies of relation were take in consideration:

- personal contact
- share of information
- share of resources
- partaking in projects
- participation in association or institutional bodies
- superficial recognition
  (I know them but I have not contact we them)
- no contact

In the simulator to each connection typologies correspond a different grade of strength that impact of the network grow rate. In the next research, when more SMEs will be interviewed (drivers, as well as implementer and users) we whish to be able to introduce in the simulator more variables, mainly correlated to SMEs economic characteristics (size, sector, turnover, N. of clients and providers and so on) but also related to their approach to innovation, ICT and collaboration and we'll try to simulate possible impacts of those variables on the network growth rate, service migration rate and connectivity rate.

By introducing those variable the simulator will acquire a new use for social science. Beside the possibility to visualize in a dynamic way static data, it will be also an interesting instrument for training and communication. By modifying each SMEs characteristic, in fact, it would be possible to visualize the outputs in term of collaboration pats and related business benefits. Introducing those new variables will imply a modification of the simulators and will require more research from both side, that of social science and that of computer science.

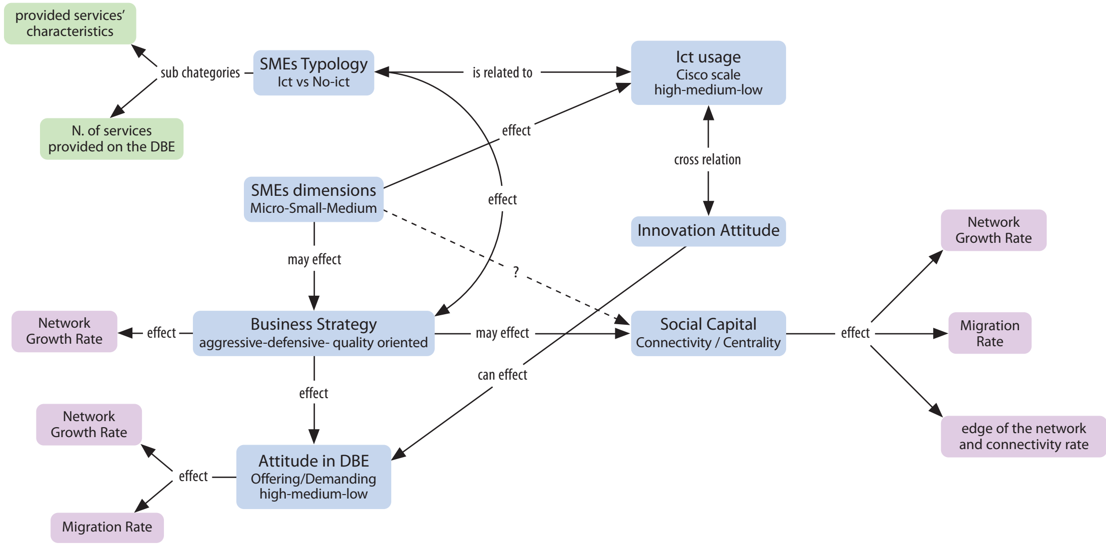


Fig. 2

Possible variable for future simulator development Source: Censis, 2006

Evolutionary Environment Simulator - EVESIM

In the following, we discuss the technical implementation of the EVESIM. The according source code can be found at (Kurz et al, 2006). To keep the simulations as realistic as possible while attaining the goal of speeding up the process of evolution, a number of tools were used to simulate the DBE system. The EVESIM tackles the goal of having a system where the network nodes remember past interactions with different other nodes and services to continually improve the system in a smooth way. Moreover, the EVESIM provides a simulation framework with rich configuration and visualisation capabilities for being applicable for different digital ecosystems during future research (see Fig. 3).

The implementation of the simulator itself required the collaboration of many different disciplines. The EVESIM stands to benefit from the input of partners concerning genetic algorithms, global optimisations, symbiosis and competition, social networks as well as software engineering. These groups, consequently, can utilise the EVESIM sources. By adding code and features to this project it became a cross-domain collaboration platform.

The results of the simulations, though, do not claim to be a one hundred percent realistic. The intention of the cross-domain collaboration is to make the results more realistic and the EVESIM provides a test bed for this endeavour. Moreover, by restricting the variables used in a system, the disciplines can run first there simulations on a restricted area, e.g. high scale networks for genetic algorithms research, and then afterwards apply the algorithms to a more realistic and customized network structure.

Although the EVESIM model is intended to be as close as possible to reality, the model represents an abstraction layer, which enables the simulation of the behaviour of small real-world networks as well as the simulation of well-defined problems in high-scale networks. The representation of SMEs and especially of service descriptions within the EVESIM are an abstraction of Semantic Business Vocabularies and Business Rules (SBVR) and therefore a mapping of SBVR logic into a set of features (flattening), which results in a simplified model that does not take into account the full set of SBVR capabilities. SBVR is a natural language approach for business modelling (see MOG, 2005 and OMG, 2006). Nevertheless, this model is a compromise between the real SBVR representation and the abstraction level that facilitates a simulation that is close to reality. Additionally, the matching of SBVR models and its theoretical implications are still in research status. Consequently, a level of abstraction has to be found so that a generic objective function can be defined, capable of being applied to a broader set of service descriptions (potentially any version of business modelling language).

As delineated as A in Fig. 3, each service is represented by a number of attributes. These attributes can be symbolic (colour of a car) or numeric (discount for a price). As symbolic attributes can be simplified by using natural numbers, the range of attributes could be chosen as real numbers for both, symbolic and numeric values. As SBVR describes models and as the search will also be on the basis of models, real numeric attributes are not the main focus. Therefore, the values of attributes within the EVESIM are currently set as a subset of natural numbers. In case of service combinations, the attributes of the individual services are merged and consequently construct a new service description, e.g. a word processor consisting of word processing, thesaurus and spell-checker. The comparison of two services is a comparison of all the attributes of one service to all the attributes of another service and service combination, respectively. When a new service is produced it appears first in the portfolio of its producer SME. The producer SME is presented by the actor, which produces or offers a service. In case of a service combination of two existing services, the actor who combined the service becomes the owner of the new service. This is why we assume that additional effort was needed to combine existing services. From user perspective for example, a travel agency is the owner of the travel-service, though it merely books the corresponding flights, the airport transfers and the hotel.

The social network analysis within the DBE currently uses a SME table for retaining the relationships between SMEs. Rows as well as columns hold the names of the SMEs. The type of relationship is represented as the value in the intersection of axes. As to provide a common import from a broad range of spreadsheet software, the import files for the EVESIM have to be CSV (Comma Separated Values) using a semicolon for separation.

For visualising the capabilities of the Evolutionary Environment, the actors, services and the whole network topology can be displayed through the EVESIM Display (see Fig. 4). For each type of actor, a picture label can be chosen from the file system to indicate the different actors in the network. The edges between the actors represent the bidirectional relationships of two actors. Beside the visualisation of the network, a label for displaying the gross Network-Fitness was introduced. The algorithm for calculating this network fitness as well as other parameters can be easily modified according to the users' needs.

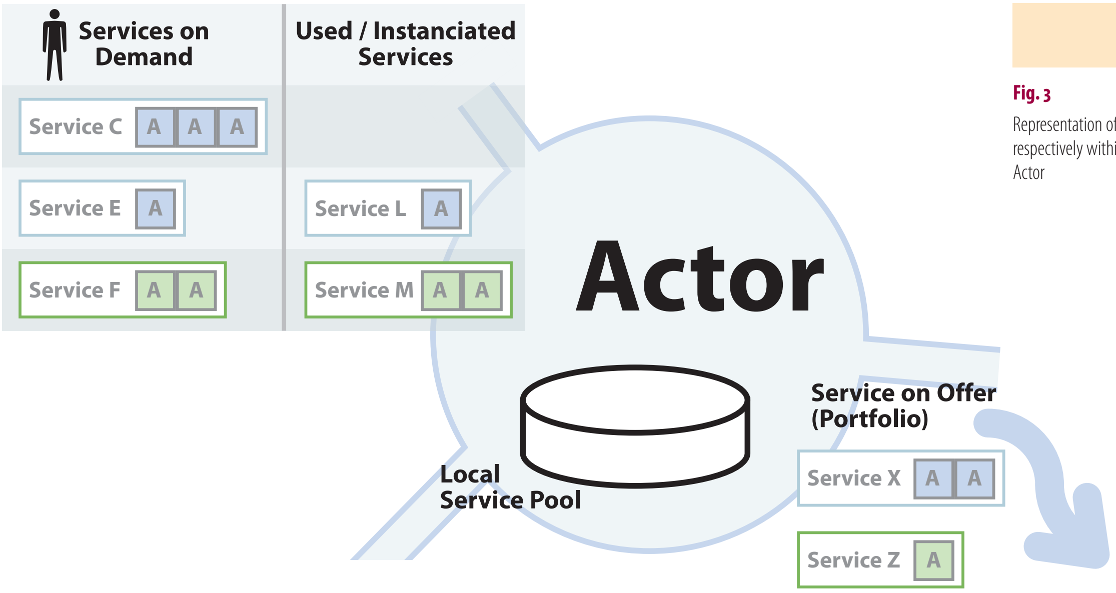


Fig. 3
Representation of a SME and Habitat, respectively within the EVESIM called Actor

In order to set up the network based on the social variables described in Section 2.7.2, we introduced capabilities for variable actor configuration. Each actor can be configured along seven social variables and can be represented by a user-defined picture and a name. For assigning this configuration to the SME agents, two approaches are possible.

First, a network of a region is imported through a CSV file import and the types of actors can be defined through the seven social variables. After importing the SMEs in terms of name and social connectivity to other SMEs, each SME can be associated with a type and therefore the specific behaviour is set, e.g. number of services on offers and demand.

Second, by configuring types of actors and including a number of actors present in the network, a higher-scale network can be extrapolated for testing algorithms for certain topologies and types of actors. One important indicator here is the so-called 'social capital' that indicates the connectivity of a certain actor with other actors in the network. Though, extrapolating a network is hard and not accurate at all, the usage of roughly defined actors make a simulation of a higher-scale network at least closer to reality than using a random network independent of the types of regional actors.

The technical aspects outlined here enable the EVESIM to emulate boot-strapping behaviour of digital business ecosystems based on real-world-data which is an important feature for providing visualisation-based convincing forecasts for new DBE users and organizations.

Collaboration: A Process of Reciprocal Understanding

The collaboration between Social Science and Natural Science has been focused in the first step on the possibility to transfer knowledge on engaged SMEs to the EVESIM. Social research, in fact, focuses on SMEs relational networks and - thanks to network analysis - visualises the correlation among SMEs and other local players.

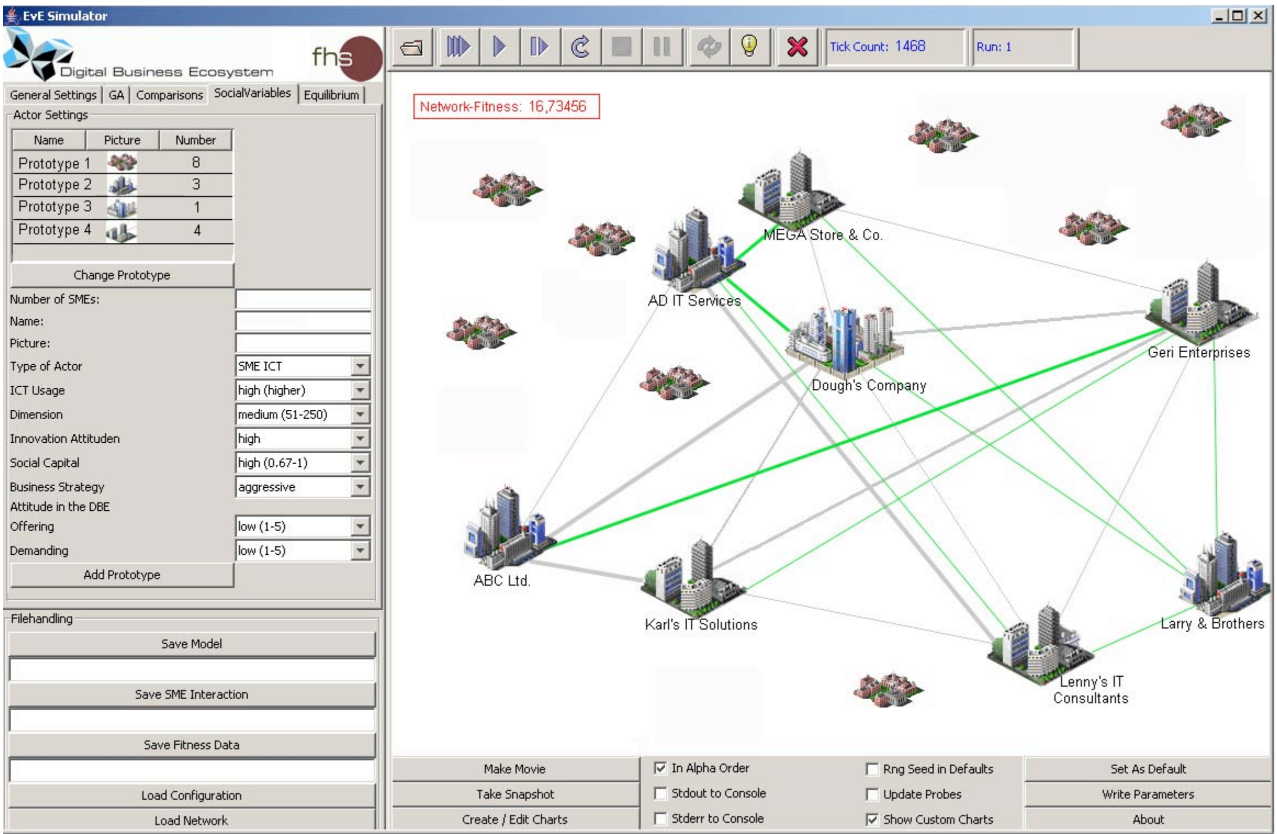
The successful transfer of data is a first result of the collaboration described here. Respective data was related to regional catalysts (RC) and Driver SMEs only and did not impact the general structure of the simulator (variables, SMEs profile, algorithm, etc.). Now that Implementers and Users SMEs have been engaged, new data will be available and will be integrated in the simulator thanks to the input/export features provided by EVESIM. This first impact on the structure of the simulator is now visible. The connection typologies studied so far have been already introduced above. These connection typologies go from personal contact and sharing of resources (as maximum of connection among SMEs) to more sporadic or absent relations. Those networks are not networks of services (pieces of software migrating from one ambient to another) but relational and business networks of SMEs engaged in the DBE. Nevertheless, the two layers - network of services and real-word connections - show important points of contact.

For example, three SMEs involved in several projects together may wish to share an agenda and look in the DBE platform for an agenda synchroniser. Besides this, face-to-face or business collaborations can have an impact on the level of trust between two enterprises. A high degree of trust, consequently, may invite one SME to prefer services provided by an already known enterprise instead of an unknown entity by this way introducing an important element in the migration pattern.

In both examples, a real word connection impacts a digital activity and service exchange. Besides this, the collaboration developed so far provided a common ground for understanding and developing a common language. This is a first step for real interdisciplinary research.

Fig. 4

Screenshot of the EVESIM with different actors in a small network.



The next step was to define different actors by introducing certain characteristics for each actor in a DBE network. We tried to visualise a possible definition of different actors (drivers, implementers, users, other local actors) in terms of interactions, i.e. trying to understand if a connection exists between the actor's role in the network and its level of interaction/collaboration with other local players. Besides this, social analysis provides a sort of typology of SME profile in terms of business domain, business organisation and possibly of a service to be requested.

In the future it will be important - again thanks to the collaboration of Social Science with Natural Science via the EVESIM - to understand the possible relationship between SMEs profile and service migration rate. This will require further analysis but will be of great impact on the simulations itself. At this stage it is interesting to consider different advantages that different DBE partners can take of the simulator.

From a computing perspective, the simulator is an important tool for visualising positive aspects of Peer-To-Peer Networks and self-organisation. From a Social Science perspective and a training respective RC's perspective, the simulator can become an interesting instrument for explaining to SMEs and regional players the relevance of collaboration and of DBE. By modulation of SMEs' profiles and other contextual variables it will be possible to show which are the positive mechanisms of knowledge sharing, collaboration and clustering. Besides the potential of making benefits of DBE visible amongst all stakeholders, EVESIM acts as important building block for the conceptual study of the intrinsic optimisation potential of the DBE. It offers pre-flight features for further steps in conceptual and technical development and it makes it possible to adjust technical aspects of the infrastructure based on hypothesis testing and prior emulation. But not secondarily, it becomes an unexpected field of interdisciplinary collaboration.

References

Bourdieu, P. (1980), *Le capital social. Actes de la Recherche en Sciences Sociales,* 31

Briscoe, G. and De Wilde, P, (2006), *Digital ecosystems: Evolving service-orientated architectures. In Proc.* IEEE Bionetics

Colin, R. Reeves and Jonathan E. Rowe (2002), Genetic Algorithms - Principles and Perspectives

Heistracher,T. Kurz, T. Masuch, C., Ferronato, P. Vidal,M. Corallo, A. Briscoe,G and Dini, P (2004) *Pervasive Service Architecture for a Digital Business Ecosystem*, CoRR, cs.CE/0408047:71-80

Heistracher, T. Kurz, T. Marcon, G. and Masuch, C, (2006) *Collaborative software engineering with a digital ecosystem*, In Proc. IEEE ICGSE

Kurz, T. et al. *Evolutionary Environment Simulator – evesim*, Webpage. http://sourceforge.net/projects/evesim. Last accessed on 12/22/2006.

Kurz, T and Heistracher., T (2007), *Simulation of a Self-Optimising Digital Ecosystem.* Accepted at the IEEE DEST

Kurz, T. Marcon, G. Masuch, C and Heistracher, T. (2005), *A network of SMEs for competitive services*, In Proc. Managing Global Trends and Challenges in a Turbulent Economy

Linthicum, D. (2003), *Next Generation Application Integration: From Simple Information to Web Services*, Addison Wesley

Masuch, C (2006), *Evolutionary Environment Network*. Webpage. http://evenet.sourceforge.net. Last accessed on 12/22/2006.

Narayan D. and Cassady, M.F. (2001) *A dimensional Approach to Measuring Social capital: Development and Validation of a Social Capital Inventory*, Current Sociology, Vol. 49(2): 59-102, London. SAGE

OMG (2005), *Semantics of Business Vocabulary and Business Rules (SBVR*), Draft adopted specification.

OMG (2006), *Semantics of Business Vocabulary and Business Rules Specification*, First interim specification,http://www.omg.org/docs/dtc/06-03-02.pdf. Last accessed on 12/22/2006.

Portes, A.(1998), *Social Capital: Its Origins and Applications in Modern Socio*logy, Annual Review of Sociology 24: 15-18.

Woolcock, M. (2001), *The Place of Social Capital in Understanding Social and Economic Outcomes*, Isuma 2, no. 1: 11-17