

NESSI Strategic Research Agenda

Vol. 2 Strategy to Build NESSI

Review Draft 2.1 - Version 2007-12-03



CONFIDENTIAL DRAFT
FOR DISTRIBUTION TO NESSI MEMBERS ONLY

The information in this document is provided "as is", and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

ONCE RELEASED, This work will be licensed under the Creative Commons Attribution-NonCommercial-NoDerivs 2.5 License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-nd/2.5/> or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.

1 Introduction

[In this document NEXOF means NESSI Open Service Framework and does not refer to the NEXOF-RA project.]

A general introduction, common to both Vol. 2a and Vol. 2b, on the purpose of the document and its position in the series of NESSI SRA documents (Volume 1, Volume 2, Volume 3.X).

“What is not” this document.]

1.1 Research horizon

[This section describes the advances in current baseline technologies, to push the boundaries of what is possible with the "state of the art" in order to better support the NESSI vision. The idea is to define the research horizon without taking into account possible disruptive inventions.]

1.2 Overall Scope

[The strategy for the implementation is conceived to be described into three different scopes:

- Systemic foundation (people and process) which provides the scope of Vol. 2a. The purpose is to describe the scope in terms of societal issues including economy, human beings, and regulatory aspects.
- Technology foundation which provides the scope for Vol. 2b. The purpose is to describe all the technological aspects, including standards, which will be at the basis of the NESSI Open Platform.
- NESSI foundation to support vertical domains (specifies requirements for NESSI) which provides the scope for an amended version of Vol. 1. The purpose is to provide the requirements for the NESSI Open Platform which come from various domains (e.g. Health, Gov, Automotive) and from its possible instantiation in different market sectors as those represented by large industry and SMEs.]

1.3 Building process

This document has been built under the responsibility of the NESSI Strategic Research Agenda (SRA) Committee. In the process of building this document the SRA Committee solicited and received contributions from various NESSI Working Groups (NWGs), such as “Business Process Modelling”, “Semantic Technologies”, “Service Engineering”, “Service-Oriented Infrastructure”, “Services Sciences and Systems”, “Trust, Security and Dependability”, and many individual organisations and researchers.

The definition of the structure of Volume 2 started right after the publication of Volume 1 (February 2006). In November 2006, within the process of defining NESSI Strategic Projects (NSPs), an Expert Group was established. This group produced a very first draft of the overall functional architecture of the NESSI Open Service Platform which served for the purpose of contextualising the NSPs but also to understand their mutual complementarities. This draft is the predecessor of the what in this document is in Section 3 of Volume 2b.

During 2007 several workshops, both virtual and physical, occurred where people from SRA Committee and NWGs, attended. November 2007 was dedicated to an open consultation and call for contribution phase, the result of which is fully integrated in this Volume 2.

The SRA Committee would like to thank the many contributors from the Committee and NWGs for their time, passion and effort devoted to produce Volume 2 and particularly: Luigi Telesca (Create-Net, Italy), Thierry Bouron, Christophe Cordier and Martine Guerlus (France Telecom – Orange, France), Dora Christodoulou (Ionian Technologies, Greece), Ismael Olea (Planeta Olea S.L., Spain), Luciano Baresi and Sam Guinea (Politecnico di Milano, Italy), Arne Berre, Parastoo Mohagheghi and Arnor Solberg (SINTEF, Norway).

Table of Content

1	Introduction.....	2
1.1	Research horizon.....	2
1.2	Overall Scope.....	2
1.3	Building process.....	2
	Volume 2a: Strategy to build systemic foundations for the service economy	6
1	Importance of the Service Economy.....	6
2	The Statistical Picture.....	7
3	Servicing the Research Needs	8
4	Education, Learning and Skills for an Innovative Service Economy	10
5	Creating New Working Environments & Employment.....	11
6	Legal and regulatory Frameworks for Service Innovations.....	12
7	Supporting IPR for Service Innovation	12
8	The Role of Demand in Stimulating Innovation in Services.....	14
9	The Role of Standards in Stimulating Innovation in Services	15
9.1	Public recognition of standards from consortia/fora:	15
9.2	Communication on standards and mandate initiatives	16
10	Removing barriers to the Internal and Global Market for Services	16
11	European Institute of Technology (EIT)	17
12	Conclusions.....	17
	Volume 2b: Strategy to build technological foundations for the service economy	19
1	Scope of this volume.....	19
1.1	From technological foundations to technological challenges (pillars).....	20
1.2	Addressing the whole “Functionality” (foundation) continuum	20
2	Relevant Background	20
2.1	Introduction.....	20
2.2	Software as a Service	20
2.2.1	Standards.....	22
2.2.2	Reference Models, Architectures, and Implementations.....	23
2.3	Infrastructure Layer.....	24
2.3.1	Adaptive Service-aware Infrastructure.....	24
2.3.2	Data Management	26
2.3.2.1	Application Server Replication	26
2.3.2.2	Database Replication.....	26
2.3.2.3	Self-healing	27
2.3.2.4	Self-optimizing and self-provisioning	27
2.3.2.5	System Infrastructure and communication support.....	27
2.4	Service Platform Layer	28
2.4.1	Service-Centric Systems Engineering	28

2.4.2	Service-Oriented Architectures (SOAs).....	28
2.4.3	Business Process Management.....	31
2.4.3.1	Concepts and Models for Service Value Networks (SVNs).....	32
2.4.3.2	Concepts and Models for Collaborative, Cross-organisational Business Processes.....	32
2.4.3.3	Concepts and Models for Adaptive and Autonomic Business Processes.....	33
2.4.3.4	Business Process Management and Analysis Platforms.....	33
2.5	Service Consumer Layer.....	33
2.5.1	Advanced User-Service Interactions.....	33
2.5.1.1	State of the Art overview of Context Modelling.....	35
2.5.2	The Semantic Web.....	37
2.5.3	Contextual Computing.....	38
2.5.4	Web 2.0.....	39
2.6	Engineering Approaches.....	43
2.6.1	MDE.....	43
2.6.2	Goal oriented software development.....	44
2.7	Limits of today’s systems (some examples).....	45
3	The Open Service Platform.....	45
3.1	Overall view.....	45
3.2	Description of Functionalities.....	47
3.2.1	Service consumer layer.....	48
3.2.1.1	User-Service Interaction.....	48
3.2.1.2	End-user interfaces.....	49
3.2.1.3	Context Modelling.....	51
3.2.2	Consumer Adaptation (functional blocks).....	52
3.2.2.1	Mapping users perspectives to business/Integration.....	52
3.2.2.2	Context handling.....	52
3.2.3	BPM View.....	53
3.2.3.1	<i>Adaptive</i> Business Process Execution.....	54
3.2.3.2	SVN Lifecycle Management.....	55
3.2.3.3	Business Process Execution.....	55
3.2.3.4	BPM Modelling.....	56
3.2.3.5	Business Process Simulation.....	56
3.2.3.6	Knowledge Modelling.....	57
3.2.3.7	SVN Modelling.....	57
3.2.4	Service Composition Layer.....	57
3.2.4.1	Service Discovery.....	58
3.2.4.2	Service Communication.....	60
3.2.4.3	SLA Negotiation.....	61
3.2.4.4	Mediation.....	62
3.2.4.5	Service Coordination.....	63
3.2.4.6	Service Registration.....	65
3.2.4.7	SBS/SBA Modelling.....	66
3.2.5	Service Layer.....	66
3.2.5.1	Service Execution.....	66
3.2.5.2	Integration Services.....	68
3.2.5.3	Lifecycle Management.....	68
3.2.5.4	Reasoning.....	68
3.2.5.5	Service Modelling.....	69
3.2.6	Abstracted Infrastructure Layer.....	69

3.2.6.1	Infrastructure Abstraction.....	69
3.2.7	Infrastructure Layer.....	69
3.2.7.1	Resources Management.....	69
3.2.7.2	Data Management	70
3.2.7.3	Infrastructure Modelling	72
3.2.8	Monitoring	73
3.2.8.1	Collaborative Business Intelligence	74
3.3	Description of Non-Functional Aspects	75
3.3.1	Introduction.....	75
3.3.2	Service Non-Functional Properties	76
3.3.2.1	Security Non-Functional Properties	76
3.3.2.2	Security services.....	78
3.3.2.3	Secure (and dependable) services.....	80
3.3.2.4	Performance Non-Functional Properties	82
3.3.2.5	Behavioural Non-Functional Properties	82
3.3.2.6	Deployment Non-Functional Properties.....	82
3.3.2.7	Business Non-Functional Properties	82
3.3.3	Infrastructure Non-Functional Properties.....	82
3.3.3.1	Dependability	82
3.3.3.2	Security	82
3.3.3.3	Performance	83
3.3.3.4	Interoperability	84
3.3.3.5	Manageability.....	85
3.3.3.6	Governance	85
3.3.3.7	Flexibility	86
4	Core Principles	86
5	References.....	88
5.1	Volume 2a.....	88
5.2	Volume 2b.....	88
6	Appendix A – SOA standards table.....	95

Volume 2a: Strategy to build systemic foundations for the service economy

1 Importance of the Service Economy

Traditionally considered as a heterogeneous ‘left-over’ collection of activities that are not included in the agriculture or industry sectors, the services/software sector has, until recently been a neglected area of economic policy making¹.

However the services sector in the European Union (EU) is growing considerably and now accounts for over 70% of total EU economic value added². Services are essential for the efficient operation of an economy, facilitating commercial transactions and enabling the production and delivery of goods and other services. As companies learn to trade products and services in new ways, often through ICT, services have become a pillar of the European economy. A country with an open, dynamic and efficient service sector enjoys a competitive advantage in the production of both goods and services, as compared to countries with less developed service sectors.

Also manufacturing industries are changing to include more and more services. This is partly due to the fact that services are increasingly becoming part of tangible and intangible products. Service packages covering installation, maintenance, updating, training and so on become an integrated part of delivery. In this way - connected to the outsourcing of traditional production to low-cost countries - many European companies concentrate on services and development. In many cases the attached services become the main products.

Over 70% of EU employment is in the services sector and this figure is set to rise in the coming years³. Services have been the source of most job growth over the last decade.

One misconception about the growth of the service sector is that it is creating more low skill, low value jobs than high skill, high value jobs. While the service sector includes some low-skill jobs, many other service jobs require high levels of skills or advanced education to perform complex tasks in the information economy. In fact, the percentage of employees with a college degree is greater in the service sector than in the manufacturing sector⁴.

According to Eurostat, ”services are the only sector of the European economy that has generated jobs in the last decades”. Despite so much of Europe being dependent on services they have received little or no attention in terms of policy-making.

When we think of services we may not appreciate the breadth of economic activity that they encompass: the engineer’s network design, the barber’s haircut, the doctor’s diagnosis, the waitress’s service, the architect’s building plans, the carpenter’s craftsmanship and the consultant’s business strategy.

The term service covers a broad range of activities that is difficult to encapsulate in a single definition. Services are activities that produce value by providing solutions to customer problems. Services may create change in the customer or the customer’s assets. The service sector includes everything from child care, to legal advice, to custom software development and management consulting. In some cases, it is not easy to separate services from the goods with which they are associated, such as an extended warranty purchased with a consumer electronic device or the

¹ European Trend chart on innovation in services.

² OECD, growth in Services, p.1.

³ OECD, growth in Services, p.1.

⁴ OECD, Promoting Innovation in Services, p.26.

rental of an automobile. Services can also be embedded in a manufacturing process, as manufacturers procure inputs, such as inventory management and logistics services, from service providers, rather than perform these functions themselves.

For the purpose of the recommendations made by NESSI, we focus on innovation in ICT services, ICT-enabled services and knowledge-intensive services. The pace of growth in business services (i.e. those supplied to other firms, such as computing, financial, legal, consulting, advertising and marketing services) is spectacular, even measured against those of other services sectors. A Commission Communication on business services⁵ noted that they provide 8.5 % of total employment in the EU, and 15.3 % of value added (more than banking, insurance, transport and communications services combined).

More and more services are being realized in Software. The different application domains - financial, legal, governmental, industrial, business and personal services are converging. Companies will not be able to provide all services needed for particular applications - the collaboration of services coming from various sources, big and small companies, is absolutely necessary. Efforts for providing interoperability of all kinds of services, especially software services are needed that will lead to standardization for interoperability.

More and more companies will not ask just for products; they will buy solutions that allow them to modernise their business processes. Efforts for modelling business processes and implementing them on top of software services will be necessary to support the competitiveness of European industry.

Transforming Europe into a knowledge intensive and globally competitive services & software economy is, even with sufficient investment for research in software and services, not straightforward. This requires a paradigm shift through complementary investments and changes, e.g. in human capital, organisational change, education, trust and security. It is paramount that we manage the transformation to a European services economy not only from a technological but also from a socio-economic and human capital perspective.

2 The Statistical Picture

The lack of adequate data, indicators and methods to analyse services and service innovation, has been the constant refrain of researchers studying services over the years. Firstly, there is a problem in analysing and studying services and their innovative potential have because services are simply too big a ‘sector’ to study in any meaningful or coherent form. The sheer size and significance of the sector within the economy has, therefore, created its own problems in terms of analysis and policy formulation. Services not only account for an increasing share of the economy, but services are also far from homogeneous. In addition, they also interact amongst themselves and with other sectors of the economy (notably manufacturing) in complex ways[Hamdani00]. Thus, there is a significant challenge in reporting the diversity of activities covered by the services sector, and to provide an informed commentary on the innovation trends across these diverse activities. With such size and diversity, it is perhaps not surprising that anyone could satisfactorily hope to cover such a heterogeneous and diverse set of industries with a single monomorphic model or paradigm.

Problems over lack of metrics, indicators and data have crucial policy implications as well. Thus, the European Commission in 2003 noted “... serious deficiencies in our understanding of the structure of the services sector and the factors influencing the growth of services enterprises remain. The available statistical material does not reflect appropriately the dominant position of services in the economy...”⁶. However welcome subsequent initiatives have been, such as extending the European Community Innovation Survey beyond manufacturing to include services, insight into services and service innovation, and in turn policy formation, are still hampered by this lack of adequate basic statistics on services and service industries, which is a prerequisite for policy formulation, monitoring and evaluation. There, therefore, needs to be recognition amongst all the key stakeholders associated with the study and

⁵ ‘The contribution of business services to industrial performance’ COM (1998) 534 final.

⁶ Commission of the European Communities (2003b), op. cit., 36.

analysis of service innovation that we need a much better understanding and measurement of the innovation process in relation to services. Innovation statistics are still strongly biased towards technological innovation, and the measurement of knowledge inputs and innovative processes and outputs in services is one of the key areas where initiatives are needed within member states [VanArk03], but also within a European and international level. Much more effort needs to be done to compare service innovation between countries [Kanerva06], but this has been hampered, amongst other things, by comparability of datasets.

However it is important to recognise that services and service innovation remain difficult to study and conceptualise. It is therefore not ‘simply’ a matter of funding the collection of more data and the creation of new and more comprehensive datasets. More effort also needs to be undertaken by the research community in developing new, more robust, indicators that can actually better articulate and measure what service innovation is about, rather than simply trying to adapt old modes of thinking in relation to innovation.

Services are becoming more R&D-intensive: between 1990 and 2003, service sector R&D increased at an average annual rate of 12% across OECD member countries, compared with only 3% for manufacturing sectors. Thus, in 2002 the European Union average for the share of services in business R&D rose to over 15%. Services are increasingly innovative: the share of business service firms reporting that they were innovative in terms of introducing an innovation between 2002 and 2004 in the EU ranged from around 50% in Germany to less than 20% in Denmark.

We should be seeking better measurement and understanding of service and non-technological innovation. Statistics on services require further development. Co-ordinated efforts by the member states, European Commission, Eurostat and the OECD provide a key to the effective development of more accurate service statistics.

Recommendations:

1. Encourage key stakeholders, such as member state governments and statistical agencies, associated with the study and analysis of service innovation that we need a much better understanding and measurement of the innovation process in relation to services. Innovation statistics remain strongly biased towards technological innovation, and measurement inputs and outputs in service innovation needs to be further developed and supported.
2. Support the research and statistical community in developing new, but robust, indicators that can actually better articulate and measure what service innovation actually is.

3 Servicing the Research Needs

Government innovation policies, R&D budgets and programs have historically focused on hard sciences and manufacturing. This needs to be rebalanced given the fact that services are the source of most jobs and economic activity. How these programs are designed is important, because the innovation process in services can differ from that in manufacturing.

Over the last two decades or so, advances in internet technologies, open systems, and global reach have fundamentally shifted the way enterprises are managed. Many have moved from being centralized monolithic organizations to being networked collections of firms collaborating and sharing services with specialized and niche partners and customers worldwide to produce goods and services more quickly, more efficiently, and more effectively than before. Enterprises in Europe and elsewhere that have traditionally focused on building everything in-house now embrace partnering with specialists and service providers from different business ecosystems to provide critical and core products and services to customers. These sorts of complicated business networks are just now beginning to take shape in many industries, and so now there is increased complexity in harnessing the right services for value-creation, and a correspondingly increased need for understanding complex business services ecosystems.

In the context of services sciences, network dynamics depend not only on connectivity between firms and customers but also on connectivity within firms: that is, the socio-technical interface resides within and between firms, as well as at the point of service delivery. But what is the ‘theory of the network’ (or the ‘theory of the ecosystem’) that is of

comparable analytical value to the classic 'theory of the firm'? Understanding the rapid creation of service-chains and services networks, and understanding the value they provide to the ecosystem, is critical for current and future social and business environments.

NESSI is beginning to address what it means to be in a world in which the harnessing of services systems is a complex problem. We envision a multi-disciplinary (anthropological, social scientific, economic, computational, management, and other) effort for tackling the challenges of understanding services and deriving principles for harnessing services for providing value effectively. The NESSI Landscape aims to focus not only on enabling frameworks and technologies for business services in Europe, but also on long-term economic, social, and organizational aspects of services. From a scientific point of view, the main challenge is to understand, model, and validate the complex, networked services (whether for society, business, or IT) that will provide a solid foundation for understanding networks of business collaboration, of social collaboration, and of human interactions.

Non-technological aspects of innovation and research in services, aiming to integrate theories, methods, and findings from a broad set of disciplines are crucial. The overarching challenge will be to develop a counterpart to the classical theory of the firm that takes account of networks of relationships (among people and technology) within firms and across firms, and particularly as these affect service innovation and service delivery and ultimately create value.

A secondary challenge is to apply this new theory of the firm to shape laws and policies, aiming to place Europe in position of economic leadership. Fundamental issues include whether legal barriers could prevent the development of new markets and business models, whether safeguards are needed to make new economic activities acceptable, and whether policy-makers will embrace the concepts and measures of services science.

From a disciplinary perspective, a number of areas and issues readily suggest themselves, including, but not limited to:

- Anthropology: What is going on? Who communicates with whom and to what purpose? How is the communication changing (e.g., blogging)?
- Sociology: How to conceptualize what is going on? What are the underlying social structures? Who are the new 'effluentials'? In which way and to what extent have societies already transformed (the gaming generation, second life gurus, etc)?
- Economics: How to capture the value of what is going on? What economic activity is being supported by the social structures and what are the dynamics of the value flows? Do we need a strategic analysis (i.e., game-theoretic) of network dynamics? How can viable business models be built around this?
- Mathematics: How to model what is going on? Can we build predictive models of the structure, dynamics, and value of networks and the constituent components? Can we appropriately capture the socio-technical interface?
- Engineering: How to build the systems? Can we design and construct systems, to deliver complicated, composed services that embody our understanding of networks and address socio-economic challenges, as described above?

Research funding plays an important role in stimulating service sector innovation. Research could help solve problems that ICT services providers face in managing and ensuring the reliability of complex service delivery systems. Research can also aim at better understanding the non-technical aspects of service-sector innovation, in particular organisational innovation, drawing on advances in the social sciences and management.

Service related R&D reflects the heterogeneity of the sector itself. Hence, there is a need to develop better understanding on the nature of service innovation and related R&D in connection with different types of services. For instance, service related R&D can be very close to basic research (e.g., insurance mathematics and financial modelling), or close to market activity (e.g., hotel reception process design).

The European Commission has many opportunities to facilitate service innovation research and indicator development horizontally (e.g. CIP and FP7). However, some specific research initiatives may be introduced on the need basis.

Above all, it needs to be taken care of that the research themes, project assessment criteria, and the evaluators do not form a systemic barrier to service research. The inherent bias towards manufacturing industry and technology based projects needs to be dealt with. For instance, typical service innovations are multidimensional, including organisational, service concept, business model, customer interface, and delivery system as well as technological elements. This bears significant influence on the R&D activities in services that also have many distinct features. Such features include: informal nature of service R&D, importance of customer interaction and overall great variety of service related R&D ranging from close to market activities to basic research type activities.

In fact, NESSI aims at supporting collaborative scenarios involving also the creation of complex and dynamic business eco-systems. To represent with the right abstraction levels, model and subsequently simulate these eco-systems (that include technology infrastructure, tools, business and legal frameworks, etc.) is an essential aspects to be investigated.

The effort should be in identification and definition of languages for systems modelling integrated in advanced, context sensitive (domain dependencies) and adaptive development environments enabling a multi-disciplinary research approach for operational analysis, operation research and models optimisation, stochastic and fractal processes.

For the models simulations it should be taken into account past experience such as, for example, ones related to the IEEE HLA (High Level Architecture standard) and the connections/interrelation with Grid infrastructure as well.

Recommendations:

1. Research support should be provided and actively promoted, both at EU and Member State levels. Inter-disciplinary research should give priority to developing measurement criteria and indicators for service activities in both the public and private sectors. Collaborative research based on, for example, behavioural sciences, mathematics and modelling needs to be actively encouraged.
2. Interdisciplinarity, especially between Humanities and Social Sciences and Science, Engineering and Technology, should be central to any funding initiative.
3. Funding programmes for enterprises and their eligibility criteria are primarily aligned with technology based research. Risk adversity in services research will be greater and require greater levels of flexibility. At national and EU levels, the levels of research funding available need to be adapted and favour increased non-technology based research including their evaluation criteria.

4 Education, Learning and Skills for an Innovative Service Economy

Skilled and creative employees are a fundamental factor in the innovation process and a major source of competitive advantage. In the Agricultural Age, land and farm production defined competitive advantage. In the Industrial Age, it was raw materials and manufacturing capability. Today, it is the ability to create and apply intellectual capital based on multidimensional expertise – increasingly in the area of services.

Of course, the importance of Education and Training to improve skills and competences has to be recognised in our society as well as the centrality of the learner (e.g. a citizen) in the learning process and the potential impact of new learning paradigms combining several educational models such as, for example, socio-constructivist and experiential based forms of education. During the learning process, learning activities should be aimed at facilitating the construction of knowledge and skills in the learner instead of the memorisation of information. Information transfer will still obviously exist in this paradigm, but only as a simple component, not the main goal.

Workforce skills must include both technology and strategic expertise. An understanding of technology – its current capabilities as well as its future potential – is now integral to business decision making. Importantly, these skills are not static, requiring continual refreshing through life-long learning and retraining. Technology and skills in relation to innovation is not an either/or decision. The majority of service firms attach equal importance to investing in new technologies and in skills.

In the past IT services were all about “repair and maintenance.” Today, services are about optimizing business. There is a lack of people in Europe with both IT and business skills which understand the new role of IT services and who be interested to work at a European level. This is probably the biggest challenge for Europe: create a mobile workforce which can operate across the cultural and language barriers.

Consequently we need to adapt education and training policies to rapidly changing requirements for new skills and create a new discipline for services sciences. Services science is a multidisciplinary field that seeks to bring together knowledge from diverse areas to improve the service industry’s operations, performance, and innovation.

In essence, it represents a melding of technology with an understanding of business processes and organization. It is a shift from a technology-centric view to a holistic view that encompasses both technology and business and the centrality of the users (e.g. the learners). Professionals need new skills and education in a variety of fields to yield the best results in service industries as well as new learning paradigm focused on active knowledge construction. It is critical to develop and foster a broad perspective that includes research from many areas, including economics and law.

Recommendation:

1. Governments, industry and universities together must enable the creation of a new academic discipline on Service Science to bring together ongoing work in computer science, operations research, industrial engineering, business strategy, management sciences, social and cognitive sciences, and legal sciences to develop the skills required in a services-led economy. Also schools should be involved in this process. There will be a change from ICT workers with specialized technical skills towards hybrid professionals with competencies in business or scientific areas beyond traditional ICT who will be able to respond to the challenges of a more dynamic service oriented economy. Well targeted education policies will have a significant positive effect on the competitiveness of the European ICT and knowledge service providers given the sector’s dependability on highly skilled workers.
2. In response to the political pressures for education, skill and employability, it should be avoided a technology push answer and the EU Governments, Industries and Universities have to propose new strategies focused on the learners and its needs, preferences, dispositions. Services supporting personalisation, contextualisation, and realisms should be part of this strategy.
3. Eventually, the new strategies to develop should take into account the mutual dependencies between enterprise and organizational learning and business processes and their integration in order to allow the optimisation (tailoring and further personalisation) of employees’ learning plans with respect to the business processes (including competency development, skill gap shortening, etc.) and the optimisation of business processes taking into account competencies, skills, performance and knowledge available inside the organization.

5 Creating New Working Environments & Employment

In OECD countries, most employment growth over the last decade was due to services, and in particular business services⁷. Services are the only part of the European economy that has generated a net employment gain over the past two decades (now accounting for over 70% of all jobs). However, it is important to recognize that the ICT marketplace is continuing to change – and change dramatically – and that the skills and working methods needed in that changing marketplace have to be further developed. This is all happening fast, and in many different dimensions.

More ICT customers/clients are buying on the basis of business value, and not on the basis of technology. This will result in a major restructuring of enterprises in Europe including the virtualisation of enterprises. Work will be more mobile, migrating back and forth between centres of activity as people with the right skills migrate to where work is needed. In this context the modernisation of working environments will be crucial.

⁷ OECD, Promoting Innovation in Services, October 2005, p.9.

More people will work in more flexible arrangements including self-employment and teleworking. This could raise issues in a variety of policy areas including employment legislation. It will be crucial to anticipate and facilitate change to help to improve our competitiveness and will contribute towards a better quality of European working life.

Recommendation:

1. The European Commission and Member States should promote employment policies that provide greater adaptability of firms and workers to better anticipate and facilitate change. By transforming Europe into a services economy we will move ‘up market’ to secure higher level jobs in management, problem solving and creative thinking. Furthermore it will keep the SME-sector vibrant by increasing competitiveness. Participation of SMEs in this process should be guaranteed.

6 Legal and regulatory Frameworks for Service Innovations

More transparent service markets are a key European level issue that bears significant influence on innovation activity. It is important that member states continue their efforts in identifying regulatory problems that defer the development of service markets and innovations. Overall, policy benchmarking should be an important tool for member states as they develop service innovation policy.

One factor behind the success of service sectors is that they have been able to grow in a generally deregulatory and non-interventionist climate. It is therefore necessary to place great weight on getting the environment right – particularly regulation with a light touch.

We caution against a presumption of going down the regulatory route unless there is a very clear need and in clearly defined areas. For example standards may support innovation if business driven, but regulation of professional standards could reduce competition.

Services display a rapid growth in smaller economies, often fuelled by inward investment, innovation and deregulation. There will be a need to recognise the business model which drives enterprises providing services.

Both the regulatory and legal frameworks not only need to be aware of this but work proactively with the enterprises concerned. There will be a need for the European Commission and the Member States to work closely together to achieve this.

The formation process of services often require short term access to specialised activities and depend on great levels of flexibility, often provided by small to mid-sized SMEs. Their modus operandi, for example through combination of SMEs and disaggregation of consortia, will provide significant challenges to all stakeholders in the legal and regulatory framework environment.

Nevertheless this will involve a change of culture that is not easy to accomplish. Consequently there is considerable potential for the Commission to consider how best this could be implemented in the important context of its own work as initiators and guardians of the many regulatory processes. The Commission can play a powerful role and act as an exemplar to Member States in showing that being supportive and creative in services innovation does not necessarily involve any weakening or compromise of regulators’ primary objectives.

Recommendation:

1. We would support an approach that encourages good self-regulation with regulation being the last resort. There should be an encouragement of good practice in self regulation of professional standards as this has the benefit of being industry driven, flexible and facilitating choice of both quality and cost of the service.

7 Supporting IPR for Service Innovation

Informal IP protection has a very important role in the service innovation context. Importantly, IP protection in services is not limited to formal IPR methods that tend to be more suitable for industrial manufacturing purposes.

In contrast to technology based innovation, innovation in services is more likely to be protected by IP processes other than patents. Any IP mechanism and support to provide protection to innovations in the provision for services will need to be flexible, able to facilitate the perishability of the service, facilitate the very short delay between innovation and delivery to market and recognise that first to market is often the difference between successes and failure.

Open standards provide technical specifications for implementing features and functions that are developed by neutral consensus-based organizations in an open participatory environment, and made publicly available for all to implement on equal terms. Open source refers to software code that is publicly available in human understandable (source code) form, enabling anyone to copy, modify and redistribute it without paying royalties or fees. Development of open source software is a powerful example of collaborative innovation. Typically, open source programmers collaboratively create software, improve it and constantly share code changes within the community. Open source software can accelerate open standards by serving as a basis for common implementation. Open source stands in contrast to proprietary software in which one company or organization controls the development of the software and makes it available in object code form only.

NESSI believe that both open source and proprietary software are important parts of a contemporary IT marketplace. Open source communities quickly deliver to market innovations that can be adopted by companies which then build proprietary offerings on top of them. Both kinds of solutions should continue to coexist and complement one another. Governments should ensure that their policies -- including IP policy -- not discriminate against either software development model.

Increasingly, knowledge is global, as is business. We believe that policy goals should be framed to allow European companies to prosper on the basis of new technology not only in Europe, but as leaders in the world market. It is therefore vital that Intellectual Property regimes are formulated in such a manner that they do not inhibit European researchers and developers from participating in world-wide knowledge communities, nor European companies with foreign interests from exploiting their IPR outside Europe.

Unlocking innovation also demands that a contemporary IP policy, especially patent policy, enable both strong IP ownership, an essential driver of innovation, and fundamental technological advances that today are dependent on shared knowledge, open standards, and collaborative innovation. NESSI believes that a strong, global, intellectual property system encourages innovation⁸. But the strength of that system depends on the quality it produces. Perhaps the greatest threat to innovation is low-quality patents – patents that are given to inventions that are not truly new and useful. Governments should make sure that they only grant high-quality patents – patents for ideas that embody genuine scientific progress and technological innovation. Improvidently granted patents on old inventions or overly broad concepts remove from the hands of the public the very tools of innovation a healthy system is intended to foster. Low-quality patents can unjustly reward the patentee and make it difficult for competitors and innovators to use patented teachings to achieve meaningful advances.

In this context we also urge European policy-makers to address the undoubted issues with current patent regimes in the context of world-wide discussions. It is essential to avoid too divergent rules at an international level which would seriously increase the costs of European companies wishing to grow world-wide.

Recommendation

1. Governments should take into account implementation of open standards and the importance of interoperability when considering patent policy that promotes innovation.
2. There is a need for Member States to take a collective initiative to ensure cross national compatibility whilst accommodating national legal and cultural variability. In developing any IP mechanism for services it is important that services activities are acknowledged rather than taking a services sector, i.e. a sectoral approach.

⁸ Please note that there is no unified position within NESSI on the usefulness and support of software patentability.

8 The Role of Demand in Stimulating Innovation in Services

Public procurement can provide leverage to increase the demand in innovation in services. Public procurement is however faced with the dichotomy of risk adversity (accountability) versus creative solutions and the need to overcome the unproven and uncertainty dilemma. Demand driven research in this context may encourage risk sharing in the first instance and reducing uncertainty subsequently. Specific measures to support exploratory research are required to promote such an approach including flexibility in State Aid Regulations.

As a general principle, demand driven service innovation policy represents a highly relevant perspective as described in the Aho report [Aho06] and several other recent documents⁹. Public procurement related standards and regulations represent another promising area where demand driven policy measures can stimulate innovative services. At present the number of demand driven policy measures is very limited and they represent an area that has a lot of development potential.

Understanding new European state-aid regulations¹⁰ and their influence on service innovation policy is crucial. Member states face a steep learning curve in adopting the new state-aid regulations in the evolving service innovation policy context. Hence, it would be most important to establish an effective knowledge and good practice sharing process that would benefit the delivery of the European level service innovation policy as well as individual member states.

Member states ought to adopt new innovative policy programmes that can promote demand for external expertise in innovation projects. One way to do this is to offer demand-stimulating incentives for those who can make use of expert services. By creating demand also the supply and quality of expert services can be improved. Voucher schemes offer one example of such activities, however comprehensive evaluation results of such schemes are not yet available.

Regional clusters represent the operational environment where service innovation policy can be tailored to meet the specific needs of the surrounding economic environment. Such regional approach can be a basis for effective bottom-up developed service innovation policy that can stimulate demand as well as supply of innovative services.

Horizontal policies play an important part, for instance, structural funds and service innovation policy should support each other and thus facilitate effective delivery of the policy measures.

Public procurement can be a driver for business investment in innovation. Private suppliers of services react and interact with their customers and their demands on a daily basis regardless of the customer being a public purchaser or a private business.

The exception of R&D services to the public procurement directives and the recent published guidelines from the Commission on state aid for innovation both allow for Member States to launch R&D programmes to enhance innovation. We support that Member States stimulate these programmes in becoming more demand-driven.

Governments are also consumers in their own right and should seek ways to become more innovative procurers for services. As such, governments are in a unique position to support service innovation by acting as lead customers with ambitious requirements when procuring services. Governments should be encouraged to develop and share experiences of innovative procurement policies.

Recommendations: The European Commission and National Governments should:

⁹ Council of the European Union (2006), PRESS RELEASE, 2769th Council meeting Competitiveness (Internal Market, Industry and Research), Brussels, 4 December.

¹⁰ European Commission (2006), Community Framework for State Aid for Research and Development and Innovation (2006/C 323/01), Official Journal of the European Union C 323/1. 30.12.2006, Brussels, Belgium.

1. Foster more experimentation in developing more demand-driven R&D programmes and promote the introduction of advanced education and training in public procurement for civil servants of contracting authorities and suppliers;
2. Address the problem of risk aversion. Innovation may be considered in pilot or short-term projects, but often this is not followed through into longer-term projects, instead low-risk solutions are sought;
3. Clarify and elaborate further on the relationship between the state aid and public procurement rules;
4. Support innovation by acting as an early adopter of new ideas. Early adoption of ideas can have a major impact on supply-side business. Providing firms with their first significant customer for a new innovation can form the platform of respectability from which further sales and long-term growth can be achieved;
5. Use procurement more strategically to stimulate innovative firms. For example they could promote the use, by public purchasers, of innovative criteria in the award of contracts;
6. Put forward legislation that will encourage public purchasers to make procurement policies and to share these plans with the private suppliers. Early supplier involvement in the procurement process is critical if innovation is to be captured.

9 The Role of Standards in Stimulating Innovation in Services

The use of standards and regulation is a highly effective way to influence markets and the development of innovative services. At the same time, increasing regulatory burden can also be an effective barrier to innovation and the development of competitive services. Hence, standards and regulation needs to be used in a very focused manner in the carefully chosen areas, for instance in, a) sustainable energy production, and b) in connection with public procurement. Sustainable energy production and environmental issues represent an area where supportive regulatory environment could effectively stimulate the development of innovative services. In the future, global markets for energy and environment related expert services look very promising.

Standards in the ICT domain are largely developed in industry standards organisations, consortia and fora. Most of the times these are specialised organisations with strong expertise in the respective technology area and thus best suited for the respective standardisation project. The goal of such standards projects is always to develop global standards on a voluntary base and with clear market relevance.

In this context, it is one of the focal areas in NESSI to identify market needs for standardization lying within the technological scope of NESSI and promote industry initiatives for closing potential gaps in standardisation and taking the respective requirements into the proper standards development organisation (SDO), consortium or forum based on their key competence.

For proper management and execution of this task, NESSI has installed a committee on standardisation (NESSI-COSTA). This committee will coordinate all standards issues across the various NESSI working groups as well as with outside organisations.

With respect to the European standards policy, in the next two sections we highlight what NESSI sees as two major areas for improvement.

9.1 Public recognition of standards from consortia/fora:

Globally acting consortia and fora with strong representation of European companies have established as key competence centres for standards development in the ICT domain, e.g. W3C, OASIS, OGF (Open Grid Forum), OMG or IETF. The specifications developed by such consortia/fora are both high quality and meet the criteria of open standards.

Therefore, NESSI recommends that in the course of the currently ongoing revision of the EU Framework Directive 98/34 (and, thus, with direct impact to the Procurement Directive 2004/17) provisions are made to allow global specifications that have been produced outside the formal standards organisations to be referenced for public procurement. A possible way to implement such a solution is by defining binding criteria that a standard and the

respective standards development process need to meet in order to be eligible for acceptance in the European standardisation framework.

We believe that such changes in the European standardisation framework will help making high quality and state-of-the-art technology available for the European marketplace faster and will encourage participation of European companies in the various global standardisation activities thus increasing their competitive advantages on the global marketplace.

9.2 Communication on standards and mandate initiatives

Strengthened upfront communication between the EU Commission and industry on standards and mandate initiatives.

NESSI believes that efficiency can be gained by early involvement and consultation of industry before a public standards initiative is started or a mandate is issued. The main purpose of such a consultation process will be (1) to jointly evaluate the marketplace need for a particular standard and (2) to jointly evaluate whether international standards are already available or under way. This includes actions to avoid duplication of work and foster coordination between SDOs, consortia and fora working on similar items.

10 Removing barriers to the Internal and Global Market for Services

Despite the service sector's large share of the economy, services account for only 20% of intra-EU trade. Trade in services at a global level accounts for 20% exports. One reason for the low level of trade in services is that significant trade barriers exist across a range of services sectors in many countries.

ICT is enabling trade of many knowledge-based services and creating tremendous opportunities for exports to other Member States. Many successful services companies owe their existence and success to the opening up of markets. Opening services markets will create fresh opportunities for firms to develop new, often ICT-related, services and meet emerging global demands. In many ways business services companies are the model enterprise in the new economy – they trade heavily in 'knowledge' products, are often built primarily in intangible assets (such as people) and are particularly well placed to exploit the potential of the new ICT marketplace.

We need to achieve a legal and administrative framework, which allows for cross-border movement of services within the EU, ideally enabling enterprises to easily export innovative services business models beyond national borders. In this context business would like to remind the importance of the mutual recognition and country of origin principles for the creation of the internal market in services as laid down in the EC¹¹ treaty.

Furthermore, the European Commission should pursue an ambitious free trade agreement on services under the umbrella of WTO.

Recommendation

1. The European Commission and the Member States need to foster the creation of an Internal Market for knowledge intensive services beyond the provisions currently discussed in the Services Directive and support and secure an ambitious outcome in the WTO Doha Round, including significant market-opening commitments in services from as many countries as possible. In recent decades, services' share of GDP has grown significantly, yet the growth in services as a share of total exports has not kept pace. This implies that there is tremendous opportunity for expanded trade in services as the global economy grows, especially considering how many IT-enabled services are now more readily tradable. Lowering or eliminating existing trade barriers should spur further growth in the sector, by opening up new opportunities to start up companies, compete, and create jobs. Businesses and citizens will benefit from

¹¹ Art. 49 of the EC treaty.

greater choice of high quality services, increased market opportunities, and also better employment possibilities.

11 European Institute of Technology (EIT)

Innovation in services should be part of the European Institute of Technology (EIT) and it would represent a highly relevant activity in connection with this institute.

The EIT should respond to both societal and economic priorities and especially answer to the pressing needs of the European services sector. It is vital that the EIT is put in the context of delivering an education and research agenda relevant to the growing service sector and the overall importance of services to the economy and international trade. So maybe a technology focus is - at best - incomplete and we would like to see a far wider range of disciplines coming together - drawn from the business, legal and social sciences that affect business just as much as technology. After all it is the ability to *exploit* technology to deliver value - including allowing public services provide better public value - that will deliver the Lisbon agenda.

Similarly the EIT must not be understood as a substitute for continued focus on the other pressing challenges the EU faces to deliver the Lisbon agenda. It should contribute to enabling public and private organisational change, better regulation that enables business to innovate and creating the right environment for investment.

After achieving its' general mission to drive excellence and provide for a competitive and entrepreneurial spirit the EIT would be an ideal place to bring together all stakeholders and related constituencies to create a new discipline focusing on Service Science. Service Science should bring together ongoing work in computer science, operations research, industrial engineering, business strategy, management sciences, social and cognitive sciences, and legal sciences to develop the skills required in the services-led economy of the 21st Century.

NESSI believes the EIT should organise its activities around interdisciplinary issues moving away from traditional style of organisation around "silo" approaches. EIT allocation of funds should be based on excellence only. To ensure this the EIT should adopt an innovative governance structure with a large degree of autonomy. In particular, the EIT should set its own scientific agenda, avoiding any sort of political pressure. The European Commission should resist the temptation to appease different lobby groups, which are insisting that the EIT should solely rely on existing European structures without changing them. It will be insufficient to merely increase Europe's research funding or to build a networked constellation of (selected) existing institutions.

As a consequence, the EIT should have a structure able to exploit local value. Europe is large and diverse, and the structure should be adapted to that. We do have existing infrastructures, but not enough mobility, competition and exchange of best practice between these. Countries and individual universities should be empowered and encouraged to shape up their programs to strive for excellence and innovation in education and research and produce results that will be embraced by industry.

The EIT should focus on:

1. Integrating Innovation in services in the activities of the EIT and more focus should be put on services in the Knowledge Intensive Communities (KICs);
2. Driving collaboration between the academic/research world, governments and large-scale industry and employers to create a new discipline focusing on Service Science;
3. Addressing issues related to structural change/overregulation/better regulation;
4. Creating an autonomous governance structure, avoiding a heavy top down bureaucracy.

12 Conclusions

We need a better understanding of service sectors, the vital contribution they make to economic growth, and the need to promote innovation within the service sectors.

CONFIDENTIAL DRAFT – FOR DISTRIBUTION TO NESSI MEMBERS ONLY

It is crucial to understand the roles that different policy actors have in the design and delivery of proposed service innovation policy measures. In description of each policy measure, it should be made clear what the division of labour between the European Commission and member states is. Also joint and transnational activities should be clearly indicated when appropriate. Well co-ordinated national level, transnational level, and joint activities can support each other, and create favourable framework conditions for service innovation. The increasing trend of services growth also needs to be seen in the context of changing business models and increased globalisation.

Volume 2b: Strategy to build technological foundations for the service economy

[Target Audience:

*people who want to build (NESSI) the NESSI Open Service Framework including the writing of Vol. 3.x
Vol. 2 is a Technical Book.*

Scope

The version of the document delivered for the NESSI General Assembly (16 October) will be constrained by:

- Time,*
- Current knowledge,*
- Needs of the document audience for the purpose of the document.*

The time horizon concerns mid-term, i.e. (predictable) evolution over the state of the art (not considering possible paradigm shifts).

The what (functionalities offered).

The how: guidance to ... including:

- Skeleton architecture*
 - Adaptability mechanisms to any domain, technology and market segment*
- Principles (e.g. openness; adaptable to any domain, technology; **how to inject coherence? Glossary, other?**)*
- **How to assess the coherence of the whole?***

Process to build the book

Build and owned by the SRA Committee

Integrates inputs from NWGs.

What it is not:

- A roadmap*
- A market analysis*
- A design document, including functional architecture*
- A marketing document*
- Intended to discuss financing, define how to allocate projects, or make any points about a possible JTI*
- A guideline to implement the SRA]*

1 Scope of this volume

[This document specifies the technological concerns on how to build the NESSI Open Service Framework. This sections details the main point listed in the above 1.2 section.]

1.1 From technological foundations to technological challenges (pillars)

[From technological foundations (for the service economy) to (strategic) technological challenges (pillars). Setting the overall scene ... (provide some background vs. Foreground contextual information).]

1.2 Addressing the whole “Functionality” (foundation) continuum

[First stress coherency and consistency of the Technological foundation Functionality continuum,

Second map it to the Systemic one as stated in Vol. 2a.

Third provide the overall Big Picture (The Whole “Functionality” continuum) and stress contribution of Vol. 2b to it (as an enabler), also provide some additional rationale.]

[Missing input on this topic, to be developed]

2 Relevant Background

[This section will describe the research starting points and will be based on different angles such as:

- FP6 leverage
- NESSI Strategic Projects coverage and contributions from the NWGs
- Other relevant initiatives (e.g. relevant standards)]

2.1 Introduction

[Missing input on this topic, to be developed]

There are several open research areas that will be addresses in the development of the technology foundations. The next section 2.2 deals with Software as a Service and is detailed into a discussion on existing efforts on standardisation (section 2.2.1) which provides the context for the following section 2.2.2 which surveys recent studies on reference models and architectures. The three last main sections survey in the specific the state of the art according to the main layers of NESSI Open Service Framework “Functional Blocks”: Infrastructure (Sec. 2.3, Service Platform (Sec. 2.4), Service Consumer (Sec. 2.5).

2.2 Software as a Service

Software as a service (SaaS) is a growing movement amongst leading software vendors. **The overall approach of NESSI in the development of the “technology and systemic foundations for the service economy”, particularly for what concerns the technology foundations includes the needs of such approach and goes beyond to implement new paradigms for “software-intensive services”.** Over the last 2-3 years SaaS was introduced to exploit the ability of high speed Internet connections and the capability of modern web browsers to provide software applications as services to the users. In this emerging paradigm, the software vendor designs a web-native application that is hosted and operated at a data centre (possibly through a third party). The software is available over the Internet on-demand for the user to be accessed at any place and any time. The software performs a service for the user and the use of the software is delivered as a service and most often paid for electronically according to how much it is used.

Vendors typically use a multi-tenant architecture meaning that multiple customers run the same application which shares the same infrastructure which is virtually partitioned. SaaS is a disruptive trend that is reshaping the landscape of software and services. SaaS can be seen as an evolution of previous approaches to application service provision based on on-demand and utility computing.

Among the SaaS giants we find Google, Amazon, and eBay, with an increasing number of new players such as salesforce.com, WebEX, etc. Google is an ad-centric company, that is, it makes money through targeted advertisements shown in its services such as web search, e-mail, office applications. Google is also starting to make business through premium services (e.g. premium Gmail) with defined service level agreements (SLAs). Other SaaS stakeholders act as mediators between sellers and buyers (Amazon, eBay, eTravel companies, etc.). Major software vendors are moving to the SaaS delivery model. Microsoft has taken the first steps towards SaaS with Office Live (competing with Google Apps For Your Domain).

The great advantage for users is that they do not have to bother about typical administrative issues such as security, reliability or scalability. Instead of this, they can simply rely on a trusted service provider. In addition to that, users can be sure to always use the latest version of the software. Moreover, well defined and agreed SLAs allow transparent service costs for the users.

SaaS providers do not generally use heavyweight enterprise software for their applications. Enterprise software often responds to increasing load by scaling up (using large SMP servers). This approach has been successful in controlled enterprise environments. Scaling out (using clusters of off-the-shelf hardware) is an alternative approach that can offer greater flexibility and reduced equipment costs. Large SaaS providers are making good use of clustering and lightweight architectures. Two widely used architectures are LAMP and lightweight Java. Most of the first instances of “Web 2.0” like Flickr, MySpace, Friendster, FaceBook, etc. rely on LAMP. LAMP stands for Linux, Apache, MySQL, and PHP/Perl/Python. Other big SaaS, like eBay and e*TRADE rely on the so-called lightweight Java stack. Basically they move away from the full-blown J2EE and rely on some open source instances of components of this stack, such as Tomcat, Hibernate and Spring. In summary, large SaaS providers today are adopting off-the-shelf, typically open-source, and lightweight software stacks and then design their applications to scale for large number of clients based on them. Internally, they have their own proprietary infrastructure to run all these applications in large data centres.

Some SaaS providers are offering SaaS infrastructure services. For instance, Amazon offers computing, storage, and persistent queue services accessible through web service interfaces. Amazon S3 storage system offers low cost highly reliable storage. The client pays a quota per used GB/month and per GB transferred. Amazon EC2 provides a virtualized environment in which to run clustered applications. EC2 allows using from a single site to as many sites as needed. Customers pay on machine-hour used and bandwidth used between Amazon hosting environment and the external world. Amazon SQS offers persistent queues for web service messages, providing reliable storage for messages transiting among computers separated by a WAN.

Digging deeper in the infrastructure of large SaaS providers, there are three key aspects that correspond to the traditional tiers of multi-tier architectures: presentation, computation, and storage. The scalability of the presentation tier is not especially challenging, so we will concentrate on the other two. We will focus on the Google infrastructure, since it is better documented. Regarding the computation layer Google uses MapReduce, a framework to perform parallel computations in a straightforward manner. MapReduce is based on functional programming list iterators. Map transforms a list into another list by applying a function to each element. Map enables to perform a large computation in parallel. For instance, the counting of all word occurrences in a set of documents is distributed across a set of servers, each of them counting the occurrences in 1 or more documents. Then, reduce takes the aggregation of results of the map stage and computes the final result of the computation. In the provided example, the reduce stage just aggregates the results. MapReduce middleware takes care of distributing the jobs of the map, tolerating failures, and also collecting the results at another site that performs the reduce aggregation.

All above approaches are undertaken for developing and delivering their applications. However, NESSI goal is more ambitious. NESSI aims at creating a technology foundations for creating and delivering applications not within a service provider, but also by third parties, enabling the creation of service based ecosystems. These foundations will enable a service provider to offer to third parties the development of applications to be hosted by the service provider. One of the few providers adopting this approach is Salesforce. They have built a framework, AppExchange, to create SaaS applications hosted by them. The first version of this framework is far in functionality from what can be done with traditional enterprise software. We will go beyond the capabilities of AppExchange by providing a richer service platform for creating, hosting, running and operating SaaS applications.

2.2.1 Standards

Standardisation in SOA is currently very dynamic with literally hundreds of standards, often in competition. The situation is complex because of many reasons: a myriad of available standards, overlapping functionality, lack of interoperability among them, standards abandoned before being adopted. In what follows we identify the relevant standardisation bodies and the current situation with SOA standards and then discuss how new research should plan to address standardisation and make sense out of the crowd of existing standards. In here, we do not survey SOA standards, since in our current study there are about 200 standards that might be relevant to NESSI (see section 6 Appendix A).

The main standardisation bodies¹² in SOA considered in this document are: IETF (Internet), W3C (XML and core web services), OASIS (web services), WS-I (web service interoperability), OSOA (SOA), JCP (Java and middleware technologies), OMG (middleware), WfMC (workflow).

According to a recent study from Forrester Research [Forrester06] there are only a few SOA standards that are widely adopted:

- Messaging: SOAP 1.1 and SOAP messages with attachments.
- Service interface definition: WSDL 1.1
- Interoperability: WS-I basic profiles 1.0 and 1.1.
- Security: WS-Security and WS-I basic security profile.
- Orchestration: BPEL4WS.

The study also identifies some standards being used but not widely adopted:

- Registry/metadata: UDDI, WS-Policy, WS-Metadata.
- Advanced security: Kerberos (combined with SOAP with attachments), WS-SecureExchange, WS-SecurityPolicy, WS-SecureConversation.
- Workflow: BPEL4WS extended for people, WS Choreography, WS Eventing.
- Management: WS-Distributed Management, WS-Management.
- Reliable messaging: WS-ReliableMessaging, WS-RM Policy
- Service Bus/Middleware: JBI, SCA.

This means that from the several hundreds of available standards for SOA there are very few that are widely extended or significantly used. One of the major contributions of NESSI in the area of standards, while building the technology foundations for the service economy, will be the identification of relevant standards and the selection of a coherent set of them. The selection will be driven by the need of having a set of standards that are sufficient for the NESSI vision and purposes in terms of functionality. The principles for selecting them will be the expected functionality to be provided, interoperability, and stability; existence of open source software of sufficient quality will be also considered. For sure, there will be gaps and incompatibilities in the selected set. The task of NESSI SRA is to propose gaps to be filled in standards, address lack of interoperability among them. The NESSI Standardisation Committee will coordinate the identified task in order to reach these goals.

¹² including some organisations which produce specifications in a collaborative manner, but do not implement a formal standardisation process.

2.2.2 Reference Models, Architectures, and Implementations

Reference models, architectures and implementations have become essential instruments for describing and leveraging open (distributed) systems. There are reference models that are already several decades old such as the OSI reference model. Here, we are interested in reference models, architectures and implementation in the area of service-based systems.

We will now focus in the three main **reference models and architectures** in this area, namely the OASIS reference model for SOA, the W3C reference architecture for web services, and the reference model and architecture applicable to Grids namely the one proposed by the Enterprise Grid Alliance for enterprise grids now merged with the Global Grid Forum (GGF) in the OGF – Open Grid Forum.

OASIS proposes a **Reference Model for SOA**. Since SOA (Service Oriented Architecture – a detail discussion of SOA is in Sec. 2.4.2) is being used as a term in an increasing number of contexts and specific technology implementations, and very often, the term is used with differing – or worse, conflicting – understandings of implicit terminology and components, OASIS has decided to define a reference model for SOA to set a common terminology and model. This reference model is being developed to encourage the continued growth of different and specialized SOA implementations whilst preserving a common layer of understanding about what SOA is.

W3C has developed a **Reference Architecture for Web Services**. This reference architecture identifies the functional components, defines the relationships among those components, and establishes a set of constraints which affect the desired properties of the overall architecture. The reference architecture is intended to promote interoperability and extensibility among these various applications, platforms and frameworks in a manner that remains consistent with the architecture of the Web. The architecture was derived from a set of user requirements that were identified prior to the definition of the architecture.

In the **Grid area** two main initiatives investigate the development of reference architecture and models for Grids; they are respectively the OGSA-WG (Open Grid Service Architecture- Working Group) and the RM-WG (Reference Model-Working Group) both in the frame of the OGF Architecture Area.

The OGSA-WG has developed a reference architecture for Grids (named Open Grid Service Architecture) that is assumed to be as a blueprint for standards-based grid computing: "Open" refers to the process used to develop standards that achieve interoperability; "Grid" is concerned with the integration, virtualisation, and management of services and resources in a distributed, heterogeneous environment; it is "service-oriented" because it delivers functionality as loosely coupled, interacting services aligned with industry-accepted Web service standards. The "architecture" defines the components, their organizations and interactions, and the design philosophy used.

The RM-WG started in 2007 and has the goals of:

- defining a common glossary and set of terms in order to describe grids;
- defining more formally the sets of components that comprise a grid, their relationships and life-cycle.
- Developing pragmatically the broader model that brings together, references and extends where appropriate.

To achieve these goals, the RM-WG will use the EGA Reference Model as its starting point. The **Enterprise Grid Alliance Reference Model** provides a specific context for describing requirements, standards, comparing technologies and implementing grid solutions for enterprise applications, to evolve current data centres to a more dynamic and self-managed setting. The model delivers a framework and set of customer-based requirements needed to accelerate enterprise grid adoption.

For specific application domains, dedicated reference models have been created. In the e-government domain, for instance, different countries and organizations have defined their specific reference models. For example, The Netherlands have developed the “Nederlandse Overheid Referentie Architectuur” and the United States the “Federal Enterprise Architecture”. On the European level, there is the “European Interoperability Framework” with the aim of

enabling interoperability between e-government services. <http://www.w3.org/TR/2002/WD-ws-arch-20021114/> - [TAGDOC](#)

Propose to add a reference model/architecture for services engineering. The OMG Model Driven Architecture is a strong candidate. The Eclipse Modelling Framework (which can be regarded as a concrete implementation of the MDA reference architecture) should be mentioned.

The **main distinguishing feature** of the NESSI technology foundations with respect to these reference model and architecture efforts is that its broader scope will not focus on a particular application domain, technology or set of standards. Instead, NESSI aims at the development of a coherent, consistent and comprehensive reference model as well as architecture. In this regard, it is necessary not only to integrate concepts from existing standards, models as well as architectures but also to extend and refine them in order to meet the needs of the NESSI Open Service Framework.

Reference implementations are typically used to exemplify the implementation of a standard and help others in producing their own versions of the standard. Reference implementations therefore increase awareness and allow full understanding of all aspects of a standard and whether its realization is feasible. Also, a reference implementation – especially an open one – can help to jump-start the adoption of new technologies or standards, as a working implementation is available.

One of the major efforts regarding reference implementations has been made by Sun in the area of Java technologies. Most standards in the Java arena, including application servers and other service-oriented infrastructure, provide a reference implementation providing the full functionality of the standard to help software vendors to produce their own version of the standard. Some of the most critical standards have also an associated certification process, as it is the case with Java Enterprise Edition application servers.

In the Web Services arena, reference implementations have become a major instrument for many of the available standards. Some examples are: SOAP, ASAP, JAX-RPC, UDDI, ebXML Registry, WS-Security, XSLT, WS-RF and WS-Notification, WS-Modelling Ontology, etc. Some standards and reference implementations are also starting to take into account basic interoperability validated through WS-I and interoperability with other standards that are considered critical. For instance, JAX-RPC considers basic interoperability via WS-I and interoperability with WS-Security.

2.3 Infrastructure Layer

2.3.1 Adaptive Service-aware Infrastructure

The ICT infrastructure used in any enterprise of more than moderate scale is highly complex and consists of a wide range of computing and networking hardware, middleware products and software. Operational support systems and management tools are also essential components of the infrastructure. ICT administration and management is expensive and it is hard to establish a sound relationship between business requirements and IT capabilities [NESSI-Grid 07]. Typical practices result in systems that are difficult to modify and which therefore limit the ability of an enterprise to adapt its processes to respond to changing market conditions.

The last few years have seen the development of a range of distinct approaches to improving the flexibility and manageability of ICT. A prominent paradigm in this area can be described as virtualisation which aims at involve breaking the tight coupling between an application and the specific resources it requires, but this can be achieved in a number of different ways, involving innovation in processor architecture, operating systems and software.

Selected virtualisation areas are sketched in the following. Network virtualisation in the form of virtual private networks (VPN) and virtual local area networks (VLAN) allow different users or applications to be isolated from one another even though they share the same physical connectivity.

Mobile networks are an important connectivity means for nomadic users. Mobile network performance is difficult to predict and vary greatly with respect to the location of users. Mobile ad-hoc network represent a form of mobile

network virtualization that promises to mitigate the connectivity issue in congested area by allowing each network participant to act as both a client and a provider of the networking service.

Storage virtualisation supports hierarchical storage management (HSM) and information lifecycle management (ILM) which are becoming increasingly important as data volumes explode in both commercial and scientific applications.

Server virtualisation includes both clustering (building a large virtual computer from a number of smaller ones) for horizontal scaling and partitioning (running several independent instances of an operating system on the same computer).

Component containers (such as application servers, EJB containers, web servers, etc.) can also be regarded as offering virtualisation – components written for a particular kind of container can be deployed to any instance of the container, regardless of the underlying server architecture or operating system.

Other important paradigms centre around autonomic computing and model-driven system management.

All of the above approaches can contribute to building flexible ICT infrastructure that can be adapted to the changing requirements of applications. There are many examples of enterprises obtaining significant business benefit from more efficient use of their internal resources by exploiting virtualisation and other related technologies.

The NESSI Open Framework is about more than cost savings within individual enterprises. The real benefits to competitiveness will come from enabling an open market in the SaaS arena. From a technical point of view, this emphasises the importance of **service interactions across organisational boundaries**. There are several scientific challenges that must be addressed before infrastructure services can be effectively deployed and used by multiple independent stakeholders.

From the point of view of a service consumer, interface definition is inadequate in most current approaches to building distributed systems (such as Web Services, CORBA, etc.). These generally focus on defining the protocols required to invoke some remote functionality. Description of service functionality is informally specified in documentation and so not suitable for automating service selection. Non-functional characteristics of services are also not dealt with. Performance, availability, compliance with security and regulatory policies etc. are generally not specified. Standard ways of naming and describing infrastructure resources are required so that the requirements of applications can be specified independently of how the infrastructure is configured. These descriptions will form the basis of SLAs associated with infrastructure services. The description of an infrastructure service, as specified in the relevant SLA, must contain sufficient information for a consumer to be able to predict the behaviour of a composite infrastructure built of services from multiple, independent providers.

From the point of view of a service provider, infrastructure must be manageable so that services can be offered with assured non-functional characteristics to multiple customers. Efficient management will be essential for a provider to be competitive in the market for infrastructure services. Consistent approaches to management that are applicable across different technologies, products and vendors remain a challenge. A service provider must be able to achieve an acceptable balance between risk and reward in the way available resources are deployed. There is a natural tension between the commitments made in the set of SLAs offered to customers and the cost of providing the resources to meet these commitments. Effective modelling techniques are required to support decisions about what SLAs can be offered and for capacity planning.

Mobile network quality of service raises particular difficulties in the sense that service level offering, which are today specified at a large geographical level do not reflect individual user's experience for whom the service level depend on their exact location. Mobile network may have a very good overall coverage but fail to adequately cover certain specific geographical area. Providing SLA for mobile network should be done at a user per user level since the offering may vary greatly from one user, with its specific geographical area of concern and another one.

A detailed analysis on these infrastructure issues can be found in the strategic research agenda from NESSI-Grid [NESSI-Grid 07].

2.3.2 Data Management

The following paragraphs present a short overview related to the areas defined above. The first two topics are very much related to replication of application servers and databases. Therefore we concentrate on these two topics, at the same time stressing again that a disruptive approach is to think about a vertical solution instead of a horizontal one,

2.3.2.1 Application Server Replication

Basically all replication approaches we are aware of, assume a shared database or horizontal replication (i.e., independent replication of the tiers). Clustering is a facility provided by many J2EE application servers. However, current approaches focus on replication of session beans and rely on a shared database that becomes a single point of failure, and additionally the bottleneck of the system. Additionally, many of the existing clustering solutions are not transaction-aware which results in inconsistencies in the case of failover.

Examples for industrial products include JBoss open source J2EE AS [JBoss] (providing clustering facilities for session beans), Oracle9iAS [Oracle] (replicating sessions after each method invocation, not replicating entity beans), WebLogic clustering [WebLogic] (propagating the state of sessions to backups after each method invocation). The latter is one of the few approaches that is transaction aware however since changes are propagated after the method/transaction finishes, inconsistencies may arise if there is a failure. The Data Replication Service is in charge of handling replication in WebSphere 6.0 [WSv6]. It uses reliable multicast and changes can be propagated synchronously or asynchronously. Sessions accessed inside a transaction will not be replicated until that transaction completes providing transaction awareness. All described products rely on a shared database with the database becoming a single point of failure and the bottleneck of the system if not replicated, or not running on powerful hardware and not having special fault-tolerance features.

2.3.2.2 Database Replication

1-copy serializability: An important categorization for database replication is whether updates that occur at one replica are propagated lazily, i.e., after the update transaction commits, to the other replicas, or eagerly, i.e., some time before transaction commit. Lazy replication provides fast response times, especially in wide-area networks, since transaction response time does not include any inter-replica communication. However, using lazy replication, data might be stale or even inconsistent. Lazy replication has been explored from many different perspectives [Breitbart99, Daudjee05, Holliday03, Rohm02, Pape04, Akal05, Gañçarski07]. In eager data replication, replica control can be either pessimistic or optimistic. In the former, potentially conflictive transactions are not allowed to execute in parallel which is typically achieved via locking. In the latter, transactions are freely executed. At the end of a transaction there is a validation phase for each transaction in which a transaction is aborted if a conflict occurred that violates the correctness criterion. Optimistic approaches might lead to higher abort rates while pessimistic approaches lead to less concurrency. Database systems internally often use pessimistic approaches, and so do many replication protocols, e.g., [Amir02, Amza03, Cecchet03, Patiño05, Pacitti05].

1-Copy-Snapshot Isolation (1CSI) and other Snapshot Isolation (SI) approaches: More recently, some protocols have explored extensions of snapshot isolation for replicated environments including [Daudjee06, Kemme05, Lin05, Elnikety05, Plattner06]. 1CSI formally defined in [Lin05] has shown great scalability potential for the database backend tier. [Elnikety05] proposes a more relaxed form of 1-copy correctness based on SI termed generalized prefix SI. [Plattner06] applies 1CSI to database farming, and lazy replication. [Daudjee06] proposes a relaxed version of 1-copy correctness based on SI termed strong session snapshot isolation.

Partial Replication: [Holliday02] proposes an epidemic protocol for partially replicated databases. The protocol aims at a WAN environment. Each data item has one or more permanent sites that always have a copy of that data item. Other sites may have a temporary cached copy. The correctness criterion is 1CS. Readsets and writesets are propagated to maintain consistency. If a data item is not stored at the site where the transaction executes, a request is sent to one of the permanent sites and propagated with the associated lock table information. 1CS results in a high number of aborts due to read-write conflicts what severely limits the scalability. The database state machine [Pedone03] as aforementioned proposes an optimistic protocol for database replication that provides 1CS. This protocol has been extended to partial database replication in [Oliveira06]. Partial replication has also been studied in

[Cecchet03]. The model of replication is that some sites do not store the whole database. Transaction requests are parsed in order to determine the sites where the transaction can be executed. This may lead to full replication when there are complex requests. The correctness criterion is also 1CS.

2.3.2.3 Self-healing

There exists a large body of research in regard to fault-tolerance and recovery in distributed systems. Most of the work focuses on how to handle the failure of components, and how to guarantee that the component taking over has the accurate state of the failed component as of the time the component failed. Techniques such as passive or active replication are common. In other approaches a component checkpoints its state on a regular basis and logs events after. At failure time, a new component is started and reconstructs the state. This restart is also often called recovery. When we discussed related work regarding application server replication several of these techniques were mentioned.

A recent initiative dealing with recovery is recovery oriented computing (ROC) [ROC]. In this initiative, it is recognized that systems are not free from failing and the proposed techniques aim at attaining a fast system recovery via a recursive start. Smaller components are typically lighter to restart, and therefore in ROC, in the advent of failures restart is first applied to the smaller component involved in the failure. If the restart is not successful, restart is tried on the next containing component, and so on.

Little work, however, has been done on letting failed replicas rejoin or new replicas join a running replicated system. We refer to this as online recovery. [Kemme01, Jimenez02] settles the principles upon which a quiescent state can be extracted from a database in an online manner and transferred consistently to a new site. So far, these approaches do not directly work with a vertical replication approach that has to consider the data of both tiers. Also partial replication is ignored.

2.3.2.4 Self-optimizing and self-provisioning

Most research on self-* protocols has focused on database and web-server tier. [Milan04] explores self-optimization for database replication. Each replica decides locally on the number of requests to execute concurrently. At the global level, the type of requests to be executed at each replica is adjusted whenever imbalances incur. [Amza05] exploits several load-balancing strategies to distribute the load over a replicated database cluster. In [Chen06, Soundararajan06], the authors explore provisioning techniques for database replication. In [Gançarski07], the authors explore load-balancing strategies for a database system with lazy replication, and take into account that some transactions might be willing to read data that is not completely fresh. [Zhang05] describes a load-balancing mechanism for web-servers where they assign each web-server replica documents of similar size. There exists less work on application server optimization. Commercial systems often use very simple approaches such as round-robin.

2.3.2.5 System Infrastructure and communication support

Recently, there has been a lot of progress in high-speed communication at the 10Gbit/s-range over commodity interconnects [MPB07, PKK07, MXP06]. This progress relies to a large extent on previous research in low-latency high-bandwidth interconnects for commodity systems with important contributions from [ZBJ02, ABD98, DBL97]. This progress has mainly targeted to understand and eliminate CPU overheads in high-performance communication protocols over certain interconnection network architectures and mostly for scientific applications. Moreover, there has been an effort to introduce spatial parallelism at the link level in a transparent manner but also to extend communication APIs so that new applications can take explicit advantage of out-of-order packet delivery. Furthermore, there has been research activity in providing efficient communication protocols for networked disk I/O and supporting higher system layers that provide storage virtualization. These developments make it extremely timely to consider how infrastructures should be designed to scale to much larger sizes than has been possible today in terms of the communication subsystem.

2.4 Service Platform Layer

2.4.1 Service-Centric Systems Engineering

From an economic perspective, the time to market and the reuse of valuable business logic are highly desirable. Service-oriented architectures aim at improving the flexibility and agility of business processes. Service-oriented architectures are aimed at distributed applications that require to integrate heterogeneous systems and that aim at platform independence. They are based on contracts, dynamic binding and message-orientation. There are already well established key principles for service-oriented architectures, namely loose coupling between services, coarse grained service interfaces, dynamic service discovery and binding, self containment of services, service interoperability and protocol independence [Erl04], [Kaye 06]. From an economic perspective, the time to market and the reuse of valuable business logic are highly desirable. Service-oriented architectures aim at improving the flexibility and agility of business processes.

Quality properties (performance attributes, security, scalability, transactional semantics, etc.) are of paramount importance in the application scenarios that are addressed by the NESSI Open Service Framework. A common attribute of quality properties is that they are cross-cutting aspects affecting most of the functionality of service and/or service infrastructure code. With traditional software development techniques the code of cross-cutting aspects is scattered and/or tangled, making it hard to understand and maintain. One of the challenges is how to engineer services with quality properties providing separation of concerns. Separation of concerns entails breaking down the server code into distinct parts that overlap in functionality as little as possible leading to independent “components” in which each cross-cutting aspect is encapsulated.

Service composition across different institution or different application domains, which includes crossing technical boundaries, is not yet happening for the broad range in the software industry. Examples that stick out of the mass can be found in the area of general information services and ordering of mass customer goods e.g. Google, Amazon, or EBay. For a wider spectrum of application areas it remains a challenge to support a controlled and predictable end-to-end service provision for services whose intended usage context is not fixed and is not completely known at development time. Some standards for service composition like BPEL are being widely adopted. However, choreography standards like WS-CDL [W3C] have rarely been adopted. There are also gaps not covered by standards such as controlling and monitoring mechanisms. Another unsolved challenge is to describe functionality and quality in a way that allows for precise matching of requirements and to use standards in a way that it is truly technology independent and therefore also vendor independent.

Requirements for a service cannot be fully captured in advance. It is not realistic that one solution fits all. Flexibility and capability to quickly react to new requirements is a key challenge and an important factor for staying competitive and for creating sustainable long living systems. Services have to be designed for reuse, adaptation and evolution, and the business workflows composed of services need to be maintainable and adaptable. It is therefore of utmost importance to identify design principles, guidelines and best practices and leverage sound software engineering methodologies for well-structured service ecosystems.

2.4.2 Service-Oriented Architectures (SOAs)

Since the late 1990s, many **definitions of SOA** have been published [Alonso04, Frost98]. The normative OASIS Reference Model [MacKenzie06] defines SOA as “...*a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains. It provides a uniform means to offer, discover, interact with and use capabilities to produce desired effects consistent with measurable preconditions and expectations*”.

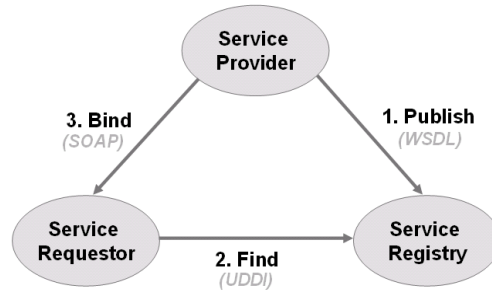


Figure 1: Basic SOA Architecture

According to this model, the major components of a basic SOA and their possible interactions as depicted in Figure 1 are: A **service provider** publishes his service interface via a **service registry** where a **service requester/ consumer** can find it and subsequently may bind to the service provider. The central concept of the SOA Reference Model is the existence of services which provide access to capabilities by well-defined interfaces to be exercised following a service contract with constraints and policies. This enables a loose coupling of services (thereby minimizing mutual dependencies) and complies with some of the probably most-known principles in software-engineering, information-hiding and modularization [Parnas72]. Services are provided by entities, the service providers, and are to be used by others, the service consumers. Services may be composed on the basis of other, existing services, thereby adhering to the principle of reuse. They are autonomous (solely control the logic they encapsulate), uniformly described and publicly retrievable via certain discovery mechanisms.

The concept of supporting loosely coupled, business-aligned and networked services as introduced above can be realized with the help of numerous different technologies, such as Web Service Description Language (WSDL) and **SOAP-based Web Services** which are the most widely spread standards used to setup SOAs [Alonso04]. WSDL¹³ defines a uniform, machine-readable XML format for service interfaces. The Universal Description, Discovery, and Integration (UDDI¹⁴) standard specifies publicly available service registries (see Figure 1) that are needed for service search, identification and invocation. SOAP specifies the data format and an exchange protocol for the messages to be sent between service providers and requesters. While SOAP only defines formal aspects of data structure, the interpretation of information semantics has not yet been defined by a common standard. Machines may describe the same piece of information with different terms, thereby preventing from seamless system interoperability. The Business Process Execution Language (BPEL¹⁵) is widely accepted as a standard for imposing different services orchestration. Also based on XML, it focuses on the description of a business process from one participant's point of view and enables defining a business process as a set of coordinated Web Services interactions. In this way BPEL generates new, higher-level services whose functionality are the union of the functionalities of the Web services called.

¹³ <http://www.w3.org/tr>

¹⁴ <http://www.oasis-open.org/>

¹⁵ <http://www-128.ibm.com/developerworks/>

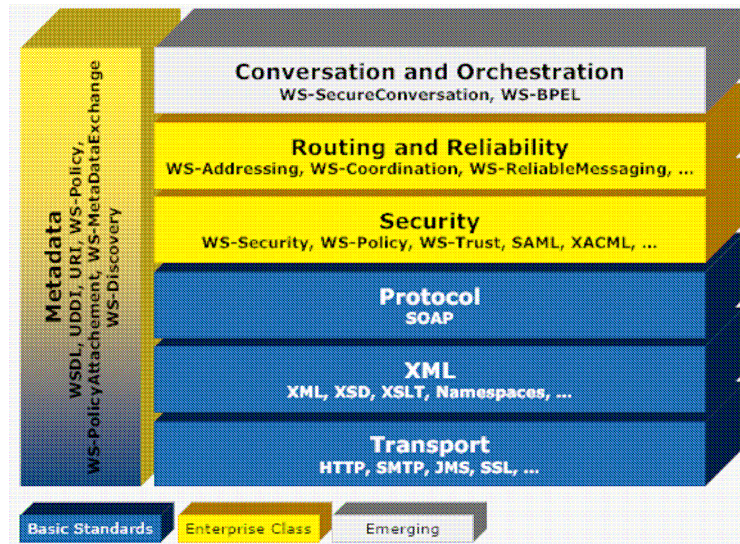


Figure 2: Web Services Stack with a selection of related standards

In the last years, many have generated great **enthusiasm about Web Services**, which are considered as “open, Internet-era standards for exchanging data between applications” [McAfee05]. With a SOA, tight couplings between the applications of different companies are expected to be replaced by loose couplings. As a consequence of common standards for data description and connection protocols, applications can be seamlessly connected to others without costly reprogramming efforts. By loosely coupling application functionality via commonly defined interfaces, companies are enabled to modify their operations more agilely in response to changing business requirements. Summing up, especially companies expected the dawn of Web Services to turn traditional supply chains inside out and to replace static cross-organizational business relations by a flexible mesh of loosely coupled services which can be identified via publicly accessible registries.

Right now, **Web Services have not lived up to this promise** as mostly people are still involved in loose couplings all over the world (and thereby increasing complexity and limiting the degree of automation). Rather than automatically inter-connecting applications across company boundaries with the help of Web Services technologies, most loose couplings are based on “e-mail and instant-messaging applications, groupware such as Lotus Notes/ Domino and nascent “social software” such as wikis, Friendster.com and LinkedIn.com” [McAfee05]. As argued in [Chen01], the “corporate household” problem prevents Web Services-based SOAs from being setup quickly and easily.

State-of-the-art SOA implementations (based on WS-* protocols) must be aware of **semantics and syntactical descriptions** of exchanged data. Philip Merrick, former chairman and CEO of webMethods, has described part of the underlying problem in the following way [Merrick03]: “...there’s a whole other layer to deal with, what I call the semantic integration problem. Web services are great but they standardize pure connectivity between applications. The applications still have highly varied data models, extremely different ideas of what business processes should look like”. In other words, the promised Web Services technological independence has not become a reality. Since two entities must implement the same stack of protocols in order to interact, the technological coupling has been translated to the current Web Services protocols stack. Consequently, this issue can be seen as a higher level technological coupling.

Also, companies and their respective systems are subject to rigid **governance mechanisms** [Weill04] and adhere to different conventions with respect to business documents and also internal business processes: “The applications used in different departments or different locations of a single company, for example, are virtually guaranteed to have inconsistency in customer lists, part-numbering schemes, approved suppliers, addresses for these suppliers and so on“ [Chen01].

Besides the heterogeneity of conventions with respect to data structures, semantics and processes, the **lack of** comprehensive, trustworthy and **widely accepted service registries** prevents global SOAs from quickly establishing.

The few globally available UDDI-based registries are mostly only usable for **technical experts**, do not offer any additional functionality such as service performance monitoring and only feature a small number of references to different Web Services.

Apart from that, the traditional Web Services stack aims at supporting the setup of loosely coupled application interconnections especially in a professional context and assumes users to be technically sophisticated. This prevents normal Internet users from easily discovering and interacting with services available via the Web. WSDL-based service interfaces, but also the UDDI-compliant service registries are not designed to be interpreted and used by humans but rather by machines.

2.4.3 Business Process Management

Enterprises currently and in the past have dealt with interoperation and integration in a variety of ways, but more in a centralized, monolithic and point-to-point way with control being done by the Enterprise and not by the partners (whether small or medium businesses) in the network. This sort of interaction has significant initial costs, risks and growth issues, and subsequent costs and risk when trying to reorganize or realign the network of partners and suppliers (Interoperability Roadmap reference¹⁶).

Substantial research on Web Services interoperability, integration of applications, interoperability of large and medium companies, semantic interoperability, virtual organizations, networks of SMEs and others mainly focussed on protocols, SLAs, Web other key items. Several studies have been conducted on interoperability in Enterprise networks, SLAs, Integration methodologies and some level of adoptability of the protocols and services.

The gales of innovation, of disruptive new technologies and of intensifying competition are forcing value networks across the world to keep re-inventing themselves at an accelerating pace. As disruptive technologies keep creating new niche markets, old value networks may be dismantled (Christensen's bestselling textbook "Innovator's Dilemma"¹⁷), and old companies have to swiftly form new alliances, moving into new markets, or face extinction or acquisition by nimbler competitors.

Companies are therefore forced, on a continuous basis, to perform the following tasks; a) Track Key Performance Indicators (KPIs) such as revenues and costs, market share and customer satisfaction index; and b) Track future market state indicators, or emerging new technology market indicators. Several research initiatives, including FP6 research projects over the last 10 years have focused on interoperability (e.g. ATHENA, DBE, TRUSTCOM and others), and several standards bodies such as BMI, OASIS, BMMP and others have substantial activities and working-groups on interoperability. The state of the art in this area can be classified in the following coarse grained way:

Interoperation at the Technology Level

1. Invasive model of interoperation, which is currently being done in most enterprises and in most environments, where specific applications are connected and integrated in point-to-point from an enterprise (e.g. an OEM) to applications managed by its partners, suppliers and others.
2. Non-invasive model of interoperation, which is being practiced by some enterprises through well-established standards and services. In this model, specific applications are automatically integrated to services and application-services offered by partners and suppliers. In this model, the semantics of the interaction and interoperation is well-established and defined before the application interoperability is done.

¹⁶ Enterprise Interoperability Roadmap, 2007.

¹⁷ Innovators Dilemma, 2006

3. Centrally managed interoperation: Most enterprises still consider a centrally managed model of doing business integration and interoperation across the enterprise network and boundaries. Models of third-parties have come into existence (e.g. supplier side marketplaces or messaging integration hubs or centers).
4. Ad-hoc connectivity interoperation: This is currently becoming the modus-operandi of most companies world-wide, where application systems across enterprises in a value network are connected on a need-basis, using point-to-point and sometimes ad-hoc methods, which do not scale.

Interoperability in the Business Network

1. Business Protocols for interoperation: For most businesses, the specifics of interoperation at the business level have been done manually first with agreements on specific protocols for interoperation, and agreements on specific SLAs and policies. Without these manual agreements the business interoperation is not done. The automation of the business protocols for standard business functions such as contract establishment, purchasing or others has not been done, and is much needed in the future.
2. Semantic business models in interoperation: Semantic notions of the collaboration and sharing of critical business terms and operations across the value network.
3. Policies and governance in the network: One of the biggest open areas and concerns is the governance model for processes over service value networks.
4. Interoperation globally across partners and suppliers world-wide: Interoperation of multiple partners globally across international boundaries, legal policies and local municipal policies.

2.4.3.1 Concepts and Models for Service Value Networks (SVNs)

Formation and management of a SVN involves three main aspects:

1. identification of the appropriate form of SVN in terms of SVN partners and of scope and structure of interactions;
2. identification of how much beneficial it is for a party to join a network;
3. definition of the contractual agreements among members.

In all these areas, current state of the art mostly focuses on defining languages and protocols for negotiations among organisations and for the modelling and standardisation of SLA definitions [Camarinha-Matos99]. While some research results are promising, current work did not gain acceptance in practice and it covers only a small subset of the scope of SVN formation and management. In particular, what is missing is a set of effective methodologies, best practices, standard concepts and semantics, and tools for the identification of value networks, for the semi-automated analysis of the benefit a partner has in joining a network, and for the formation of contracts.

2.4.3.2 Concepts and Models for Collaborative, Cross-organisational Business Processes

This area of research is related to the operational interaction among Service Value Network partners, in terms of cross-organisational processes. Over the last decade, there has been a lot of research in business process modelling, advanced transaction models for business processes, and architectures and systems for business process management and enactment. In the meanwhile numerous commercial products have appeared. However, the design of business processes is still intra-organisation focused, and standards for business process design and for organisational structures are only partly addressed. Recent attempts to address inter-enterprise collaborative business process management have resulted in consortia such as ebXML and RosettaNet starting to introduce standards for inter-organisational processes.

Collaborative business processing in dynamic SVNs brings about inter-organisational scenarios where no common transaction semantic, transaction context representation and coordination protocol can be assumed to exist. The challenge still to be solved is how to achieve context representation and service coordination in a decoupled, decentralised manner, and in accordance with the collaboration agreements between the SVN participants and their local policies. Designing business processes for networked organisations, together with the challenge of reducing

collaboration expenses, will enable the networked economy. The challenge is particularly hard in Europe, because of different legal aspects in different countries throughout the continent.

2.4.3.3 Concepts and Models for Adaptive and Autonomic Business Processes

The key aspect of adaptive processes and semi-automated support for managing changes implies a need for autonomic process management, where business processes, both the inter-organizational ones that form the Service Value Network and the internal ones that each partner needs to execute as a result of SVN interactions, need to evolve in a synchronized fashion and in accordance with SVN contracts and with each partner's business goals. In this area, research has focused mostly on dynamic service selection and automated derivation of service composition models from high-level goals. While existing results in these areas will be inspiring, the challenge here differs from what was done in prior research as the focus is to try to dynamically drive and optimize a collaborative process which is not owned by a partner but connected based on policies through a SVN, and that has to be driven by the objectives of the SVN and of all its members.

2.4.3.4 Business Process Management and Analysis Platforms

In terms of state of the practice, there are many currently available systems that provide support for BPM. On the modelling and execution side, IBM's WebSphere Business Components Studio provides solutions developers with the tools for creating applications for solving particular process problems. Microsoft's .NET framework and BizTalk Enterprise server support cross-domain business systems integration, through XML and Web Service technology (based on open standards – e.g. OASIS, W3C, XML and others). Dozens of analogous products, with different levels of sophistication and quality, exist today on the market. Many of them (including the ones mentioned above) can achieve cross-organizational interoperability by supporting Web services standards. Such tools do not provide support for adaptive or autonomic processes, and also do not support any specific support for modelling SVN processes that are aware of SVN policies, contracts, and SLAs.

In terms of execution analysis and improvement, related tools include generic business intelligence platforms such as Business Object XI or specific process data analysis tools such as SAP's Business Intelligence and Business Warehouse, which together facilitate fine-grained process data analysis. These tools however do not provide automated support for deriving models correlating SVN indicators and SLA violations with their root causes and for identifying changes to the defined processes. Furthermore, they typically allow only the analysis of processes executed by a given process automation system, as opposed to being able to collect and analyse process events and data possibly coming from multiple sources, which is a necessary feature in a multi-organization environment.

2.5 Service Consumer Layer

2.5.1 Advanced User-Service Interactions

User service interaction technologies can be considered from two different viewpoints: the human-service interaction view and the engineering view.

Human-service interaction view

Human service interaction is mediated by different devices (such as keyboards, screens, mouse, sensors, speakers, microphones, video cameras, etc.) providing media and multi-media interface modalities (text, graphics, voice, images, etc.). Significant progress of the presentation layer has been achieved through innovative middleware and supporting tools for visualization and direct acquisition and manipulation of service as well as application information. Current web based interfaces demonstrate significant achievements in this respect. However usability of services is more and more dependent on the layers behind the interface, which control the visualization process and the interpretation, analysis and assimilation of information according to the specific users' contexts.

Existing solutions range from platform independent formalism to represent user-system dialogue (e.g. UI interaction languages, UML interaction diagrams, dialogue grammars, interaction patterns), to domain dependent systems which take advantage of domain semantics and user modelling (e.g. computer games, simulation systems). While generic

models are suited to systems offering limited functionality, complex systems will require easily understandable and accessible interfaces based on knowledge-rich interaction models integrating knowledge about the user, the service and the user context, as well as the specific domain semantic.

A brief summary of the most relevant approaches follows:

User modelling aims at gather all the relevant information that a system needs to know about the user. Existing techniques addresses static aspects such as profiles, preferences, resources, etc. Managing dynamic aspects such as user goals, ongoing tasks, user situations, previously executed actions, etc. still are an open issue. NEXOF will propose the use of existing user model ontologies and representation-formalism-based Semantic Web technologies such as RDF(S)¹⁸, OWL¹⁹ and SWRL²⁰ to enable reuse and sharing of user data between different applications and services.

Context modelling is a challenging task due to the vagueness of its definition [Dey00] (“Context is any information that can be used to characterize the situation of an entity”). However the representation and management of the context model is essential for developing “smart” interfaces which enable the flexible, adaptive and proactive communication with the user. The concept of “Context computing” [Moran01] has emerged with the aim to incorporate context awareness into software systems. This includes the capabilities to detect contextual information, to represent it, to manipulate it, and to influence the behaviour of applications with the goal to support human users in a proper and personalized way. Proof of concepts prototypes have in different area such as e-Business, e-Learning, Healthcare, Mobile Communication Networks and many more. Specific examples are: AmbieSense where contextual information is used to provide personalized, context-sensitive information to mobile, “nomadic” for the adaptation of technical systems. “MILK” addresses context-awareness for information sharing in distributed organizations. XEROX WID (Web Information Discovery) aims at automating the “Hidden Web” discovery process. Unresolved issues inherent in contextual computing comprise the following:

1. need for a homogeneous, interoperable description of contextual information. Beyond the wide variety of existing approaches, there is a broad consensus on the information which is needed: the user situation (e.g. by who the user is, where the user is, what the user is doing, when the user does it, and on which the user focuses), the service context (state, application goals, tasks, current activity, resources, etc), and the target platform.
2. need for computing models for the acquisition, modification, correlation, and utilization of user and context information which is dependent on, and is controlled by the application functionality. Information sources to feed user and context models are i) the user (e.g. preferences, tasks, history, failure, information exchanged or selected, etc); ii) the system (e.g. user goals, repeated actions, sessions, etc), and iii) the computing and communication environment (user terminal, bandwidth, computing platform, user location etc).

Interaction modelling. To achieve flexibility and smart interaction the system needs additional knowledge to manage the communication patterns among the user and the system in order to adapt them to the domain semantics, to the user model and to the context. Interaction models aims at representing the behaviour of the user interface. However there is a wide variety of formalisms and interpreters ranging from finite state automata, and graphs to logic formalisms and sophisticated interface cognitive agents. Research challenges include categorising existing UI interaction models and tools, according to different criteria like domain, semantics, formalisms, platforms, rendering engine, etc. and to operationalise them in order to be assembled and composed with other application components. Those issues will be addressed in NEXOF in order to provide platform neutral components and tools which could be orchestrated into UI service components, which in turn could be easily assembled with other service components.

¹⁸ www.w3.org/RDF/

¹⁹ www.w3.org/2004/OWL/

²⁰ www.w3.org/Submission/SWRL/

Engineering view

The Web 2.0 phenomena [Oreilly05] introduced new models of interaction with web contents, and with social communities where users may play real or fictitious roles. This movement relies on a user-centric perspective where users are fully empowered to select resources of their interest, annotate, and configure their own services and application suites, sharing these results and their knowledge and resources with other users. Typical examples are Mash-up platforms e.g. Google front-site (gadgets can be assembled according to the users' hearts desires), Netvibe²¹ (an RSS-aggregator), Yahoo pipes²² (arbitrary and rich design of content-filters, based on the RSS protocol), Kapow Technologies²³ (one of the first providers to head for so-called Enterprise Mash-ups). Shortcomings of existing mash-ups are due to their restricted functionality, as they only focus on the combination of simple content rather than application functionality. Standards facilitating publication and self-management of visible resources are also needed.

Web communities aim at unifying their users by means of a common ideal such as social networking (e.g. LinkedIn Swicki for knowledge sharing, or GiveMeaning for participation in a special interest group, just to mention a few). Users mainly benefit from having access to social networks and from finding important information in a very efficient way.

Platforms or tools for information and application sharing like Web logs²⁴ and online directories that support navigation across a broad audience of individuals and enable to identify people who have similar interests.

The aim of NESSI technology foundations is to go beyond existing Web 2.0 technology, and tagging mechanisms, to provide new functionality to a) customize the operational environment by selecting, adapting, and composing the user interface of their services; b) produce semantic based annotations of selected entities and services which may be used to improve discovery, composition of content and services, and human-system interaction; c) to facilitate secure social interaction by providing, visual mechanisms to manage user communities, user identity, privacy and security constraints, and the information to be shared, annotated, or send to specific groups and individuals d) platform independent and smart user-system interaction.

Service-oriented UI-components. Encapsulation and abstraction mechanisms that help to handle and hide the complexity of the UI are one of the main engineering requirements to efficiently build service centric applications. User interface technology provide a wide range of solutions for modelling and implementing basic aspects of the UI such as information presentation (e.g. textual, graphical, voice, multi-media), low level information acquisition and modelling (e.g. sensors, terminals, single and multi-channel), and user- service and user-application interaction modelling (e.g. UI interaction languages, UML interaction diagrams, dialogue grammars, interaction patterns). Integrating all these technologies and UI related artefacts into service-centric components offering homogenous descriptions, still an open issue which will be addressed by providing a holistic, comprehensive and consistent view of the user interface made up of a baseline of concepts formalisms reference architectures, implementations, and supporting tools for building advanced User Interfaces.

2.5.1.1 State of the Art overview of Context Modelling

The following paragraphs provide a brief overview and classification of existing approaches to the modeling of context. It builds, among others, on the finding of a comprehensive study published in [Strang04].

²¹ www.netvibes.com/

²² pipes.yahoo.com/

²³ www.kapowtech.com

²⁴ www.weblogs.com/

Key-Value Models: In [SAW94], a simple, but major approach to systematically capturing contextual information has been presented. Key-value pairs have been introduced “to model the context by providing the value of a context information (e.g. location information) to an application as an environment variable”. [Strang04] The pairs of certain keys (which can be referred to as context parameters) and related values (i.e. context values) can then be used to systematically influence and adapt the state and behavior of an application.

Markup Scheme Models represent the second major class of context modeling approaches: They rely on a hierarchical data structure consisting of markup tags with specific attributes and contents. Representatives of this context modeling approach are frequently referred to as *profiles* [Strang04] which are based on a “serialization of a derivative of the *Standard Generic Markup Language* (SGML), the superclass of all markup languages such as the popular XML.” [Strang04] Held [HBS02] has proposed *Comprehensive Structured Context Profiles* (CSCP) which do not define any fixed hierarchy for the representation of contextual information. They rather support “the full flexibility of RDF/S to express natural structures of profile information as required for contextual” [Strang04] data. The *CC/PP Context Extension* [IRRH03] (introducing a number of component-attribute trees related to specific aspects of context) as well as the *Pervasive Profile Description Language* (PPDL) [Chtcherbina03] represent further representatives of the markup scheme-based approach to modeling context.

Graphical Models: The *Unified Modeling Language* (UML) can be leveraged to use context data. Henriksen et al. [HIR03] have extended the graphical *Object-Role Modeling* (ORM) approach [Halpin01] with respect to certain contextual classification and description properties [HIR02]. Contextual parameters such as the specific device used for accessing a software application, the user’s location, his current activity, and others may thus be modeled and visualized coherently and in a determined relation to each other.

Object-Oriented Models: The goal of using object-oriented methods for the modeling of contextual data is “the intention to employ the main benefits of any object-oriented approach—namely encapsulation and reusability—to cover parts of the problems arising from the dynamics of the context in ubiquitous environments. The details of context processing is encapsulated on an object level and hence hidden to other components. Access to contextual information is provided through specified interfaces only.” [Strang04] *CUES* [SBG99] and the *Active Object Model* [CMD99] are representatives of the object-oriented context modeling approach.

Logic-Based Models: “A logic defines the conditions on which a concluding expression or fact may be derived (a process known as reasoning or inferencing) from a set of other expressions or facts. To describe these conditions in a set of rules a formal system is applied. In a logic based context model, the context is consequently defined as facts, expressions and rules. Usually contextual information is added to, updated in and deleted from a logic based system in terms of facts or inferred from the rules in the system respectively. Common to all *logic based models* is a *high degree of formality*.” [Strang04]. *The Stanford-based Formalizing Context* [McCarthy93], Giunchiglia’s approach [Giunchiglia97] which is frequently referred to as *Multicontext Systems*, as well as the *Extended Situation Theory* [Akman97] constitute some of the most important representatives of the logic-based class of context modeling approaches.

Ontology-based Models: Ontologies are usually referred to as data models that represent sets of concepts within specific domains as well as the relationships between those concepts. They can be leveraged to “reason” about the objects within a domain. In [Krummenacher07], an analysis of the *requirements for the use of ontologies* for context-modeling is provided: Benefits and challenges of ontology-based models are identified and a comparative evaluation is conducted to further improve the usability of ontologies in context-aware applications. In [SLF03a], a basic context modeling approach is introduced which uses ontologies as a formal fundament: The so called *Aspect-Scale-Context* (ASC) model is presented and compared with different, related models. The *Context Ontology Language* (CoOL) is derived from this model, which may be used to enable context-awareness and contextual interoperability during service discovery and execution in a proposed distributed system architecture. In [SLF03b], the real-world applicability of CoOL is shown on the basis of numerous case studies. In [Schmidt06], the authors stress the importance of *robustness and scalability of ontology-based context modeling* approaches as the key enablers for all different kinds of emerging context- and situation-aware applications. Novel ways for addressing imperfection of high-level context information, its time-dependence and other factors deteriorating scalability and robustness of existing approaches are proposed. Finally, in [KLS 06], methods for the modeling of context rules particularly in the field of Web Services are presented. Modeling context information based on formal descriptions is thereby regarded a core aspect of service integration and interoperability. An improved and simplified version of the *Context Ontology*

Language modeled in *WSML-Rule* is proposed to demonstrate the potential use of context rules in pervasive computing applications and in particular as part of Semantic Web service descriptions.

2.5.2 The Semantic Web

Besides the design philosophy of Service-Oriented Architectures, the **Semantic Web**²⁵ emerges as major building block.

The Semantic Web aims at augmenting the existing World Wide Web by **machine-readable semantics, making the content of today’s Web accessible to intelligent queries and machine-reasoning**. Additional information attached to resources will formally describe content, meaning and relations between processes in order to enable the evaluation and process this information by machines. The future goes towards a Web improvement allowing a greater interoperability of information systems and reducing the mediation of human operators involved in information management.

By formalizing the meaning of Web-based resources, the **process of retrieving and dealing with the desired information or functionality** can be significantly accelerated and improved. So far, resources are designed for consumption by human beings who may request and read content or consume application functionality offered by a certain Website. In this context, the procedure of finding relevant resources is supported by search engines such as Google²⁶. However, a computer is not enabled to accomplish the same tasks without human control as Web sites are designed to be read by humans (in an informal way), not machines. The vision of the Semantic Web relies on additional, formalized pieces of information attached to all resources to make them interpretable for machines, so that they can perform more of the efforts involved in finding, sharing and combining information on the Web.

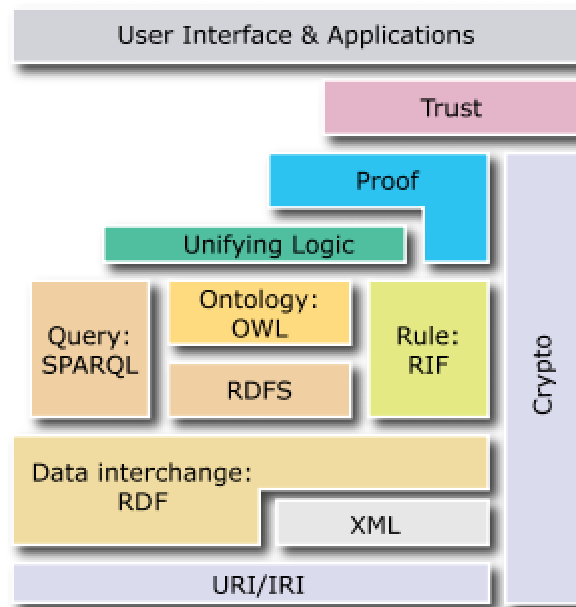


Figure XXX: W3C Semantic Web Layer Cake²⁷

²⁵ <http://www.w3.org/DesignIssues/Semantic.html>

²⁶ <http://www.google.com/>

²⁷ <http://www.w3.org/>

This kind of **semantic annotation** is based on representation meta-languages like XML, XML Schema, RDF²⁸, RDF Schema and OWL. From XML documents, a layer of annotations is created using the RDF-semantic model. RDF Schema is a vocabulary for describing properties and classes of RDF-based resources, including semantics for generalization-hierarchies of such properties and classes. This data with some semantic are given meaning through ontologies, which are common conceptualizations created by domain experts, and usually described using OWL²⁹ (which adds relevant vocabulary for describing properties and classes: among others, relations between classes (e.g. disjointness), cardinality (e.g. “exactly one”), equality etc.). Since concepts are formally expressed, some data sources can be automatically processed and combined. SPARQL³⁰ finally represents a protocol and query language for Semantic Web data sources.

Semantic Web Services (SWS) enrich Web Service technology by semantic annotations, providing the framework for automated discovery and composition of program functionality. European research has yielded the leading conceptual SWS framework WSMO³¹, a family of respective representation languages (WSML³²), and the prototype implementation of a WSMO-compliant execution environment WSMX.

All efforts in the Semantic Web community have so far focused on formalizing Web-based resources to simplify their retrieval, reasoning and handling based on machines. The **wide acceptance of certain standard ontologies** and the **integration of human users** represent the two major challenges for the future in this field. “Semantic islands” are only useful to a limited degree. The inclusion of human users can be achieved by interconnecting user-driven, lightweight ontologies (often referred to as **folksonomies**) with more formal ones (e.g., OWL) on the one hand and by making machine-based reasoning easily and intuitively accessible to human users.

2.5.3 Contextual Computing

Context can be defined as follows: “The context of an idea or event is the general situation that relates to it and which helps it to be understood” [Hu 2006]. Systems theory [Bertalanffy1968] deals with context as it aims at identifying the boundary between systems and their respective environmental context. A widely accepted approach is to assume the boundary delimiting that part for which we seek a detailed understanding in terms of the constituent components, their behaviours and relationships. What is inside the boundary is “the system”. The “context” or “environment” is everything that is outside the boundary, but that may interact with the system and influence its behaviour.

Contextual computing aims at developing systems incorporating awareness of and the ability to adapt to contextual information about users and their respective environments with the goal to enhance application services. Currently, technical applications frequently behave the same way regardless of who and where the user is or what he is actually doing. Summing up, contextual computing is about extending systems with the ability to detect contextual information, to represent it and display it to the user, to manipulate it, and to influence the behaviour of applications with the goal to support human users in a proper and personalized fashion.

Numerous different promising **use-cases** of contextual computing have emerged and continue to emerge in various fields such as e-Business, e-Learning, Healthcare, Mobile Communication Networks and many more. AmbieSense³³

²⁸ <http://www.w3.org/RDF/>

²⁹ <http://www.w3.org/TR/owl-features/>

³⁰ <http://www.w3.org/TR/rdf-sparql-query/>

³¹ <http://www.w3.org/Submission/WSMO/>

³² <http://www.wsmo.org/wsml/>

³³ <http://www.ambiesense.com/>

represents one example for the extensive utilization of contextual information for the adaptation of technical systems. The goal of AmbieSense is to provide personalized, context-sensitive information to mobile, “nomadic” user. An ambient information environment will be provided through a combination of context tag technology, a novel software platform to manage and deliver the relevant information and personal computing devices to which the information is served. The “MILK³⁴” project has been funded by the EU and developed context-aware applications for information sharing in distributed organizations. Another project dealing with related issues has been conducted by XEROX and is called WID³⁵ (Web Information Discovery). It aims at automating the “Hidden Web” discovery process and is divided into three subprojects: the discovery, the analysis, and the classification of resources.

In general, applications which may **benefit** from leveraging contextual data include: The classification and ranking of documents, presentation of data to users, information generation and filtering as well as adaptation to diverse media.

Challenges inherent to the deployment of systems which realize the goals described above comprise the following: There is no homogeneous, interoperable description of contextual information available that is accepted across a significant amount of systems. An open question is also the physical location of information that relates to permanent user context such as gender, age, preferences, etc. Taking into account the numerous channels for accessing Web-based resources, such kind of contextual information must be centrally available. Apart from that, there is no comprehensive approach available for the context-sensitive rendering (representation) of resources depending on the device used for displaying.

2.5.4 Web 2.0

Numerous various **definitions** of Web 2.0 exist that mostly only describe certain aspects of the overall concept. Tim O’Reilly, who originally coined the term, identifies seven major characteristics inherent to the Web 2.0 concept [O’Reilly05]:

- First, the Web is considered as a platform for building systems that do not necessarily have a specific owner and are “tied together by a set of protocols, open standards and agreements for cooperation”.
- Harnessing collective intelligence of Web users represents the second major paradigm.
- The ownership of mission-critical data is regarded a further cornerstone of numerous Web 2.0 applications.
- Fourth, O’Reilly propagates the end of the software release cycle as another central paradigm.
- The use of lightweight programming models that allow for loosely coupled systems and applications, the provision of software above the level of a single device and the realization of rich user experience represent the last major paradigms inherent to the concept of Web 2.0.

Besides such analyses that properly describe parts of the super-ordinate concept, there are only very few comprehensive scientific definitions available. An in-depth investigation [Hoegg06] of 40 different successful Web 2.0 applications condensed the respective characteristics in the following statement which works as underlying definition for this work: “*Web 2.0 is defined as the philosophy of mutually maximizing collective intelligence and added value for each participant by formalized and dynamic information sharing and creation*”.

Users typically consume Web 2.0 applications in different forms [62]: first, **communities** aim at unifying their users by means of a common ideal such as social networking (e.g. LinkedIn³⁶), knowledge sharing (e.g. Swicki³⁷) or

³⁴ <http://klee.cootech.disco.unimib.it/milk/>

³⁵ <http://www.xrce.xerox.com/competencies/contextual-computing/projects/wid/home.html>

³⁶ <http://www.linkedin.com>

participation in a special interest group (e.g., GiveMeaning³⁸). Users mainly benefit from having access to social networks and from finding important information in a very efficient way. In general, Web 2.0 is known as the transition occurred in the Internet from traditional Web sites to user-centric web applications. This kind of software is expected to replace desktop applications in many uses, conceiving the Web in an opposite way to Web 1.0. Traditional Web was centred in data, its presentation and its distribution among servers. New Web 2.0 approach is centred in end-users, understanding the web as a huge net of users, considering every of them like computational and knowledge-management interconnected atoms in a kind of social web or collaborative web, therefore global collective intelligence can be exploited. Web 2.0 promotes this exploitation in several ways. Web use would be oriented to users' interactions and their social nets, outsourcing global results and efforts. That is, Web 2.0 sites are authentically meeting user-oriented points, against the traditional raising. An important pragmatic example of this collaborative social Web is the appearance of *folksonomies*, resulting from a social tagging technique of web pages, resources, services, etc. Any user can associate several tags (particular words) to an URI, creating a tagging space with two levels (URI tagged and tags utilized to tag it) and arity N.

With this technique it is intended to provide (informal) semantics to every tagged element, through the meaning associated by user tags. Using user tags in this way, it is intended to exploit this semantic donation allowing direct searches with fixed elements semantic orientation, for example. It is important to emphasize that this semantic notation is fully user-oriented, in a free and flexible form. User-provided meaning is given to each desired URI using tags so, are the users by their own who assign meaning to all resources they found interesting.

Second, **online collaboration tools** aim at supporting users in collaboratively executing certain processes such as maintaining time schedules, performing online brainstorming or processing text online.

Third and last, certain **platforms or tools** empower users to create and share mere information or even application functionality with a broad audience and to identify people who have similar interests. Web logs and online directories that support navigation are cited as exemplary applications of this category.

As argued in [Kolbitsch06] and [McAfee07], Web 2.0 does not stride along with a fundamental **technological innovation**. However, current maturity and spread acceptance of this technologies have promoted a real use innovation. In this way, certain Web programming languages, technologies and architectural styles have facilitated the fast Web 2.0 proliferation [Hoegg06].

First of all, the Representational State Transfer (**REST**) [Fielding00] is an architectural style that has been proposed by Fielding in his doctoral dissertation. With the help of REST, Web clients can interact with arbitrary Web resources (sources of specific information) in a uniform way and thereby exchange XML based messages. Resources are uniquely addressable by means of a global identifier (URI) and support a well defined and also constrained set of supported operations (so-called HTTP verbs such as GET, PUT). By using resource identifiers and HTTP verbs together, clients can invoke REST-based services and request the delivery of, for example, an XML file containing the desired specific information. The Simple Storage Service (S3³⁹) provided by Amazon is one of many exemplary applications of REST technology: On the basis of a simple service interface, users are enabled to store any amount of data on Amazon servers and also retrieve it from arbitrary locations on the Web. REST incorporates the principles of complexity-hiding and uniformity since it significantly reduces programming efforts and forces both clients and recourses to adhere to a common set of supported operations and interface descriptions.

³⁷ <http://swicki.eurekster.com/>

³⁸ <http://www.givemeaning.com/>

³⁹ <http://www.amazon.com/gp/browse.html?node=16427261>

The Really Simple Syndication (**RSS**⁴⁰) format represents a further technology that is affiliated with numerous Web 2.0 applications. An XML based file format, RSS supports the easy aggregation of content from arbitrary sources in the Web. With the help of this standard, the information offered by frequently changing Web sites such as Web logs can be exported into XML files of a uniform structure and is thus made publicly available. The files' structures are determined by the RSS specification which only allows for a very limited set of content categories such as title, Web link and textual description of a piece of information. Web applications that aim at distributing content via RSS technology must ensure that the respective data files always contain the latest information. Applications that present the information obtained from different RSS-based sources in a combined manner are referred to as content aggregators. Users are thus enabled to gather information that has been created by a number of authors and is easily accessible via uniform interfaces, redesign it and also publish it.

AJAX represents a composite of several other technologies that together are devoted to allowing for rich user experience which is one of the key paradigms of Web 2.0 applications. As described in [Malone01], AJAX comprises the use of XHTML, CSS, Document Object Model, XML and XSLT, Javascript and others to enable Web interface applications that are as impressive as usually only local applications are. One key to this user interface innovation is the reduction of data transferred between servers and clients to save bandwidth and thereby avoid interruptions of applications due to limited network resources: in case a Web page must be updated, only the modifications are loaded from the server instead of the complete page.

Folksonomies (e.g., del.icio.us⁴) are an example of social mechanism for improving the share and acquisition of user knowledge by means of a collaborative tagging technique of web pages, resources, services, etc. Any user can associate several tags (particular words) to an URI, creating a tagging space with two levels (URI tagged and tags utilized to tag it).

With this technique it is intended to provide (informal) semantics to every tagged element, through the meaning associated by user tags. Using user tags in this way, it is intended to exploit this semantic donation allowing direct searches with fixed elements semantic orientation, for example. It is important to emphasize that this semantic notation is fully user-oriented, in a free and flexible form. User-provided meaning is given to each desired URI using tags so, are the users by their own who assign meaning to all resources they found interesting.

One major characteristic inherent to all Web 2.0 applications is the focus on the “**long tail**” as introduced by Chris Anderson in [x4]. Instead of heading for the small number of huge customers, Web 2.0 applications aim at involving the bulk of private users or small businesses. To really allow for “customer self-service” as promoted by O’Reilly in [61], the above presented light-weight programming models are crucial.

Besides social communities (e.g., LinkedIn.com, GiveMeaning.com) or online collaboration tools, **Mash-ups** are considered as central to the Web 2.0 era. *The Economist* defines Mash-ups as follows [TheEconomist05]: “The term Mash-up is borrowed from the world of music, where it refers to the unauthorised combination of the vocal from one song with the musical backing of another, usually from a completely different genre. Web Mash-ups do the same sort of thing, combining websites to produce useful hybrid sites and illustrating the internet's underlying philosophy: that open standards allow and promote unexpected forms of innovation.” *The BusinessWeek* [Hof07] explains this novel form of using Web-based resources similarly: “Mash-ups portend big changes for software companies, Web sites, and everyone online. No longer just a collection of pages, the Web is morphing into a sort of global operating system, à la Microsoft [...] Windows. And now, people are learning to program Web 2.0 with much of the same innovative energy of the personal computer's early days”.

We basically adhere to these explanations and take the following summary as foundation for this work: A Mash-up is a Web-based resource, be it content or application functionality, which has been created through reuse and composition of two or more different resources.

⁴⁰ <http://www.rssboard.org/rss-specification>

Mash-ups can be setup by using resources such as the OpenStreetMap⁴¹ service, which provides free geographical data such as street maps. By leveraging lightweight protocols and a simple interface, it can be seamlessly integrated into other Web-based applications and thus be enriched with additional, arbitrary features such as pictures of local buildings or tourist attractions. Platforms and tools are provided in the Web so that to empower users to loosely couple readily available applications, to enrich and compose them into a novel service, which can again be made publicly available. As a result, meshes of different Web-based resources emerge that can be considered rather un-governed since there is no central control entity existing that imposes formal guidelines for reuse and syndication. The possibility to actively contribute and gain reputation by publishing resources on the one side and to harness the aggregated knowledge and intelligence of all other platform participants on the other side represent core benefits for users.

The Website ProgrammableWeb.com acts as a major aggregator of numerous (currently more than 1600) references to diverse existing Mash-ups all over the world. It provides statistics about used resources, a classification of Mash-ups into categories as well as user statistics to evaluate their popularity. Therefore it serves as one central source of empirical data used in this work.

To investigate the most frequently used resources and their respective Application Programming Interfaces (APIs), the Web Site exposes an “API-Scorecard” (Figure 2) which has six major API-providers on the X-axis and the most relevant categories of APIs offered on the Y-axis. In this context, the term API refers to the interface type, data formats and the protocol to be used for interconnecting to an application. The last line of the figure lists the number of APIs referenced on the Website and the number of Mash-ups using one of these APIs for each of the 6 major providers. For example, 20 Google-based APIs and 1046 Mash-ups that leverage one of these APIs are listed on ProgrammableWeb.Com.

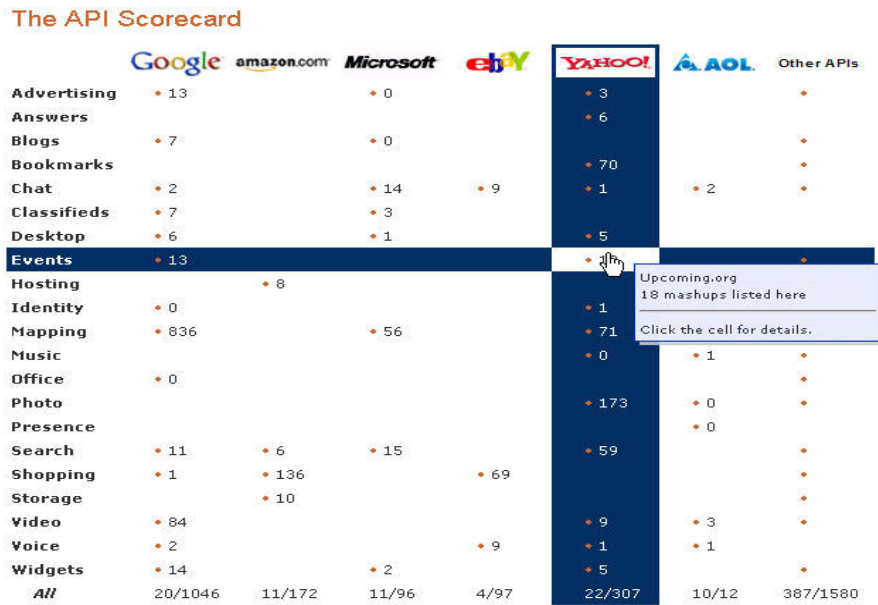


Figure 3: Mash-up API Scorecard

Several facts can be inferred from the above graphic: First, large companies such as Google, Amazon and Microsoft start to successfully provide Web-based resources which are leveraged by numerous users for creating Mash-ups on their basis. As can be seen in Figure 2, at least 836 Web-based applications have already integrated the “Google

⁴¹ <http://wiki.openstreetmap.org/index.php/REST>

Maps”- resource that offers geographical data. The Website “Vuelta a California” can be cited⁴² as an excellent example for a Google Maps-based Mash-up. It uses information from a Web log that deals with road profiles for bikers in California and mashes it with road map data from Google Maps to achieve a rich visual representation.

The second important fact that can be inferred from the API scorecard is the shift away from the professional corporate context towards a private, end-user driven field of applications. As opposed to the focus of traditional Web Services, the resources used for building Mash-Ups aim at the long tail of Internet users and deal with media management, shopping functionality, entertainment or desktop applications. Heavy-weight enterprise applications for automating business transactions can only rarely be found in this context.

Major existing **Mash-up platforms** that allow users to create Mash-ups as shown above include the Google⁴³ front-site (gadgets can be assembled according to the users’ hearts desires), Netvibes⁴⁴ (an RSS-aggregator), Yahoo pipes⁴⁵ (arbitrary and rich design of content-filters, based on the RSS protocol), Kapow Technologies⁴⁶ (one of the first providers to head for so-called Enterprise Mash-ups)

Shortcomings of current Web 2.0 technologies and design principles include:

- Existing Mash-ups mostly focus on the combination of simple content rather than application functionality (mere “syndication” rather than stateful “coordination”)
- Dispersed platforms exist that follow different standards and request resource providers to adhere to divergent templates for publishing their content or functionality. A consistent approach (“resource template”) is required.
- The majority of providers of Web 2.0-related applications have not been able to realize profitable business models. Novel and sustainable models are required for the new era of an “Internet of Services”.
- Typical Web 2.0 applications are rather lightweight and simple. We aim at bridging the gap between more advanced, partially even enterprise class applications and intuitive user interfaces (“gadgets”) that allow for the integration of users.
- Most Web 2.0 applications are not subject to governance mechanisms and do not have to fulfil strict requirements with respect to trustworthiness, security or performance. This may lead to the users’ dissatisfaction and requires new approaches.

2.6 Engineering Approaches

2.6.1 MDE

Model-Driven Engineering (MDE) denotes software development processes that are model-centric as opposed to code-centric. The motivation of MDE is to develop complex and large software based systems and services that are easier to maintain and extend. This relies on the fact that models are easier to understand for humans than code since they are visual and provide higher levels of abstraction. Furthermore, several time consuming development tasks can be done semi-automatically or automatically for example by providing tools for model transformation and code generation; thus, reducing manual work which is error-prone. Also, by performing model simulations and model

⁴² <http://www.berndstraehle.com/map.html>

⁴³ <http://www.google.com>

⁴⁴ <http://www.netvibes.com/>

⁴⁵ <http://pipes.yahoo.com/>

⁴⁶ <http://www.kapowtech.com/>

analyses with respect to different concerns the models can be used for validation and detection of errors at an early stage of the development phase.

In MDE Models are considered as first class entities. Transformations are the primary operation on models, used to map information from one model to another. MDE represents an important paradigm shift in the field of software engineering that may have important consequences on the way information systems are built and maintained. Presenting their Software Factory approach, J. Greenfield and K. Short write in [Greenfield and Short, 2004]: "*The software industry remains reliant on the craftsmanship of skilled individuals engaged in labor intensive manual tasks. However, growing pressure to reduce cost and time to market and to improve software quality may catalyze a transition to more automated methods. We look at how the software industry may be industrialized, and we describe technologies that might be used to support this vision. We suggest that the current software development paradigm, based on object orientation, may have reached the point of exhaustion, and we propose a model for its successor.*"

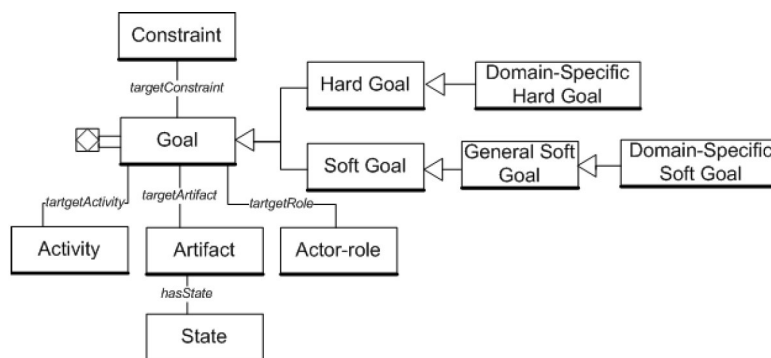
Currently there is also an increasing interest in the research communities of runtime utilization of the models, where the models are the meta representation of the running services. For instance in the areas of service discovery and composition and dynamic adaptation, where the services are automatically managed during execution by manipulating their models. This can be achieved by using meta-modelling languages which allow the definition of the semantics of meta-models and by providing platforms for model execution.

2.6.2 Goal oriented software development

The motivation of goal oriented software development is to *Build the right system or service by identifying and supporting stakeholders goals*

Related work includes:

- Intentional SOA (ISOA); which include Intention-oriented Service description, discovery and composition. In ISOA intention is viewed in the same sense as goals. Intentions express what is wanted, i.e., a state that is expected to be reached or maintained” [Collette Rolland, Rim-Samia Kaabi, and Naofel Kraiem “On ISOA: Intentional Service Oriented Architecture CAiSE 2007]. A current ISOA standardisation effort is the OMG UPMS-HA (UML Profile and Metamodel for Services for Heterogeneous Architectures)
- The Beliefs - Desires - Intentions (BDI) model (e.g., in Agent based Software Development)
- Goal oriented Business Modelling; e.g., the OMG Business Motivation Model
- Goal oriented languages and frameworks for requirements engineering such as the i* Framework and the Goal-oriented Requirements language (GRL) [University of Toronto, TROPOS project].
- Goal oriented ontologies (an example is given in figure below [Goal Annotation of Process Models for Semantic Enrichment of Process Knowledge Yun Lin and Arne Sølvberg CAiSE 2007]. In goal oriented ontologies common concepts are *hard* and *soft goals* which represent functional and non functional goals respectively.



2.7 Limits of today's systems (some examples)

[Objective: stress limits of today's systems in terms of functionalities demanded versus offered. Exemplify through some of the most emerging, pressing demands – obviously from technological perspective -]

[Missing input on this topic, to be developed]

3 The Open Service Platform

3.1 Overall view

The following architecture views have been produced in December 2006 for the purpose of confirming gaps to address in priority. This model has not been refined; in its current state it is at most a conceptual sketch of what some views of the skeleton architecture could be. It has been included here as a contribution to the collective thought process.

The model is composed of two views. The first depicts in layered way the expected major clusters of functionalities the NESSI Open Service Platform would expose, the latter models main non-functional aspects and as they relate to the agreeing-monitoring-acting paradigm.

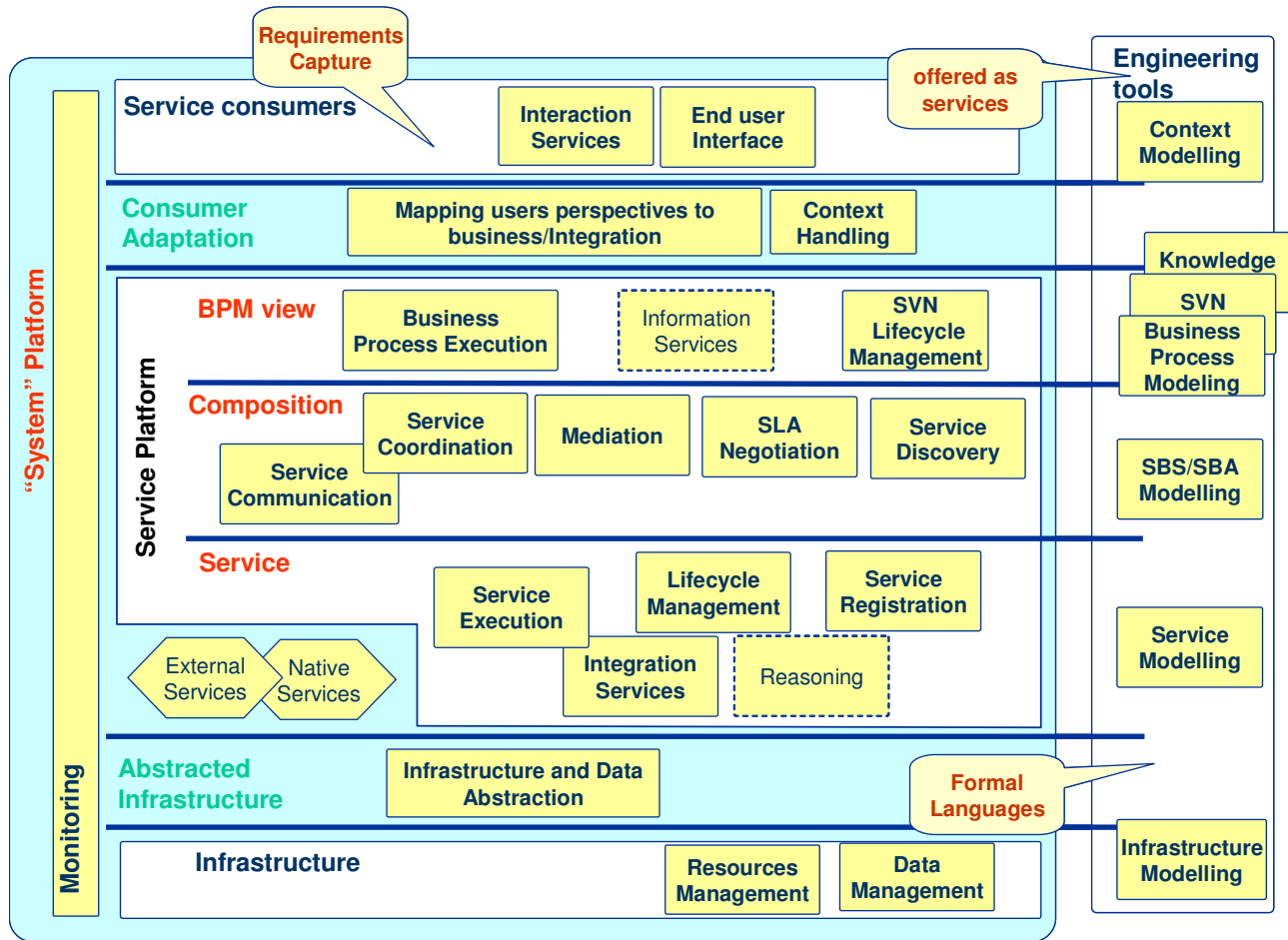


Figure 2. Draft Skeleton Layered Functional Architecture.

“Engineering tools” concerns the research area of needed in order to build (including modelling, development, simulation, etc.) services, service-based systems, etc.

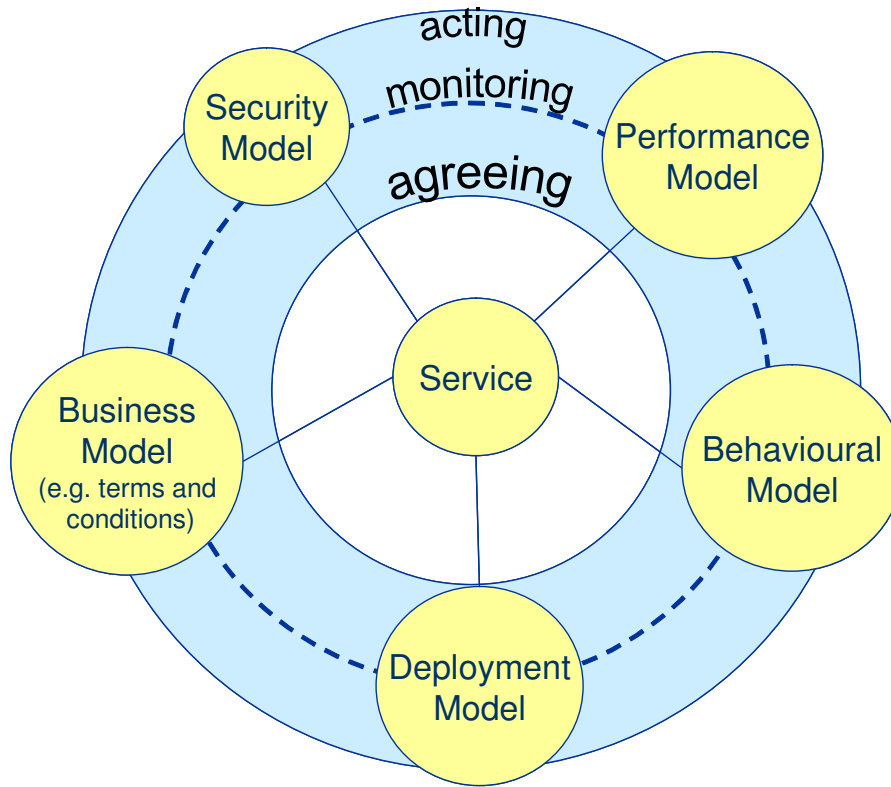


Figure 3. Service non-functional aspects.

Issue:

- Who is responsible for monitoring management? If this is not coupled with the framework (as a single or central engine) this could be “simply” (only) coupled with the service provider, so the question is how to exchange monitoring information and to which extent they exist monitoring standards.

“How to read this section”: the intent of this document is to be as much as possible agnostic with respect to specific approaches. Nevertheless, in the following sections, for some functionalities approach and abstract description are so much intertwined that it is difficult to separate them.

3.2 Description of Functionalities

[Description of each envisioned functionality (each yellow block) at coarse-grained according to some predefined template covering the following aspects:

- *overall description,*
- *intended users,*
- *technological challenge or problem addressed,*
- *positioning with respect to SofA (State of the Art), Gap to fill in,*
- *Scientific and Technological approach proposed (different approaches can be provided, e.g. semantics, security, soa),*
- *expected results (it includes benefits)*
- *...]*

Hereafter we describe the functionalities that we do envision at the level of each coarse grained functional block of the layered view of the Open Service Platform as depicted in and resulting from the work achieved in section 4.1 (clustering of functionalities according to the Conceptual view of the Skeleton Architecture Sketch as promoted/pushed so far and which has been revised in the context of SRA Vol 2. production – see Section 6 for complete update also process attached to this update).

3.2.1 Service consumer layer

Functional blocks attached to this layer are there to answer demands coming from service consumers. Thus they result from an in depth analysis of their (service consumers) specific requirements. They not only cope with existing and/or well known demands but also and probably more importantly to first acquire and second accommodate new service demands which may partly require to evolve existing services and partly to even come up with completely new breed of services. Last but not least each of those functionalities will be assessed against a number of key criteria such as usability, versatility, openness and user acceptance.

Two major functional blocks have been identified here:

- User-service Interaction
- End-user Interface

3.2.1.1 User-Service Interaction

The growing relevance of services in today's economy increases the importance of highly effective and efficient user/service interaction leading to more productivity. Productivity can be increased particularly along with the following three phases of interaction.

Retrieval (moving from data retrieval to service retrieval, Service Access, accessibility): In the Internet of today, seeking specific services can be a time-consuming task. With a significantly growing number of services in the Web, this complexity as well as the uncertainty with respect to exact service features and quality will even deteriorate the productivity of finding adequate services which exactly meet individual preferences and needs.

ServiceUsage and Intelligibility: In the Internet of today, the utilization of electronic services involves difficulties when to **cross-service interoperability** and **cross-device usability** are required: Mostly, users intend to utilize a set of services and desire to arrange and compose them in a way that solves their unique problems. However, current technologies do not support seamless interaction and message exchange between different services. Also, services are hardly usable across different (mobile) devices without interruptions, again leading to decreased productivity of user/services interaction.

(Implicit or Explicit) Intelligent Management: Managing services represents a third hurdle to efficient and effective user/service interaction. Services are rarely subject to formalized and standardized performance parameters which allow for manageability and thus predictability and dependability of the interaction. User will less and less in the near future directly deal with the technical implementation part of the services needed to answer their demand or request. Therefore User Service interaction would have to be mediated.

With the help of the following **means**, the above discussed hurdles can be tackled adequately:

First, *intermediaries* are required which take over institutional functionality. In a Web of countless different services, such entities are needed to ensure market transparency, navigation capabilities and a certain degree of governance. By gathering feedback (e.g., with respect to performance) about services, by building up a comprehensive services registry which simplifies service retrieval and by monitoring and enforcing previously agreed conditions of services utilization and bindings, uncertainty inherent to users/service interaction can be reduced significantly.

The utilization of *context* is crucial as well. Both user context (knowledge, profile, preferences, idiom, association to social networks) and usage context (static and dynamic characteristics of devices used for accessing services, geographical location and time, connection bandwidth) shall be leveraged to improve service retrieval and

utilization. By considering all major context parameters, an appropriate service can be found faster than without contextual search. Also, depending on user and usage context, the utilization phase of user/service interaction can be enhanced: Depending on bandwidth and used device, for example, the visualization should be adapted.

Usage of context should promote goal-oriented approaches. Referring to contextual knowledge about the user, environment should also empower end-users to clearly perceive how knowledge is put at their service at any time (captured from or provided by him, captured from context, shared with other users, during content and service discovery, service access, while tagging, etc.) enabling them to reach the highest degree of productivity during interaction.

Third, besides of pulling information from the users to services (e.g., for contextual search), services shall also be *pushed* towards users proactively. Indeed, unsolicited pushing of services according to automated inference of current user needs may represent a paradigm shift and a major advancement in terms of user/service interaction.

To achieve a more productive level of user/service interaction, Social Software is called to play a key role here in since it enables people to rendezvous, connect or collaborate and as such empowers them. One of the key concepts to consider here in Social Software is that of **folksonomies**⁴⁷ which have shown to be a powerful tool for associating metadata to resources and organizing knowledge in a distributed way. Folksonomies have their strength in their ease of use, which allows the generation of an increasing amount of valuable metadata by non-trained users. The power is inherent to users that massively feed the system with tags, this way avoiding the cost-intensive task of manual annotation by an expert and the inherent difficulty in developing technologies that perform automated annotation of audiovisual content. This way, the low-cost metadata provided through folksonomies can be exploited to enhance search engines and also to improve the navigation experience. Nevertheless relevant new areas of research have to be addressed in order to find out how folksonomies can be structured, linked to services (they are today mostly linked to contents) and combined with more formal technologies like **ontologies** in order to get the best of the two and better serve service access and management from a user-centric (goal-directed) perspective.

To improve the problem of composing different services to create one user interface, mash-up technologies are required. The BusinessWeek⁴⁸ explains this novel form of using Web-based resources: “Mash-ups portend big changes for software companies, Web sites, and everyone online. No longer just a collection of pages, the Web is morphing into a sort of global operating system, à la Microsoft [...] Windows. And now, people are learning to program Web 2.0 with much of the same innovative energy of the personal computer's early days.” With the help of such technologies, users are empowered to create their own interfaces, arrange services according to their unique preferences and also define message exchange patterns between different services (dependencies between services).

3.2.1.2 End-user interfaces

In order to allow for realizing Mash-ups and leveraging the knowledge of the “long tail” through folksonomies, adequate **end-user service interfaces are required**. Service-oriented Architectures are particular tailored and used for business applications within one or across several enterprises. Services provide access to business data (e.g. data of a customer) and allow for managing business workflows. However, services are rarely directly used by end users. User interfaces together with complex control logic must be developed as an additional layer on top of services. This development step is currently only insufficiently supported by methodologies and tools.

Currently, user interfaces are created independent from service interfaces and business processes and thus have to be manually designed. The separation of service design/ implementation and the actual user interface design leads to a gap between the developers’ and the actual users’ understanding of a service. An integrated approach for creating user interfaces for several services is not considered in research and Web Service standards. Even with proper user

⁴⁷ Folksonomies are a collaborative, open-ended labelling system that enables users to categorize different resources such as web pages, on-line photographs, audio or video.

⁴⁸ Mashing the Web, The Economist 376 (8444), Special section, p. 4, 2005

interfaces for each individual service of a complex scenario, an integrated view on the whole functionality would not be possible. In consequence, each service would only be usable via its own user interface, resulting in user interface cuts between neighbored services. The integration of several services to a common user interface is missing.

As an essential part of the NESSI initiative, a “face has to be put on services” so that these can easily be utilized by human users with only little IT literacy. Therefore, novel concepts are in the focus of the work, particularly with respect to service engineering methods, novel techniques for visual, model-driven service and service interface development, methods and solutions for adaptive user interfaces as well as runtime environments which support a close interrelationship between user interfaces and underlying services.

From a consumer adaptation perspective the next generation SOA front-end envisioned here in the context of NESSI Open Service Platform will also offer to end-users the capability of context-aware discovering, selection, tagging, composing (including visual ones) and rendering of resources, following a self-service philosophy. This being mainly due to the fact that end-users can only deal with things they “can see” and “understand” (i.e. speak/has sense to them) at the user interface level (i.e. things that can be visually represented and made “intelligible”). Deeply related to service usage the dynamic (on-the-fly at runtime) mashing of Web services up to the mashing of Web sites/applications can be addressed using, extending and coupling Web 2.0 and semantic techniques (Mash ups and aggregators⁴⁹ are among the research areas for service usage).

Such a front-end platform would operate as an open market-place where content/service providers and end-users get connected. An open market-place where SMEs and large companies may publish their offer, making it visible to end-users. A place where end-users can search/pay for and consume content and applications delivered as services, also a place where they could place specific demands using their (almost) own wording while being assisted in doing so. This is particularly true in the context of on-demand provisioning and delivery of services according to well defined service level agreements where a major challenge at Service customers level remains to first describe and second negotiate required service characteristics in a way that is meaningful to their level of expertise but also sufficiently accurate to be dealt with.

At the end, users will get fully empowered, being able to select resources of their interest, compose and configure using piping/mash-up techniques their own services/business processes and application suites, while validating and assessing their performance beforehand (predictability based on behavioural serviced model encompassing both functional and non functional requirements as expressed) through test suites and visualization tools and at the end sharing these results as well as knowledge with other end-users in order to fine tune and capitalize on it. This personalization process being carried out in a visual manner so as prior IT expertise is no more required.

3.2.1.2.1 Personal Interface for the Information Society

Present interface and desktop implementations not only not implement fully neither the desktop metaphor nor the document orientation: generally they avoid the easy integration of third party components. Present desktop software is not architected to direct Internet/Web interoperation being still needed the intermediation of a web browser which is getting itself into an own desktop software platform. SMEs need an interface to extract full power to the personal memory, to group work, group discussing and consensus reaching, to the creativity and an easy access to enterprise resources and applications. On the other hand SMEs need to reduce the complexity of creation and deployment of whole new types of applications based on extensive use of multimedia, remote services, knowledge management, information inference and mobility.

⁴⁹ These are similar to mash-ups superficially, but are driven by a different goal. The primary goal of aggregating sites is to collect data from heterogeneous and multiple sources and republish the cleansed, integrated and aggregated data at a single point-of-access. These suffer from the well-known information integration problem which may be eased if semantic technologies are used to find content relating to the same instances, enabling mappings between different ontologies created by experts to be re-used by all aggregators.

The goal is an object oriented / service based / semantic powered graphical desktop with full support of identity, personal memory/knowledge, coordination between mobile devices and fixed and remotes ones, webarch based, refactoring of FLOSS desktop software, quality and maturity criteria for desktop applications and support for model MDA languages in order to reduce the risks and costs of development and deploying software and able to work as interface of the «executable enterprise» allowing the creation of powerful interfaces with similar or least costs than the present Rich Internet Applications alternatives.

3.2.1.3 Context Modelling

[Hu06] defines **context** as follows: “The context of an idea or event is the general situation that relates to it and which helps it to be understood”. Systems theory [Bertalanffy68] deals with context as it aims to identify the boundary between systems and their respective environmental context. A widely accepted approach is to assume the boundary delimiting that part for which we seek a detailed understanding in terms of the constituent components, their behaviors and relationships. What is inside the boundary is "the system". The "context" or "environment" is everything that is outside the boundary, but may interact with the system and influence its behavior.

Contextual computing aims at developing systems incorporating awareness of and the ability to adapt to contextual information about users and their respective environments with the goal to enhance application services. Currently, technical applications frequently behave the same way regardless of individual user preferences, situational parameters, applied technologies and others “contextual parameters”. In a nutshell: Contextual computing is about extending systems with the ability to detect contextual information, to represent it and display it to the user, to manipulate it, and to influence the behavior of applications with the goal to support human users in a proper and personalized fashion.

While *contextual computing* research aims to develop systems incorporating awareness of and the ability to adapt to user context [Hu06], *context modeling* comprises the formal representation of contextual information (which can be considered as a basis for contextual computing). Early models mainly addressed the formalization of context with respect to one single application/ application class. Today, uniform context models, representation and query languages as well as reasoning algorithms that facilitate interoperability of applications and context sharing are in the focus of research [Strang04].

3.2.1.3.1 Exemplary Use Cases of Contextual Computing:

Numerous different promising use-cases of contextual computing have emerged and continue to emerge in various fields such as e-Business, e-Learning, Healthcare, Mobile Communication Networks and many more [SHC00, Crowley02, CCRP02, Crowley03]. AmbieSense⁵⁰ represents one example for the extensive utilization of contextual information for the adaptation of technical systems. The goal of AmbieSense is to provide personalized, context-sensitive information to mobile, “nomadic” user. An ambient information environment will be provided through a combination of context tag technology, a novel software platform to manage and deliver the relevant information and personal computing devices to which the information is served. The “MILK⁵¹” project has been funded by the EU and developed context-aware applications for information sharing in distributed organizations. Another project dealing with related issues has been conducted by XEROX and is called WID⁵² (Web Information Discovery). It aims at automating the “Hidden Web” discovery process and is divided into three subprojects: the discovery, the analysis, and the classification of resources. Lieberman at the MIT Media Lab [Liebermann00] analyzed the trade-off between the need for abstraction (to allow for modular and tractable systems) and the need for context-sensitivity. Several applications that demonstrate the power of context sensitivity have been presented at the MIT Media Labs. The use of contextual information has recently also been investigated in the field of communication networks

50 <http://www.ambiesense.com/>

51 <http://klee.cootech.disco.unimib.it/milk/>

52 <http://www.xrce.xerox.com/competencies/contextual-computing/projects/wid/home.html>

[ESKS06, KAESS06]: To improve information routing and dissemination, for example in the case of emerging vehicular ad-hoc networks, the use of manifold contextual information is currently being analyzed.

3.2.2 Consumer Adaptation (functional blocks)

3.2.2.1 Mapping users perspectives to business/Integration

Open question: what do we plan to put here ? no slide available yet ?

As a first attempt ...

We could start here to mention here the systematic grounding of functional requirements vs. non functional requirements from the business level down to the physical infrastructure. A good example could be here the systematic grounding of SLA ...

Service providers require the management of service level agreement (SLA) contracts in business terms, e.g. for minimizing financial penalties for service-level violations or maximizing service-level measurement based customer satisfaction metrics. A considerable numbers of languages (i.e. WSLA, WS-Agreement, WS-Policy) have been developed that are used for specifying SLAs between service providers and consumers. These languages are focused on the specification of functional and quality properties of web services. However, they do not provide support for specifying information that would be necessary in a business context. Early initiatives have been launched for developing an information model to represent business concepts such as SID (Shared Information/Data); see the TeleManagement Forum (TM Forum).

To overcome this limitation one approach could consist in significantly enhancing the information model defined by the TMF in SID by including business SLA definition. Furthermore, the project will enable the service providers to apply to the whole business lifecycle in a consistent way.

3.2.2.2 Context handling

Functionalities introduced hereafter are all related to context management at runtime (context modelling being addressed elsewhere in the document see section 3.2.1.3).

Context-aware computing is traditionally concerned with the impact that contextual information has on the behaviour of mobile devices and software entities. This body of work has taken a somewhat narrow view of context, focusing on simple context parameters such as location, light and noise levels and deployment platform, which can be easily obtained in an automated fashion. A more recent development of this idea is being developed under the Wireless World Initiative,⁵³ (FP6). Context in all these projects is focused on the profiles of users, devices and communication channels that help to adapt applications, user interfaces and to a limited extent the delivery methods (e.g. bandwidth determines resolution). To facilitate the deployment of semantic services on a global scale, “context” will cover a range of aspects ranging from system and location information to social settings and legal regulations.

In the context of NESSI and this section, we give to the term “context” the widest meaning that is possible, comprising both user context (knowledge, profile, preferences, idiom, info about social networks the user belongs to, etc.) and usage context (static and dynamic characteristics of the device used for access, geographical and time location, connection bandwidth, etc.)

Context modelling will serve several purpose when managed at runtime. Among others we envision the following functionalities to be offered:

- Context-based user service interaction: *personalization* of user-service interaction based on information/knowledge we have about the user (his/her profile, preferences, ...). Dynamic management of user context either to personalize/adapt the service interaction and/or learn from it to capture new knowledge and thus enrich/evolve the underlying model(s) (e.g. user model). With regard to the interaction

⁵³ <http://www.wireless-world-initiative.org/>

of users, one important question is to provide secure authentication between users and the front-end platform and also between users and service providers. One additional requirement is to deal with multiple identities of users (e.g. work and private roles). Besides, users tend to be nomadic, i.e. access services in very different environments using different devices. It has to be investigated where and how the context and profile information can be stored, retrieved and managed. Finally, another important issue to deal with, is privacy.

- Context-aware services and processes: use of contextual information (semantically based) to enhance/advance services such as search and discovery services (e.g. context-aware discovery of services to find the “right” service among the billions of services offered) and/or service composition. Context-Awareness to support collaboration across business networks in context of business processes. Methodologies and tools to acquire service context from social networks. SLA-aware service management to include SLA awareness in dynamic service discovery & composition scenarios, both within and across domains which is to date clearly missing. Service Value Network processes aware of SVN policies, contracts and SLAs.

Context management and adaptation will be mainly achieved by including context-dependent factors ranging from immediate concerns of location and language to legal issues and how financial regulations can be managed.

Among others, usage of the following techniques and/or mechanisms, as well as of others which will emerge in the future, will be explored to handle contextual information and properly adapt either service and/or user interaction:

- Context parameters (e.g. the Aspect-Scale-Context model behind CoOL) provided through the means of WSMO goals,
- Context-specific rules and constraints (as enabled by WSML-Rule and RIF) to express policies and business rules,
- Context-specific functionality, such as context adaptation involving service composition, where a core service is composed with a set of services providing context-specific information,
- Methods and tools to extract context from tag clouds and informal folksonomies through means such as rich tags, OpenID, and FOAF.

3.2.2.2.1 Teaching Applications

The adoption of a new application has not trivial costs as part of the ROI formulas. Similarly, the adoption of new practices of enhanced processes implies such costs. When combined, their impact could be, in the worst case, greater. This implies strong barriers to the productivity enhancement specially in knowledge based companies.

We identify the need of a new way to conceive applications/interfacing/knowledge development and deployment simplifying it for converting the book into an application and the application into the reference book of the managed knowledge. Look for technologies for formal structuring and annotation of knowledge, processes and procedures allowing their execution and enough enhancements on interfacing and pedagogical active methodologies for making possible applications which teach themselves or book that executes their teachings creating new levels of auto-study.

The goal is a set of new methodologies and tools to support the learning and acquisition of solutions built on business process modeled knowledge. This knowledge should be used as «a software solution» and as «digital encyclopædia», and managed in the user side with, preferably, native interface integration.

3.2.3 BPM View

Business Process Management (BPM) methods and practices are currently done over intra-enterprise applications, based on costly point-to-point application connections, “silo-driven” business workflows, and ad-hoc process deployment. There is limited process monitoring and limited flexibility in redesign of processes in most businesses

small or large. Further, there is a very little understanding of the BPM lifecycle in open, networked multi-enterprise Services Value Networks (SVN).

The biggest challenge has thus become measuring, analyzing, correcting and managing intra and inter-enterprise portfolios of processes over an open services network.

This calls for:

- adaptive, collaborative *business* processes management over SVNs and value-chains which in turn calls for both
 - next generation of services that need to be configured and composed to enable flexibility in service-value, enterprise processes and business models,
 - tools for dynamic optimization of the SVNs based on KPIs (key performance indicators).
- effective and efficient SVN Lifecycle management meeting the requirements/demands of multi-enterprise networks over a service-oriented ecosystem” (e.g. where multiple enterprises in the network can monitor, control and manage the processes locally and globally).

The targeted end-users for the resulting technologies and associated functionalities being here: Large, Small and/or Medium enterprises and this for a broad range of SVN based ecosystems (e.g. Service Providers for design, Engineering, Business Networks such in Aerospace, Service Delivery Platform in Transport and/or Telecom.)

3.2.3.1 *Adaptive Business Process Execution*

In terms of state of practice, there are many currently available systems that provide support for Business Process Management/Execution. Nevertheless such tools do not provide support for adaptive or autonomic business processes management over SVN. There is thus a need for *autonomic* process management over SVNs, where business processes, both the inter-organizational ones that form the SVN and the internal ones that each partner needs to execute as a result of SVN interactions, need to evolve in a synchronized fashion and in accordance with SVN contracts and with each partner’s business goals.

It is thus important to investigate/develop methods and tools to dynamically “drive and optimize a collaborative business process which is not owned by a partner but connected based on policies through a SVN, and that has to be driven by the objectives of the SVN and of all its members. In the promoted approach decisions of changes to the process, both dynamic/autonomic ones as well as static ones (based on decisions taken by process and SVN modellers) are supported by SVN insight, a set of algorithms and tools to perform analysis and data mining on executed processes, both from the perspective of efficiency and effectiveness of the SVN and from the perspective of each of the partners using the developed technology. The aim is to derive models of which aspects of process executions are correlated to poor value of process KPI or to SLA violations, and through this analysis drive static and dynamic process change decisions. A particularly challenging aspect lies in performing this kind of analysis in an adaptive environment where the subject of the analysis changes frequently in many aspects (SVN contracts and policies, SVN partners, business processes, etc.).

Collaborative business processing in dynamic SVNs brings about inter-organisational scenarios where no common transaction semantic, transaction context representation and coordination protocol can be assumed to exist. It is thus important to design methods and tools to semi-automatically and dynamically acquire, represent and manage collaborative business process spanning multiple organisations (dealing with all relevant aspects such as legal aspects which vary from one country to the other). As important are tools and techniques to (semi-)automatically adapt/change/evolve business processes on behalf of companies in tracking various indicators (e.g. KPIs, market indicators).

3.2.3.2 SVN Lifecycle Management

SVNs are ecosystems of enterprises, small and large, that provide services to one another in multiple ways and models to achieve common set of objectives or shared goals. According to proposed SVNs taxonomy we can quote among others

- Product or Service Networks
- Virtual Business Communities
- Large-scale Value Chains
- Aggregated networked business services on demand

What is at stake here is automated support designed as central to the SVN lifecycle and to the activity of each SVN member, both to propose changes to the SVN driven by new goals or business needs of the member as well as to manage the impact / maximize the benefit of changes initiated by other SVN partners. This to handle SVN processes that are aware of SVN policies, contracts, and SLAs but also provide automated support for deriving models correlating SVN indicators and SLA violations with their root causes and for identifying changes to the defined processes, are requested.

3.2.3.3 Business Process Execution

The cost of adopting new organizational practices, adopting CRM processes, HRM, formal quality control, adoption of information security standards implies high costs increased by acquiring software applications for supporting them. The «executable enterprise» delete the gap between processes and applications, reducing the re-engineering/adoption costs in money, time and overall impact. It offers too a complete snapshot of the enterprise, what is what in the organization and key performance indicators, indispensable tools for productivity enhancement.

Collaborative Process Model ECA CCA EDOC, MDA on SOA, BPDM and others are building bricks to the process executability useful for organizational simulation and, as the next target, the procedure executability.

The goal is an SME enterprise reference architecture; an SME enterprise reference design, thought in terms of the SOOS concept (Standard Organizational Operating System) as a complete and well defined set of enterprise processes and procedures; an «executable enterprise» business engine and a native user interface (see below). A really open FLOSS licensing is a must (see Ken Kretchmer work about openness) in order of eliminate the most of the entry barriers legacy companies could raise.

3.2.3.3.1 On-line decision support

One of the main challenges of the knowledge workers is lack or insufficient knowledge about the work and decisions they need to do/take within the business processes. In particular they need to know and understand:

- the past and the current state of the process:
 - the actions taken so far
 - the data which were processed
 - the people involved
- the future - actions needed
 - the data which have to be processed/generated
 - the governmental rules which support the actions that need to be taken
- the way of processing tasks (from various processes) that may assure that all non-functional requirements will be satisfied

BPM tools aims at supporting the knowledge workers with the above aspects. As it is observed, so far the existing tools are able to provide sufficient information about the processes themselves (i.e. control and data flows) but do not provide sufficient information about the process environment (data, resources) in terms of the domain and the nature of the process (i.e. not a document is an input of this activity, but this is THE claim for provided by THE citizen in THE activity ... registered BY .. and accepted BY ...).

What is more, the mechanisms that are provide seem to be generic but usually, every group of knowledge workers operates on different levels of abstraction (e.g. decision people) and different parts of the processes are the most important for them. Therefore a deep customisation of these mechanisms regarding various groups of workers, their rights and specific expectations in the context of the actions needed is highly required.

Finally, all the mentioned mechanisms are also important for customers of the processes and also need to be customised taking into account the profile of individual customers.

3.2.3.4 BPM Modelling

To adequately support roundtrip management of business processes from business level down to execution and vice versa an appropriate modelling stack to create, maintain and transform process models on and between the different levels is needed.

At the moment lots of different modelling languages (proprietary ones as well as standards) are available for creating process models on the different levels from business level down to the execution level. There are also solutions developed for parts of a roundtrip depending on the context and goal of particular projects. However, no solution or specification is available that covers the complete cycle.

Therefore, appropriate languages have to be selected or defined from the currently existing ones. Results from currently running or already finished research projects in that area will be considered. To make the stack broadly applicable it should be based on open standards. Proprietary alternatives for each of the selected languages should be identified.

An important prerequisite for selecting a language is that a metamodel exists or can be defined for each language in the stack. This facilitates transformation between the process models on the different levels. Transformations have to be defined based on the metamodels in both directions.

The language chain in the modelling stack should be able to fulfil the information needs of as many application scenarios as possible. To create models enriched with additional information results from the NESSI “Semantic Technologies” WG will be considered.

3.2.3.5 Business Process Simulation

In the context of business process (re)engineering, one of the main open issues is the difficulty of adapting and modifying operational dynamics without appropriate tools suitable for analysing and comparing processes in order to understand the benefits (i.e. resources optimization, integration and re-engineering costs, organizational and technological initiatives) generated by changes in process structure and management.

This situation has generated an increasing demand for IT tools supporting not only process development, but also its analysis and (re)design.

Existing tools are based on the analysis of the business process by using discrete simulation techniques. Their main limitation resides in the fact that the emulated environment is limited to rather simplistic models of the process environment (typically of the resource perspective). This means that, for instance, they provide little or no support in the understanding of the relationships between local initiatives and consequences on the whole. Other tools, based on the analysis of the financial model (usually represented using a spreadsheet) do exist, but the simulations that can be performed are not based on the concept of an underlying business process (the behaviour of the system has to be programmed ad hoc using scripting languages).

In this context we propose to improve current solutions by designing simulation techniques based on a multi-model approach: the business process model interacts with other business models (such as the organizational structures one or the financial one) in order to provide innovative tools for the analysis, development and simulation of processes inside complex organizations.

To this end we expect to design methodologies and software tools to perform a multi-model analysis of a business process supporting process (re)engineering at the business level (strategic design, modelling, definition) and at the technological level (implementation, simulation, testing, monitoring) helping the round-trip management.

3.2.3.6 Knowledge Modelling

[Missing input on this topic, to be developed]

3.2.3.7 SVN Modelling

We are currently witnessing the emergence of diverse interoperable collaborative business processes and service systems within firms, across firms, and across entire value networks composed of multiple firms.

The complexities of collaborative processes, interoperation, monitoring and measurements are enormous for enterprise networks and challenging.

In the meantime this also calls for advancing our understanding of Service Value Networks (defined as ecosystems of enterprises, small and large, that provide services to one another in multiple ways and models to achieve common set of objectives or shared goals) as well as the development of a set of effective methodologies, best practices, standard concepts and semantics, and tools for their identification/acquisition and description/modelling. This especially to allow afterwards reasoning about their properties, with particular reference to adaptive processes, interoperability and multi-enterprise service networks.

As part of such a design we also recommend to include an automated support as central to the SVN lifecycle and to the activity of each SVN member, both to propose changes to the SVN driven by new goals or business needs of the member as well as to manage the impact / maximize the benefit of changes initiated by other SVN partners.

As such SVN modelling calls for research in at least the following areas:

- Systematic and formal modelling approach to interoperable service value networks (SVNs)
- Adaptive, collaborative processes over service value networks and value-chains
- Investigate, study and model adaptive business processes in Services networks composed of multiple enterprises (small, large and medium).”
- Design methods and tools to acquire, model and manage collaborative business process spanning multiple organisations (dealing all relevant aspects such as legal aspects which vary from one country to the other.

By advancing state of the art in research areas such as SVNs design and capture, it can be anticipated that companies could be assisted in doing through emergence of new business service intelligence offer.

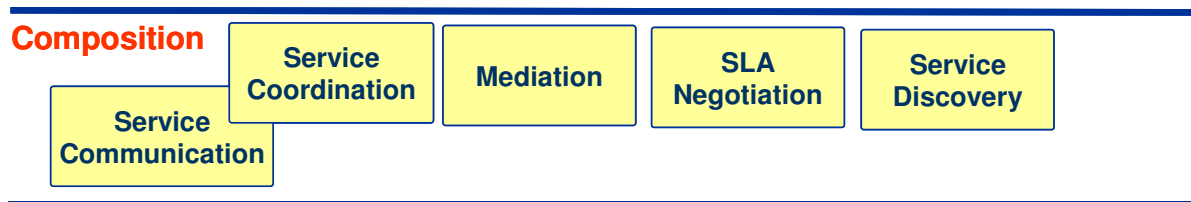
3.2.4 Service Composition Layer

Service Oriented Architecture (SOA) solutions are the next evolutionary step in software architectures. SOA is an IT architecture in which functions are defined as independent services with well-defined, invocable interfaces. SOA will enable cost-effective integration as well as bring flexibility to business processes. In line with SOA principles, several standards have been developed and are currently emerging in IT environments. In particular, Web Services technology provides means to publish services in a UDDI registry, describing their interfaces using the Web Service Description Language (WSDL) and exchanging requests and messages over a network using the SOAP protocol. The Business Process Execution Language (BPEL) allows the composition of services into complex processes as well as their execution. Although Web services technologies around UDDI, SOAP and WSDL have added a new value to

the current IT environments in regards to the integration of distributed software components using web standards, they cover mainly characteristics of syntactic interoperability. With respect to a large number of services that will exist in IT environments in the inter and intra enterprise integration settings based on SOA, the problems of service discovery or selection of the best services conforming user's needs, as well as resolving heterogeneity in services capabilities and interfaces will again be a lengthy and costly process.

For this reason, there is a clear need of evolving the current ways of service execution, for describing services in order to allow total or partial automation of tasks such as discovery, selection, composition, mediation, invocation and monitoring of services.

The composition layer is the core of the service execution and coordination. This layer should provide functionalities such as service registration, service discovery, mediation, SLA negotiation, service communication and service coordination.



3.2.4.1 Service Discovery

Overall description

Within a service-oriented-architecture the discovery of services is the essential building block for creating and utilizing dynamically created applications.

Given a description of a service, the problem of discovering a desired instance of that service can be seen as information retrieval problem that uses keywords to express the desire and uses a document index to match them. However, for mechanizing the discovery task a more fine grained approach to discovery is required, e.g. to restrict the search space along specific parameters like location, provider, price, etc. It can be seen as a search for semi structured entity.

Intended users

- End-users
- Services (that discover other services)

Technological challenge or problem addressed

Dynamic discovery and localization of services.

Positioning with respect to the State of the Art - Gap to fill in

Current Web service discovery approaches in SOA basically lack expressivity. However, current technologies only provide means to describe service interfaces on a syntactical level, providing only limited automation support. Existing solutions for discovery include the UDDI standard, which allows programmatically publishing and retrieving a set of structured information belonging to a Web Service. However it allows to retrieve services by pre-defined categories and keywords, not by their actual semantic properties. There is a lack of means that allow

describing functional and non-functional properties of a service on a semantic level. Only with such descriptions a precise discovery is possible.

Here approaches developed in the context of the Semantic Web, in particular the use of ontologies are a promising approach. Machine processable semantics could be used for describing services in order to allow total or partial automation of tasks such as discovery, selection, composition, mediation, invocation and monitoring of services.

Research efforts have created standards like OWL-S and WSMO-WSML to provide additional expressivity. However, there are currently no automatic discovery procedures that can exploit all the expressiveness of the above formalisms.

A WSML goal specification is identified by the “goal” keyword and consists of a capability and interfaces. A capability description consists of shared variables, pre- and post conditions as well as assumptions and effects. Due to the availability of shared variables WSML goals are more expressive than OWL-S profiles. However, currently there is no concrete proposal for describing the choreography and orchestration. Furthermore, due to the separation of capabilities and dynamic behaviour, it is not currently possible to specify temporal constraints on effects (e.g. “a credit card should only be debited (effect) after delivery (choreography)”).

Other approaches are based on the concept of “resource” [XXX]. Resources are simple components that will have an URI (Uniform Resource Identifier) and respond to basic http get / post / put / delete requests. They encapsulate access to content and services. Note that discovery and composition of resources occur at a different level than discovery and composition of services at the back-end. The nature of discovery and composition at both levels is different and technologies involved may be different. A relevant aspect is that output generated by resources as well as functions like discovery and composition must be context-aware.

Furthermore, SLA awareness and KPI parameters are missing in dynamic service discovery and composition scenarios, both within and across domains.

Scientific and Technological approach proposed

The application of semantic technologies can be applied for reasoning on discovery, along with new semantic matchmaking and retrieval techniques.

In order to achieve this, semantic descriptions for Web Services at many different levels, the creation of service locators based upon these descriptions and the creation of grounding mechanisms for invoking these services are needed. It is important that all of these descriptions and services are tied together into a process allowing for seamless and dynamic location of Semantic Web Services.

There is a clear need on researching on the use of different types of reasoners for the matchmaking of semantic descriptions of services. Special emphasis is to be put on a proper alignment of matchmaking and retrieval techniques to the chosen description and modelling approach for semantic service annotation. Matchmaking and retrieval algorithms are to be designed such that they exploit the structure and semantics in service descriptions to achieve sophisticated discovery behaviour beyond keyword-based matching and taxonomy look-ups.

The approach using semantic technologies consist of:

- ***Service and Domain Ontologies*** – In order to provide a semantic discovery service both service requests and offers need to be described on a semantic level. While there exist some proposals for upper level ontologies like WSMO and OWL-S, we need to refine them and provide guidelines for their usage and accompanying domain ontologies.
- ***Language and Reasoning integration*** – Potentially many different logical formalism can be used to annotate services. Each comes with a specific trade off between expressivity and computational complexity. It has to be investigated for which use cases a particular formalism is suitable. In addition the reasoning engine for a particular formalism needs to be integrated into the discovery context, such that its usage becomes transparent to the user of a discovery engine.

- **Non-Functional Properties** – Research around service discovery has so far paid much attention to the specification of the functional properties of a service, however only little effort has been spent on the investigation of the usage of non-functional properties within the discovery process. Specific ontologies, analytical models and matchmaking techniques have to be developed in order to allow a semantic retrieval on non-functional properties.
- **Field Deployment and Verification of Existing Discovery Strategies** – While many concrete formalism have been proposed only a few case studies have been performed to validate the appropriateness of a particular approach. Further real world use cases are required in order to adopt the existing semantic discovery approaches for the practical needs.

Context awareness in the discovery process.

Another goal is to investigate how different facets of user search and discovery process experience can be optimized by using context-awareness, providing the techniques and methods to apply contextual solutions to information seeking process. In that sense, it is a discovery-based on the end-user perspective applying ontologies and folksonomies, integration, exploitation of tagging info by search engines, etc.

The discovery process should be based as well on KPI and SLA parameters.

Expected results

Optimized discovery mechanisms based on several possible ways (semantic technologies, context, SLA or KPI, etc.).

3.2.4.2 Service Communication

Overall description

To define and implement a Service Web Architecture that brings Web services and Service Oriented Architecture (SOA) to a Web scale based on the abovementioned design principles. This architecture has to comply with interoperability requirements and decouple service providers to enable Web-like n:m relationships.

Intended users

Internal (Communication between services during the execution)

Technological challenge or problem addressed

SOA is largely still an enterprise specific solution. Surveys and experience show that SOA is primarily used for internal integration.

There is a need of an open standard service platform which scales to the Web (billions of services on the future Web). This issue needs to be addressed at several levels, including the one of communication and coordination of services.

Positioning with respect to the State of the Art - Gap to fill in

From the semantic perspective, Semantic Spaces are a communication and coordination paradigm tailored to the needs of Semantic Web extended Web services. Appropriate semantic descriptions of Web services and intelligent mechanisms working upon this, need a solid base in terms of the underlying semantically enabled middleware technology. Triplespace computing, which extends the publication-based communication model of tuplespaces with means to represent and manage semantics. Instead of sending messages back and forth applications exchange information and coordinate activities by writing and reading RDF triples in a shared persistent information space, the triplespace, employing the Web principle of communication through publication.

Scientific and Technological approach proposed

In this category innovative solutions for Message Routing and Event Processing should be proposed.

From the semantic technologies point of view, the solution goes for addressing the service bus and semantic spaces shortcomings:

- The **Service Bus** becomes responsible for the transport of the individual messages and service queries. As the focal point, it enables and realizes the overall operational semantics of the Web of services that enable it to achieve the promised functional semantics of its client-side interface. It takes the functionality offered by the individual services and orchestrates these atomic pieces into a coherent whole in an orderly and consistent fashion. The **Service Bus** brings together these technologies and acts as an access layer to services through the goal based invocation paradigm. based upon the standardization work currently being undertaken within the SEE TC (Semantic Execution Environment Technical Committee)⁵⁴ of the OASIS standardization body.
- **Semantic Spaces:** Current Triplespace computing activities concentrate on RDF as the data model and only marginally address the upper layers of the Semantic Web stack, particularly ontologies, rules or the more expressive formalisms. Information exchange by persistently publishing and reading RDF encoded facts is definitively a promising approach for a considerable number of application settings, as they are currently established in TSC and TripCom. If the filtering of information is required which is based on constraints, rules and policies, it becomes obvious that RDF by itself is not expressive enough. More sophisticated means for filtering and management are, however, necessary for future large-scale service networks

Expected results

Service Bus

Semantic Spaces

Message Routing

Event Processing

3.2.4.3 SLA Negotiation

Overall description

The on-demand provisioning and delivery of services according to well defined service level agreements is still a major challenge.

Intended users

- Software providers face the challenge of predicting the behaviour of services and service compositions with reasonable accuracy. The advent of service-oriented architectures has realized an unprecedented flexibility which comes at the cost of increased internal system complexity.
- Infrastructure providers face the challenge to dynamically allocate resources so that contractual obligations according to SLAs are met with minimal effort and costs.
- **Service providers** are challenged to **negotiate** and manage SLAs in a consistent, transparent and automated way both at business and technical level.
- **Service customers** face the challenge to describe and **negotiate** required service characteristics in a way that is meaningful to their level of expertise but also sufficiently accurate.

⁵⁴ <http://www.oasis-open.org/committees/semantic-ex>

Technological challenge or problem addressed

The whole process of negotiating SLAs and provisioning, delivery and monitoring of services must be automated in order to allow for dynamic and scalable service consumption.

Positioning with respect to the State of the Art - Gap to fill in

Current SLA management does not support multiple management layers with combined SLAs as required in business relevant environments. WS-Agreement is an OGF recommendation which is independent from the content of the agreement and does not provide any negotiation features. Moreover, there is no consistent model to schedule and optimize combined SLAs and eventually enact, monitor and control them on different supporting infrastructures.

Translation of high-level requirements and policies to the infrastructure level while aggregating in parallel infrastructure behaviour data to be used by higher level services creates full transparency across the IT stack and consequently allows for best optimization of performance, resource usage and costs. As pointed out in [NESSI-Grid 07 and NG 06] this kind of transparency is the most important aspect missing in today's' business IT systems.

Scientific and Technological approach proposed

- Providing a recursive SLA management framework that covers different organizational and technical layers within a service provider, this way supporting automatic negotiation, control and optimization facilities
- Systematic approach to embed automated SLA enforcement into the infrastructure layer.

Expected results

Automated e-contracting lifecycle

3.2.4.4 Mediation

Overall description

Mediation defines interoperability at the functional, data and process levels. Mediation covers different aspects:

- Data mediation: techniques for combining Ontologies and for overcoming differences between Ontologies.
- Process mediation: overcoming differences in message ordering, etc.
- Protocol mediation: solving differences among different communication protocols

Intended users

Domain experts

Service coordination (internal)

Technological challenge or problem addressed

Overcome data and process heterogeneity problems

Positioning with respect to the State of the Art - Gap to fill in

Heterogeneity problems and mediators have been intensively investigated in the last decade but still the key solution that would enable the decisive leap towards automation is yet to be found. Currently most of the semantic solutions to overcome data heterogeneity are based on using a single domain ontology. Semantics is changing the problem specifications and service orientation paradigms offers new ways of designing, deploying and using mediators while

in the same time, poses new challenges and sets new requirements. That is, data and processes can be formally and unambiguously described while services and service oriented architectures allow the development of decoupled, transparent and flexible software components, including mediators.

Scientific and Technological approach proposed

- **Advanced support for data mediation.** Semi-automatic design-time tools should be developed in order to allow domain experts to identify and capture the heterogeneity problems between different models of overlapping domains. Special attention should be given to user profiles and expertise levels in order to separate the tools for trained domain experts and the tools designated for casual users of ontologies. Furthermore, at this level, alignments between various models will be part of a community validation process where users can add and remove links between the models in order to achieve and maintain agreed upon interlinked models.
- **Advanced support for process mediation.** Heterogeneity appears on the process level as well no matter if these processes are enterprise internal processes or public processes used in describing the visible behaviour of particular services. Such heterogeneous processes need to be part of collaborative scenarios that can range from simple peer-to-peer interaction to complex compositions. Semi-automatic tool support allowing the tailoring of such process in order to overcome the heterogeneity problems should be provided. Further more such tools should support annotation of existing process representation standards with semantics based on ontological domain models.
- **Service Mediation by Mediation Services.** Mediator systems able to resolve specific types of heterogeneity problems should be encapsulated and deployed as mediation services. Such services should be developed for well defined mediation scenarios while preserving the generality of their offered functionality.
- **Semantic descriptions for the mediation services and mediation libraries.** Mediation services should be semantically described as any other resources. In this way their functionality can be properly advertised and their intended usage explicitly stated. Furthermore, they can become part of intelligent mechanism for service discovery, composition and invocation. Additionally, such semantically described mediation services will be organized in semantic mediation patterns that can be directly applied on complex heterogeneity scenarios. In addition, such services should be organized in libraries supporting intelligent mediation service retrieval (by providing customizable mediators classifications), patterns construction (based on the semantic descriptions of the mediation service and on the mediation goal to be achieved) and governance mechanisms (by exploring service and patterns dependencies and impact analysis).

Expected results

Semantic mediation at the data and process level.

3.2.4.5 Service Coordination

Overall description

Service Coordination defines a composition of services into an executable workflow. It also includes orchestration, which defines the execution of a composite process (business process) together with a conversation between a service requester and a service provider within that process, and choreography that defines formal specifications of interactions and processes between the service providers and client.

Intended users

Service providers

Client

Internal (composition layer)

Technological challenge or problem addressed

Coordination involves methods for Service composition, starting from web service descriptions at various levels of abstraction, specifically, the functional level and process level.

Positioning with respect to the State of the Art - Gap to fill in

The composition of services has received considerable interest in relation to its ability to support business-to-business or enterprise application integration. However, this area is yet in a very early stage of its development. The business world has developed a number of XML-based standards to formalize the specification of Web services, their flow based composition and execution

Several other techniques have been proposed, mostly based on AI Planning or on logical deduction, but all of those still have severe shortcomings. The existing techniques either: (I) largely or even completely ignore the constraints given in the background ontology in which the web services are specified; (II) largely or even completely ignore the complex inner behaviour and interfaces of web services; (III) have severe scalability problems, solving only toy problems with few services; or suffer from several of these deficiencies.

Techniques for service choreography play a key role in creation of new opportunities for collaborations between service requesters and providers, and thus for creation of new services. The Choreography part should define formal specifications of interactions and processes between the service providers and clients. Current approaches to service choreography languages have been criticized for being too procedural-oriented. With the move towards service-orientation, where entities are autonomous and need to agree on the collaborations between them, where no central point of control might exist, a more declarative modelling style for interactions is required (i.e. "what" without having to state the "how"). Moreover, reasoning techniques for such a language that would enable a flexible and dynamic integration of service requesters and providers in a collaborative environment are currently missing.

The current approach for service composition is primarily syntactical: Web service interfaces are like remote procedure calls and the interaction protocols are manually written, which is a time-consuming and tedious programming task involving a large amount of human labour.

The problems with the current semantic service construction approach are the following:

- Too detailed descriptions: The formal descriptions of services must all be flawless and complete.
- Completely static worlds: No consideration is given as to whether the problem may change during the attempt to solve it (which may often take some time), and no attention to notions of learning, i.e., whether previous solutions can help one in the construction of new compositions.

On the other hand, quality of service (QoS) aspects can be defined for services, but it is a major challenge to define QoS aspects for service composition, consisting of services in different technologies, and to map them to the underlying technologies and to ensure that the defined aspect will be met. To ensure availability of a system, the relocation of services might be necessary.

Scientific and Technological approach proposed

Focusing on the provision of methods and tools to achieve **on-the-fly Service composition**.

- The key technology will be to use lightweight modelling and construction techniques (services are annotated in a simple and flexible language).
- Service construction then exploits the information in these descriptions, and fills in the holes in a dynamic, Web 2.0 way, where the vast amount of information available on the Web, and the experience from wide usage, is consequently exploited.
- Context-specific functionality, such as context adaptation involving service composition, where a core service is composed with a set of services providing context-specific information.

- Composition based on resources mash-up and piping
- Management of SLA across service composition
- Management of security aspects in relation with service coordination.

From the semantic perspective the main objectives of research are;

- Development of a scalable tool for service composition with powerful background ontologies and partial matches
- Development of scalable tools for service composition with plug-in matches, dealing with Business Policies. Scalability is particularly urgent, since enterprises deal with thousands of services from which the composed service should be combined. Further, business policies -- rules governing how services can be executed within or between enterprises -- are of paramount importance.
- Integration of techniques for functional-level and process-level Web Service Composition. The idea is to combine techniques from functional-level and process-level composition, first by establishing ways for their interplay, later by integrating their underlying core principles.

In the case of the choreography, the main research areas are:

- **Declarative choreography language** which should allow for formal specifications of interactions and processes between the service providers and clients.
- **Reasoning tasks for choreography** that should be performed using the declarative language. Verification techniques such as contracting or enactments are examples of reasoning tasks. Such techniques will enable an automated, flexible, and dynamic integration of service requesters and providers in a collaborative environment.
- **Tool support for choreography**, by implementing an engine to support the execution of interactions, as well as to support reasoning in the proposed declarative language.

Expected results

Dynamic composition of services.

Orchestration and choreography enhancement.

Among other, QoS, SLA and security aspects should be dealt with.

3.2.4.6 Service Registration

So far, the publication of Web services have been tackled in several different ways, but always with only partial success. UDDI and ebXML have been around for a while, but they have still to convince users that they can really provide usable and flexible solutions for the efficient publication (and discovery) of services. Orthogonally, the semantic community has been pushing richer service descriptions, based on description logics and ontologies, to support accurate service discovery, but even proposals like METEOR-S and Pyramid-S, which provide semantically enriched means for service publication, do not fully convince. The semantic infrastructure allows us to better describe services, but it also forbids the complete control over the registries since it imposes too heavy constraints on publication policies.

These approaches (and others) allowed us to learn some interesting lessons, which can also be seen as issues still open for further research:

- Registries (and services) can always be heterogeneous. Solutions that are only able to cope with a single technology or a dedicated standard are not enough.

- Fixed and predefined descriptions of services do not pay off. Service descriptions must accomplish different needs, and thus they must be flexible, customizable, and provide different users with specific views. Users play a key role in organizing and accessing such information.
- A single predefined service description is not enough to cover all the different aspects efficiently. Services must be described by means of dedicated facets, which render the different angles (functionality, QoS, tests, monitoring data, etc.) that belong to the service. Some of these facets, can be standard (e.g., the functional interface), while others could be proprietary.
- Service data are embedded in many different artefacts that are already available, but that cannot communicate directly with current service registries. This is why, special-purpose crawlers might be used to search available artefacts to grab such data and thud populate registries with significant data automatically.
- Centralized general-purpose repositories (e.g, the UDDI Business Registry started by IBM, Microsoft, and SAP) have already failed. General-purpose registries still exist, but they are mainly used for benchmarking and research. The community is mainly interested in dedicated and proprietary registries, where the amount of public information can be set dynamically.
- Even if all the main proposals have moved towards distributed approaches, we think that this distribution cannot be super-imposed, neither can it be organized around some predefined ontologies. Different organizations have different needs, and thus the way they organize services must support their views.
- Registries can easily be "federated" to create distributed virtual registries, where service information is suitably moved closer to its users. The federation allows registries to both replicate and share significant information.

All these lessons clearly highlight two mutual independent research directions:

- The definition of a facet-based infrastructure for the full and customizable description of services (and not just Web services). This infrastructure should be able to accommodate "any" data related to a given service, but it should also be able to organize them for their efficient retrieval.
- The definition of suitable architectural means that allow heterogeneous registries to cooperate efficiently by sharing important information and also by exchanging information about the services of interest. This system should also be able to search as many sources as possible to find interesting services and contribute them to the community.

3.2.4.7 SBS/SBA Modelling

[Missing input on this topic, to be developed]

3.2.5 Service Layer

3.2.5.1 Service Execution

Grounding: Legacy systems represent valuable assets for most of their owners and usually completely replacing them is not an option. As such, methodologies that will allow the integration of these systems with the new emerging paradigms and technologies have to be developed. For example, XML Schemas and XML data have to be lifted to ontological level in order to allow semantic-aware systems to act on this data. In addition, since all tasks related to discovery, selection, composition, etc. operate in SESA on semantic descriptions the link between semantic service and underlying technology for communication (e.g. HTTP, SOAP, etc.) needs to be defined. The basis for grounding has been established within the W3C Semantic Annotations for WSDL WG (SAWSDL WG) allowing hooking semantic descriptions with WSDL elements.

Goals and Tasks

- Semi-automatic tools allowing the creation of transformation between syntax-based and semantic models - specifying the transformations between Web service XML messages and the semantic data currently

requires deep knowledge both of the structure of the XML message and the ontology. Methods for semi-automated creation of grounding should be created.

- Grounding to other specifications – grounding definitions to other specifications apart from WSDL should be defined. It should be possible to use e.g. REST services with semantic descriptions, etc.

Bind SLA to specific service instances: The management capabilities required for an SLA-aware service management of service-oriented systems is fine-grained and specific to the scenario. Current approaches to manageability infrastructures, like WS-Management or WSDM do not reflect the specific requirements of these scenarios, particularly concerning the management information modelling and the translation of SLAs into the required information.

Furthermore, SLA awareness is missing in dynamic service discovery & composition scenarios, both within and across domains.

The current challenge is to provide a unified manageability inter-face tailored for an SLA-aware management of the services and to provide the basis for including SLA-awareness in the processes of service discovery & composition.

3.2.5.1.1 Security policies enforcement

The security objectives concerning the digital assets base are expressed in terms of confidentiality (non-disclosure to unauthorized persons), integrity (non-alteration of content by hackers) and availability (the ability of licensed users to use these assets without being hindered by unintentional or malicious acts).

There are interactions between security and trust: Information security is related to the level of trust. If trust exists, security measures can be lighter. If everyone has trust, the security policy will be to do “relax”. If there is no trust in anyone or anything, a security based on the constraint can be organized. It will give the illusion of the confidence in the system, at least at the beginning, but then things deteriorate quickly, because entities do not adjust well to policy of dictatorship or terror.

If safety measures are strict and effective, to trust can be created. Should the opposite occur, they are ineffective, it can however trust there, but it is overvalued and puts the citizens, the companies and all the system in the danger: then these security measures have no sense and can be considered purely and simply, as a manipulation of the public opinion.

Usually, there is a dynamic relationship: security measures increase trust in the system, and a relationship built on trust will lower the security measures. There is thus a dialectical adjustment to minimize costs and the disturbance caused by the introduction of security measures.

We must accordingly distinguish between two concepts: security and trust. The two concepts must be separated, those that serve security and those that serve trust. In the networks, the systems and more widely infrastructures, there will be tools and algorithms to create, manage and maintain trust. The protection of infrastructures will always be created by mean of the encryption of content to protect sensitive data's, a firewall to protect access to a system's frontiers through access control, an antivirus, an intrusion detection system.., and also for the security: digital signature, digital tattooing of content to dissuade a pirate from copying and using software).

For example, Intrusion Detection Systems and Intrusion Prevention System can be considered as systems in which security and trust are mixed together. Data is collected and the variables calculated which facilitate decision according to behavioral models (normal, not normal) whether or not an operation is legal. In fact, this involves the introduction of a trust factor that facilitates deciding yes or no, about performing a transaction or certain traffic over a network.

Today we know that it is not possible to protect a large, open, interconnected system only with security components. Thus to secure it, other means of protection are employed. In place of protecting there is prevention, repression, user awareness, attackers are dissuaded and deluded, and the law will try to do the rest.

Protection of sensitive digital commodities (data, documents, programs, concepts...) belonging to responsible entities (their authors or owner organizations) represents the new challenge of the administrators of the networks deployed all around us. A second challenge need to be considered is that of the protection of the freedom of citizen, the survival of companies and the future of democratic countries, in all the fields, in private or public life, in the civilian world or in the defense.

The digital universe is intruding into all sectors of activity: industry, trade, finance, defense, administration, health, education, justice and environment. The security of the digital world has become a fundamental priority for the citizen with respect to his individual freedom and protection of his computerized identity and privacy (privacy policy); also for the company, with respect to the protection of its computerized industrial assets, the security of its business transactions and the trust level of its information networks; and finally for the state with respect to the reliability of operations and the reduction in the vulnerability of large and critical infrastructures .

It is crucial that the security rules are open, transparent and well understood by everybody, for lack, people are unaware and these rules are out of their control. We must offer tangible security that is verifiable or verified and rather certified by a trusted authority in order to get confidence in the security tools we are offered.

3.2.5.2 Integration Services

[Missing input on this topic, to be developed]

3.2.5.3 Lifecycle Management

[Missing input on this topic, to be developed]

3.2.5.4 Reasoning

The SESA (???) necessitate effective reasoning for different tasks such as service discovery, process and data mediation and integration. To enable processing of these tasks in an automated manner, the SESA utilizes machine reasoning over formally represented service specifications.

We are developing Integrated Rule Inference System (IRIS) which is a scalable and extensible reasoner tool for Web Service Modeling Language (WSML) . The system implements different deductive database algorithms and novel optimization techniques.

Goals and Tasks

- Reasoning techniques with large data sets – In the context of Semantic Web, applications might require vast volumes of data to be processed in a short time. Current reasoning algorithms are developed rather for small, closed, trustworthy, consistent, and static domains. Therefore these algorithms need to be extended and adapted in order to be applied on large and dynamically changing knowledge bases. One challenging approach to achieve a scalable reasoning is to combine existing deductive and database techniques with methods for searching the Web (utilizing semantic annotations). This line of research considers reasoning in distributed environments as well.
- New techniques for Description Logics reasoning – Description Logics (DLs) are a family of knowledge representation formalisms characterized by sound, complete and (empirically) tractable reasoning. However applications in areas such as e-Science and the Semantic Web are already stretching the capabilities of existing DL systems. Key issues here are the provision of efficient algorithms that allow (advanced) applications (1) to scale up to knowledge bases of practical relevance and (2) to leverage expressive languages for capturing domain knowledge.
- Reasoning with integrating frameworks based on classical first-order logic and non-monotonic logic programming – Two lines of research will be explored:
 - Reasoning with decidable fragments of such integrating frameworks.

- Reasoning with un-decidable fragments using proof-theoretic techniques.

3.2.5.5 Service Modelling

[Missing input on this topic, to be developed]

3.2.6 Abstracted Infrastructure Layer

3.2.6.1 Infrastructure Abstraction

Infrastructure management: Advancements in infrastructure management are mainly in the areas of autonomies and advanced diagnostics. Autonomic capabilities are policy driven and targeted at configuration and optimisation, and resilience. Advanced diagnostics are beginning to incorporate statistical models, mining and predictive analytics to drive sophistication in resource allocation and failure model analysis.

There is a plethora of virtualization solutions (XEN, VMWare, OpenVZ,, ...) and Grid middleware techniques (Globus, Condor, ...). While some harmonization has been achieved in the Grid area, this is totally missing for the virtualization space and consequently also for the combination of both fields.

Introduction of SLA map-ping with some enforcement support will significantly benefit dynamic optimisation, by reflecting criticality from upwards in the stack. Application of predictive analytics to resource consumption will enable proactive and pre-emptive reconfiguration decisions to safeguard SLAs and maximise efficiency.

The next phase in research will provide harmonized management interfaces for various virtualization/Grid-technologies thus enabling their transparent usage depending on SLA-driven requirements.

Virtual Machines Technology: Virtual machine technology has been around in various incarnations since the 1960s. It has obvious benefits when compared with running the operating system directly on the hardware e.g., server consolidation, improved resource utilization and the ability to migrate an operating system to a different physical machine while it is running. However, this technology also presents some significant challenges, especially in the area of performance when compared to running directly on hardware. New research aims to narrow the gap between the performance of selected workloads running on virtual machines (i.e. atop hypervisor) and the performance of the same workloads running directly on hardware by improvements to the core hypervisor and to guest operating systems.

This current challenge is centred on the following areas:

- Optimizing the core hypervisor and operating systems of the underlying virtualisation technology (e.g., Xen or KVM as the hypervisors and Linux as the guest).
- Optimizing IO performance (as measured by throughput, latency and CPU utilization) through both optimization of virtual IO mechanisms and enabling direct hardware access from the virtual machine to the IO devices, utilizing IOMMUs [IOMMU] and/or self-virtualising devices.
- Provide a management interface enabling the VEEMS to control Virtual Execution Environment (VEE)-level operations required for service-level optimization (e.g., create VEE, allocate additional resources to VEE, monitor VEE, migrate VEE, and adjust performance for workload characteristics).

3.2.7 Infrastructure Layer

3.2.7.1 Resources Management

Resource interoperability, portability, and reusability: One of the main issues in the domain of infrastructures for services is to provide the flexibility required to manage different virtualisation technologies and the ability to interoperate them across different administrative domains (grid nodes or service sites). This requires the disintegration of the boundaries that separate different virtualisation technologies within and across organizations.

For this purpose, there is a need to define a reference architecture for SOI. The architecture should focus on the challenging capabilities required to manage VEEs and support their dynamic deployment and relocation. Key to this vision is standardization, as the aim is to create interoperable, portable, and reusable components.

In developing the reference architecture, the objectives are:

- Demonstrate the versatility of these concepts and their applicability to different implementations of VEEs, such as Virtual Machines (VMs) and Virtual Java Service Containers (VJSCs).
- Provide a foundation for service management, including service definition, SLA management, accounting and billing.
- Define the interfaces for generic control of virtual execution environments (VEEs). This interface will follow and possibly extend applicable industry standards.
- Provide a global architecture for VEE provisioning technologies, including a policy engine and being able to deploy federated infrastructures comprising multiple sites or administrative domains.
- Investigate new security challenges introduced by the architecture, such as trusted execution in multi-tenancy environments

Relocation Enablement: One of the main objectives is to remove the technical “boundaries” which limit today’s virtualisation technologies. In particular, the challenge is to build an infrastructure in which virtual machines can be dynamically relocated to any node regardless of location, network and storage configurations, administrative domain and hypervisor technology. To enable “relocation without boundaries”, one has to build on the existing technology of "virtual machine live migration". This technology, however, is associated with serious barriers that limit the flexibility of migration across nodes. To realize this objective, exploration is required to improve directions and solutions that will allow:

- Migrating a virtual machine across different subnets.
- Migrating a virtual machine across hosts which do not have access to the same storage (e.g. using local disks).
- Migrating a virtual machine across hosts which do not have an identical configuration (e.g., virtual switch configuration)

3.2.7.2 Data Management

In the **emerging service economy** businesses have to provide continuous access to their services to both customers and trading partners. With an increasing number of clients the underlying service multi-tier infrastructure has to face immense scalability and availability requirements. Often a service is deployed at a local or regional level and then its scope is widened and the number of users gradually increases. The underlying infrastructure requires scaling in a cost effective manner with the size of the service market.

Current state of the art exhibits a number of technology roadblocks that prevents current systems from attaining the aforementioned goals. Nowadays, technological solutions to scalability in service multi-tier architectures rely on scaling-up, i.e. on repeatedly replacing high end hardware by even more powerful systems. However, the cost evolution of scale-up solutions revealing a decrease in the system price per throughput has come almost to a standstill. As a consequence an increasing demand for database throughput will become very cost intensive. In fact, solutions with large scale requirements are precisely those with most substantial profits due to the large number of clients involved. The total cost of ownership for large scale multi-tier service infrastructure is driven by two factors: 1) the cost of licenses for a parallel database management system, and 2) the cost of the top-end mainframes and their replacement by more powerful ones when the capacity is exhausted.

For an integral solution to large scale service infrastructure two fundamental characteristics of approaches seem to be promising:

- 1) The approach should be based on scaling out, i.e. increasing the infrastructure capacity by adding off-the-shelf hardware (e.g. blade PCs). This will avoid the use of expensive mainframes and also the cost of substituting them by a more powerful hardware when unable to cope with increased loads.
- 2) The clustering approach, called vertical replication will allow for simultaneously replicating both the application and the database tier. Embedding the replication logic in the application server tier paves the way for scaling out based on inexpensive off-the-shelf databases (e.g., open source ones).

3.2.7.2.1 Scientific and Technological Challenges

Attaining the proposed goal involves sorting out a number of scientific and technological obstacles. Scaling multi-tier architectures is challenging due to both tiers being inherently stateful, i.e., they maintain data that has to be kept consistent at any time. Thus, replicating the servers and their data requires coordination among the replicas. Current solutions for service oriented application servers and databases fail to scale beyond 10+ replicas, after which no further increase in system throughput can be achieved. Therefore, businesses currently resort to replicating only the stateless part of applications and buying high-end hardware and installing expensive parallel database technology. Application servers are clustered and in order to provide consistency, they share a single database system. Since the database becomes the bottleneck, the only solution for this kind of architectures is a scale-up approach where a high-end multi-processor mainframe is used. When its capacity is not enough an even more powerful mainframe needs to be bought. Basically, current solutions are based on high-end and very expensive parallel database management systems running on high-end hardware. This results in what we call a horizontal replication approach.

Since horizontal replication completely separates the application tier from the database tier, it often provides only weak levels of consistency. Inconsistency might not only occur in failure cases but even during normal processing mode. For instance, data across the tiers might not be completely consistent violating the idea of “1-copy-consistency”, i.e., the concept that all copies of a data item are synchronized. Furthermore, exactly-once semantics (i.e., a request leads to exactly one successful execution across all tiers) can easily be violated. Additionally, atomicity of execution might not be provided, i.e. execution might be rolled back in the database tier due to a failure while this does not happen in the application tier. Systems that provide full consistency incur a considerable overhead due to the complexity of interactions between the two tiers. Additionally, in horizontal approaches the database system appears as one single component, although it might be replicated and distributed transparently. Depending on how access to the database system is handled, this single access point might (and usually does) become a bottleneck. Finally, physically separating and placing by necessity the application server and database tiers on two different clusters leads to considerable coordination overhead between the two tiers.

Given the shortcomings of the horizontal approach, a novel architecture seems to be promising: *vertical replication*. In this approach, an application server and an off-the-shelf non-replicated, non-clustered database system compose a single replication unit. Typically, such a unit is put together on one machine. The entire replication logic is embedded in the application server allowing for a tighter and simpler coordination of replicas. We believe that such an approach is capable of eliminating the problems associated with horizontal replication. Businesses rely on the enhanced application server middleware to provide all scalability functionality and can resort to use a commodity database system as backend. By having one global coordination protocol, consistency will be easier to achieve. Communication between application server and database server are put to a minimum since it is local within one machine. Finally, the application server can take full advantage of sophisticated caching strategies and other advanced application server technology without concern that this might cause data consistency problems at the database tier.

The following challenges need to be addressed is to increase the scalability of service oriented infrastructures.

Consistency: Current replication solutions for application servers either resort to very weak levels of consistency or provide the very strong consistency level 1-copy serializability (1CS). Weak levels are not acceptable in update-intensive environments. 1CS, on the other hand, leads to poor scalability [Pacitti05]. Recent developments have proposed a new correctness criterion, 1-copy snapshot isolation (1CSI) [Lin05] that avoids the bottleneck of 1CS. While 1CSI provides a consistency level very close to 1CS, it avoids read-write conflicts by providing read operations with a snapshot of the data. Only write-write conflicts, which occur much less than read-write conflicts, have to be considered.

Reduction of Update Costs: Many existing replication solutions rely on full replication, that is, each replica contains the full dataset. If full replication is applied each update on any replica has to be propagated and executed at all other replicas. The problem can be relieved if an update transaction is only executed at one replica while the others only apply the updates, which can be significantly faster. However, at some point, the capacity of any new replica will mainly be dedicated to install remote updates from other replicas. Thus at this stage no increase in the overall throughput can be achieved. A solution to this inherent scalability ceiling is to use partial replication.

Self-Healing: A fundamental challenge of any replication approach for stateful environments is to be able to self-heal in the advent of crashes and voluntary exits of individual replicas without any noticeable effect on the service users. In particular, any changes executed on the leaving replica must be known in the rest of the system. Additionally, in order to keep the same degree of availability, one has to be able to add new replicas to the system or let crashed ones rejoin. Before a joining replica can become an active member it has to have a correct state of the system. In the context of database replication, online recovery has been dealt with for example in [Kemme00, Kemme01, Jimenez02] and in the context of online reconfiguration for application servers in horizontal replication [Wu04, Wu05]. These approaches need to be combined and adjusted for vertical architectures.

Self-Optimization and Self-Provisioning: Self-optimization refers to adjusting any form of configuration parameters in order to optimize performance. This will involve the configuration within a replication unit of application server and database system, and dynamic load balancing across replication units. In regard to the former, a global view looking at both application server and database tier is needed. Self-provisioning refers to the capability of the system to grow and shrink dynamically such that it only uses resources that are necessary to cope with the current load. Self-provisioning needs to continuously adjust the system configuration such that all available resources are optimally utilized. Thus, self-provisioning must be able to add new replicas to or remove replicas from the system. Self-provisioning requires a profound understanding of the scalability of the system depending on a detailed analytical model to predict the “optimal” number of replicas.

Communication management: Communication needed to coordinate replicas has significant performance implications: General purpose communication subsystems do not use available bandwidth efficiently, they impose significant overhead to the host CPU reducing available cycles for application processing, and they incur high latencies for short messages used extensively for synchronization purposes. All these factors have a significant, detrimental effect on system scalability. Approaches for next-generation communication subsystems and protocols that maximize bandwidth, reduce host CPU overhead, and improve latency in system area networks, such as 10 Gigabit Ethernet, Myrinet, and Infiniband can be found in [MPB07, PKK07, MXP06, ABD98, DBL97]. The main challenges therefore are use systems area network technologies for the before mentioned issues in a vertical replication approach.

3.2.7.3 Infrastructure Modelling

Infrastructure modelling approaches can be roughly grouped into 2 use cases: infrastructure management and infrastructure analysis.

The most common approach in the area of infrastructure management is the so-called common information model (CIM) which has been developed by the Distributed Management Task Force (DMTF) [DMTF-CIM 07]. The mission of the DMTF is to lead the development of management standards for distributed desktop, network, enterprise and internet environments. As part of this mission CIM realizes a standardized information model that allows for describing networked computer systems. The actual information model is based on the meta model MOF [OMG-MOF 07] and consists of a core model and a common model. The core model introduces general entities such as qualifiers, core elements, managed elements, statistics etc. The common models contain specific descriptions for applications, databases, devices, events, network, physical entities, users, policies, support information, systems and interoperability information. In addition, DMTF also developed a Web-compliant binding of the CIM under the Web Based Enterprise (WBEM) initiative. This includes an xml-based CIM encoding and CIM operations over HTTP. A dedicated workgroup has been recently introduced to address the modelling of virtualized, partitioned and clustered computer systems [DMTF-SVPC 07].

The service modelling language (SML) is a recent W3C initiative [W3C-SML 07]. It significantly differs from CIM as it is a meta model (comparable to MOF) and allows for rich model interactions (e.g. distinction between observed

state and desired state, transformability/validation/extensibility) and highly expressive constraints (through schematron).

The IT Infrastructure Library (ITIL) defines guidelines and best practices for the operation of IT infrastructures, including processes, organizational issues and tools. Central to the ITIL processes is the Configuration Management Database (CMDB). It is the cornerstone to the integration and sharing of information between processes. The CMDB contains information about the IT infrastructure by representing software, hardware, documents, people information (among many other IT assets) as Configuration Items (CI). In addition, the CMDB also contains information about the relationships between CIs. Typical ITIL implementations are based on CIM.

Modelling approaches for system and in particular infrastructure analysis include queueing networks, Petri nets and many variations thereof [Kounev 2005].

A specific approach is the UML profile for Schedulability, Performance, and Time Specification [OMG-UMLSPT 05] which defines standard paradigms of use for modelling of time-, schedulability-, and performance-related aspects of real-time systems" that (1.) enable the construction of models that can be used to make quantitative predictions regarding these characteristics; (2.) facilitate communication of design intent between developers in a standard way; and (3.) enable interoperability between various analysis and design tools.

The UML Profile for Modeling and Analysis of Real-time and Embedded Systems (MARTE) is currently under development and adds capabilities to UML for model-driven development of Real Time and Embedded Systems (RTES). It provides support for specification, design, and verification/validation stages. This new profile is intended to replace the existing UML Profile for Schedulability, Performance and Time.

3.2.8 Monitoring

With monitoring we intend all of those functionalities which are needed not only to strictly monitor the behaviour of a service or a system, but also to understand this behaviour with the aim to take or at least suggest proper remedial actions if the service or system does not behave according to the agreed SLAs.

In the wake of service oriented systems and applications, monitoring capabilities have grown in importance. Due to the intrinsic distributed nature of these systems, it has become extremely hard to verify their correctness at design-time using classical techniques. Since systems are built through distributed collaboration, and loose coupling techniques, partner services are free to evolve independently, within the limitations of established service level agreements. However, this means that a system's architecture is continuously evolving. Since a system can have a high number of stake-holders, it is important to always keep an eye on these evolutions, to discover anomalous behaviours as they arrive, and to try to take action to keep the system from failing its requirements.

Therefore, under the term monitoring we group all the functionality needed to capture a system's behaviour, to perform "intelligent" reasoning on the system, and to try to take (or at least suggest) a proper remedial strategy that can keep the system from entirely failing its requirements. The term is conceived in a broad sense, which has repercussions on the entire system platform. Monitoring is a cross-cutting concern that must take into account all the levels of our platform, from the infrastructure (and its resources), to the service layer, and to the consumer layer. Thus, there is a need for models, languages, and tools that can provide different levels of abstractions.

Considering the current state of the art in this field, many research issues need to be considered, amongst which:

- There is the need for models and languages that are expressive enough to capture both functional and non-functional properties, and that can be used at different levels of abstraction. For example, we need ways of capturing business-level stake-holder requirements (in terms of service level agreements) and of transforming them into properties that are verifiable during execution.
- There is a need to study the advantages of having domain specific models and languages, as well as having generic languages.

- There is a need to study both centralised models, for monitoring the behaviour of single or centralised workflow-based composite services, and distributed models, for coping with service collaborations that are built on event-based collaborations.
- There is the need to study models that provide different degrees of timeliness. There is an intrinsic payoff between timeliness and expressive power. In fact, we can envision post-mortem approaches to monitoring that can be used for statistical purposes, and therefore provide high degrees of expressive power. In this case we are not interested in providing suggestions to the system to keep it on track. We can envision on-line monitoring approaches that are performed asynchronously. These might have lesser expressive power but could be used to provide feedback to the system which might try to perform some sort of rollback. Or, we can envision on-line approaches that are performed synchronously to the system's execution. This is a more invasive approach in which a system is interrupted during execution for maximum timeliness. Since the performance impact is higher, the expressive power would need to be reduced. In this case, the error is found immediately and recovery actions can be more effective.
- We need dynamic monitoring models in which the amount of monitoring that is performed may depend on a number of issues: a particular moment in a system's life-cycle, the context in which the system is run (including infrastructure and resources), the needs of a particular stake-holder, etc. For example, we may need to be able to switch between the various models described at the previous point.
- There is the need to study models for supporting recovery strategies in service based systems. We need to study both models that provide forward recovery techniques and models that provide backward recovery techniques. These studies will need to take into account the distributed nature of the system, and peculiarities such as transactions, stateful/conversational services, stateless services that have persistent effects, etc.
- There is the need to study models that can be considered as cross-cutting concerns with respect to the business models. We need to be able to keep a clear separation of concerns that can make the models more manageable. Only through of weaving techniques, such as aspect oriented programming, or proxy-based solutions should the monitoring and the business models be mixed.

3.2.8.1 Collaborative Business Intelligence

In complex cross-organisational BPM contexts, where multiple partners collaborate together towards a common goal, traditional Business Intelligence (BI) systems are no longer sufficient to maximize the effectiveness of the monitoring and decision making process. Two new significant requirements arise:

- Cross-organization monitoring and decision making: Accessing local information is no more enough, users need to transparently and uniformly access information scattered along several heterogeneous BI platforms.
- Pervasive, secure, and personalized access to information: Users require that information can be easily and timely accessed in different contexts, through devices with different computation and visualization capabilities (PCs, palmtops, mobile phones), and with sophisticated and customizable presentations.

Thus, there is a need for designing a collaborative network of BI platforms whose members can offer monitoring and decision support functionalities to users in the form of services. Users will access information distributed over the network in a pervasive and personalized fashion. Access will be secure, depending on the access control and privacy policies adopted. The information yielded by each member BI platform may change quickly in time to keep pace with the evolution of the business domains; thus, relevant information gathered from members will be dynamically reconciled based on the user's view of the business domain, expressed by means of local ontologies.

Considering the state-of-the-art of current BI platforms, implementing the scenario outlined above requires a set of research issues to be addressed, among which we mention the following:

- 1) Devise an integrated model aimed at homogeneously representing (i) the context the user is operating in, including the user preferences; (ii) the different types of information made available by the BI platforms; (iii) the constraints in accessing information the user is subject to.

- 2) Develop solutions for efficiently processing the queries formulated by the user and returning the best results compatibly with preferences and constraints.
- 3) Integrating and resolving the access control policies of the peer the client is connected to, against those of the peers that return information chunks.
- 4) Find mapping between the single local ontologies aimed at integrating information resulting from distributed queries over the peer network.
- 5) Devise techniques for reconciling and integrating the partial information returned by the single peers into a consistent “picture”.
- 6) Understand how a measure of the quality of the integrated data can be given to the user.

3.3 Description of Non-Functional Aspects

[Description of each envisioned views (models described in each yellow circle) as well as systemic coordination (the three rings) at coarse-grained according to some predefined template covering the following aspects:

- *overall description,*
- *intended users,*
- *technological challenge or problem addressed,*
- *positioning with respect to SofA (State of the Art), Gap to fill in,*
- *Scientific and Technological approach proposed (different approaches can be provided, e.g. semantics, security, SOA),*
- *expected results (it includes benefits)*
- *...]*

3.3.1 Introduction

Non-functional properties are the essential characteristics that ultimately determine the context under which a certain service can be used and also determines the value of that service. Consequently, sound and comprehensive management of non-functional properties is a prerequisite for the successful further evolution towards a service economy, where IT-based services can be flexibly traded as economic good, i.e. under well defined and dependable conditions and with clearly associated costs.

Non-functional properties need to be considered in 3 major dimensions, the horizontal system lifecycle and the vertical business/IT stack and the various non-functional categories such as security, performance, availability etc.

The first dimension is the system lifecycle, including architectural phases, development and testing phases, deployment phases, composition phases, provisioning phases, operation phases and decommissioning phases. Non-functional properties need to be considered in all these phases at different granularities and from different stakeholder perspectives (architect, administrator, service provider ...). The fundamental challenge in this dimension is the traceable transformation of knowledge about non-functional properties or requirements between the various phases and stakeholders.

The second dimension is the business/IT stack of a running system, including business and technical (service consumption, business process, service and infrastructure) layers. In a scenario where services are traded as economic goods, non-functional requirements results from specific customer needs. These requirements need to be negotiated with one or several service providers in order to check the conditions (including feasibility and price)

under which a requested service can be provisioned. In a second strand non-functional properties need to be consistently managed across the IT stack, where higher level requirements are systematically translated towards lower levels and lower level capabilities are aggregated to higher level ones. Finally, monitoring techniques need to be put in place on all layers in order assure that a running system provides sufficient quality on all layers and in all aspects. SLAs (service level agreements) seem to be a well-suited paradigm according to which services and their non-functional properties can be consistently managed across business/IT stack.

Non-functional properties cover a wide range of categories, such as security, performance, availability, manageability etc. While there are many point solutions to these the main challenge arises from interdependencies between the various categories. These interdependencies need to be made explicit in any comprehensive management approach to non-functional properties. Another major challenge is the dependency of non-functional properties from the functional behaviour which includes both, the usage behaviour but also the process logic within business processes and services.

A holistic approach for the management of non-functional properties is likely to include the following techniques:

- in order to allow for specification of non-functional properties at various stages of a system lifecycle and mapping between them several techniques can be adopted including model-driven development and model-driven architecture.
- Semantic technologies for bridging the perspectives of different stakeholders. Most likely, these need to be tightly integrated with the model-driven approaches.
- Aspect-oriented modelling techniques are among other approaches to be adopted in order to allow for non-intrusive management of non-functional properties according to a specific perspective (e.g. security management via a system administrator).

3.3.2 Service Non-Functional Properties

3.3.2.1 Security Non-Functional Properties

Before we focus on security architecture for NESSI framework, let us define several terms.

Security Goals

We will use term “Security Goal” to denominate a set of objectives that service oriented systems would typically have. These goals correspond to measures designed to address a particular aspect of the security. The security goals are: (1) Access Control, (2) Authentication, (3) Non-repudiation, (4) Confidentiality, (5) Integrity, (6) Availability, and (7) Privacy. These goals are related to various specialised security service (e.g. identification service), tackled later in this chapter.

Security Layers

Security goals, together with orthogonal “dimensions” such as trust, must be applied to a hierarchy of service oriented software and system components and their groupings (composed services), which we will refer to as security layers. This report will cover three security layers – the service infrastructure security layer, the individual service security layer, and the composed application (service ecosystem) security layer, which build on previous two layers to provide end-to-end security solution.

Security Views

A security view is a certain type of views that could be used in order to simplify representation of security goals across layers. The number of security views is not limited but we can distinguish: (1) the security operational (discovery, deployment, execution and control) view, (2) the end-user view (3) maintenance view (4) design view,

(5) analysis and optimisation view. Other views should be analysed later in order to synchronise them with NESSI framework.

Security Architecture

NESSI Security Architecture addresses trust, security and dependability concerns for the management, control, and use of service infrastructure, services and service coalitions, as well as data handled by services. The NESSI Security Architecture provides a comprehensive, top-down, end-to-end perspective of service security and can be applied to service infrastructure elements, services, and service coalitions in order to detect, predict, and correct security vulnerabilities.

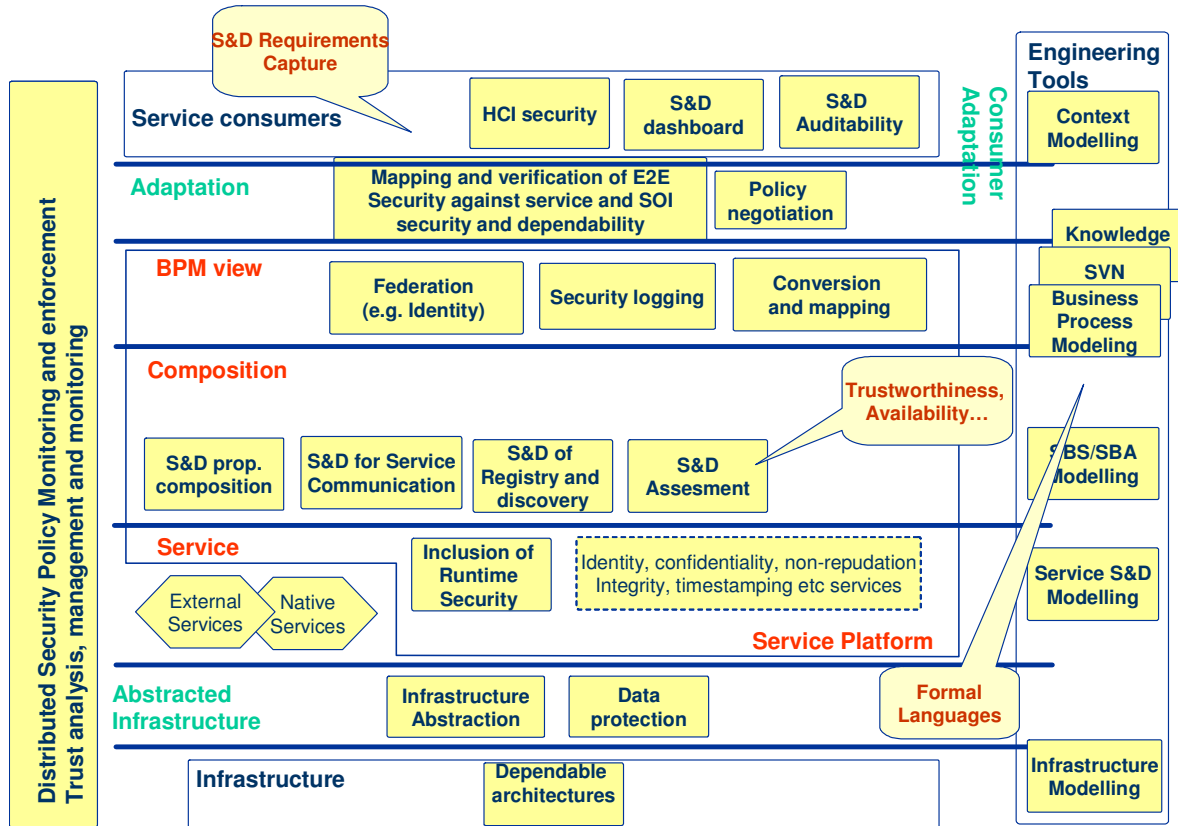


Figure 4. Trust, security and dependability topics depicted against NESSI service (layers) functional blocks

In addition, S&D properties and approach has to be analysed in relation to BPM, service middleware and infrastructures. Here, three possible models for distributed and dynamic TSD could be differentiated:

- Synchronous distributed (e.g. like previous remote procedure calls, RPC)
- Asynchronous publish/subscribe (e.g. federated security)
- Asynchronous message queue (e.g. peer to peer security)

There has been also a discussion between top-down and bottom-up approach to security engineering. Currently it takes more commonly bottom-up approach in the form of threat model—security policy—security mechanisms.

From infrastructures to user: message security

Security of service oriented communication infrastructures, for example, will largely depend on security implemented for communication between components through ESB (enterprise service bus). A set of TSD message schemes and a set of common TSD command messages have to be agreed in order to use shared infrastructure. If the

service bus is based upon publish/subscribe pattern, we could also consider TSD properties an additional criteria for pattern refinements (besides service list or service functionality based). For example, one could service composition layer subscribe only on “trusted” services or only services with availability level higher than X. For this purpose it is essential to come with widely accepted and shared TSD metrics (see also chapter 3.7).

From user to infrastructure: security requirements

On the other end of architecture, there are security requirements from end-users that have been treated at the application level or business process level. Up to now, BPM tools have been used at design time and TSD has been often treated “as it should be”, but the runtime environment and fine-grained security properties that are embedded in services impose now new approach to BPM TSD design and control. One of the main challenges will be how to capture and treat high level TSD requirements, how to use these during the service design and service oriented integration and how to link these to TSD of infrastructure elements.

In relation to the previously defined TSD views (see section on abbreviations and definitions), and more specifically operational view, we could make the following list of TSD related actions (this is a non-exhaustive list):

- 1) TSD operational view
 - a. Discovery:
 - i. convert implicit properties (e.g. trust) into explicit
 - ii. discover TSD patterns
 - iii. discover ingrained and embedded TSD properties
 - iv. secure repository and discovery services
 - b. Deployment:
 - i. Configure TSD
 - ii. Bind/integrate TSD properties of services
 - iii. Create interfaces for runtime monitoring
 - iv. Generate runtime policy
 - c. Execution and control
 - i. Service TSD state management
 - ii. Runtime monitoring
 - iii. Bind/integrate TSD properties in SOI
 - iv. Negotiation
 - v. Policy enforcement

A similar list of activities could be made for TSD design view or end-user view where many actions depend on the actual view owner (e.g. TSD requirements or trade-off decision making TSD versus cost)

3.3.2.2 Security services

Complementary and specialized providers are likely to offer security and dependability related services that will help enforce tailor made security policies. Service oriented architectures with associated integration middleware and infrastructure resources will potentially offer a new market niche for these mainly ‘non-functional’ security services. Among these services, federated secure identification, authentication and authorization services are needed to enable cross domain interoperability; they also need to support the required accounting capability. Besides certificate authority functionalities and strong authentication mechanisms, these services may support identified needs such as ad hoc provision of rights and delegation, timely revocation and enforcement of static and dynamic separation of duties. Secure Service Discovery may be deployed to protect service providers but also service consumers. Similarly, replication services will guarantee data availability and may provide fast disaster recovery facilities. Notarization services (escrow, time stamping ...) will provide certified evidence that contracts between peers are correctly executed. Trust services will automatically collect execution statistics and user ranking to support service choice based on their objective reputation. Analytics and fraud detection services will enact process optimization and early detection of threats.

It is important to notice that a number of security service will be offered by SOI (service oriented infrastructure) and that there could be some overlapping with security services at the application level (SOA security services).

Typically, SOI security services would include firewall services, intrusion protection, host protection, denial of service protection etc).

We expect a range of infrastructure security services that need to be available to higher level services. For example, in the same way directory and single sign on mechanisms simplify account management within enterprise IT architectures, there will clearly be a role for identity management services. The Liberty alliance is already providing necessary standardization for inter-domain identity management, but more may be needed to support identity services for fine grained software components, services, devices, and perhaps other abstractions besides people. Moreover, grounding identity attestations will likely require new security services such as integrity guarantees. Cryptographic techniques are likely to be used for these attestations and this in turns makes cryptographic key management a necessary component or service.

The sharing of storage and network infrastructure implies research into mechanisms (encryption and/or policy enforcement perhaps provided using virtualization as well as traditional storage and network components) to control this sharing. Moreover, new work is required to understand what service abstractions are needed to enable trusted deployment and operation in these shared contexts.

In FP6 Akogrimo project, authentication, authorisation, accounting and charging functions are based in network middleware (NM) layer. In this project, for example, application service layer components are responsible for SLA negotiations, while SLA compliance monitoring is done at infrastructure service layer.

In FP6 GST project [XXX] security architecture was specially designed for telematics (e.g. car to car) applications and services. Their layered security model permits both end-to-end (E2E) and point-to-point (P2P) security mode, as well as any combination of these two. This proves to be useful in cases when an intermediate node has to re-route service. We could generalise this approach for specific situations and operations like policy negotiation, conversion and any other operation that has to be performed between two end points.

Privacy and Identification related services

User identification includes information used for authentication and authorisation services. In some cases it may be necessary to protect the real world identity of the individual for privacy and provide pseudonymous or anonymous identity.

The aim of user authentication is to ensure that access to the service is only granted to individuals or pseudonyms whose registration information has been validated. Authorization refers to the granting of permission to a user to access a service or service based system. Unlike authentication, authorization may utilize software-based access control mechanisms operating at a service or data level.

Message security services

Message security is about providing message confidentiality, integrity, non-repudiation, and exchanging security credentials between web service client and web service. Some of the related standards are WS security (OASIS), XML Encryption and XML signature (W3C recommendations).

Confidentiality Services

Closely related to the previous class of services are confidentiality services that provide the means by which sensitive information held in services is prevented from being disclosed to individuals not authorized to see it. The security services that support confidentiality and privacy are mainly predicated upon effective access control functions and consequently are the related to those for authentication. Capturing TSD requirements for these services will likely be part of the same process.

Integrity services

The aim of a service integrity service is to provide evidence that the service used in business process has not been modified, either deliberately or accidentally.

Non-Repudiation services

Non-repudiation services are providing proofs on different events such as non-repudiation of delivery (providing proof that the intended recipient received the message and recognized the content), non-repudiation of origin (providing proof that the originator created and sent the message) etc.

3.3.2.3 Secure (and dependable) services

In addition to the individual security technologies embedded in the architecture, other crucial elements are necessary for securing the architecture; namely security management practices and a system design in which appropriate and necessary separations are part of its structural properties.

Availability and information security properties in IT architectures must also scale, as architectures scale with the volume of operations or transactions they must support for the upper service layers. The nature of the architecture will also condition the security requirements on the architecture, for instance in the case of an architecture that is deployed over different distant locations in contrast to an architecture that is confined to a single building. Securing the architecture will therefore be a process (as is the case of a proper security strategy) that needs to encompass requirements from various types of architecture properties.

Security policy tends to be stated in natural language, and at a high/business level. The problem of how to translate these policies into the way security is managed at the application and infrastructure level has been around for some time. Currently organizations resolve this much more through people and process controls, rather than technology. Separating the problem into three layers, as has been done by this paper has not removed this problem, although by introducing service notions, and focusing on automation we provide a narrower context in which to study this problem.

Service availability is achieved with an architecture that is resilient to system failures (whether physical or logical).

When it comes to resilience and dependability of services and service oriented infrastructures (SOI), the separation within research projects is not always clear as problems and solutions often overlap. We could use definition which describes Dependability as a system property, Survivability as a system capability and Resilience as a “system quality” which embraces both dependability and survivability as it captures the property and capacity of a system to (ideally) autonomously and gracefully tackle, adapt, response, recover, self-heal, reconfigure, accommodate & tolerate upsets/disruptions/failures/attacks.

In the network of excellence Resist, the first dependability issue that was identified for SOA applications is to improve the reliability of the runtime support of the Web Services, i.e., the platform on which the web service is executed, while the second dependability issue is to tackle the problem at the level of actual SOA concepts, i.e., all protocols and software components used to interact with Web Services. Up until recently, increased system performance or Quality of Service (QoS) has meant less security. Notwithstanding, it is possible to build applications that gather the best of both worlds: high resilience at the level of arbitrary failure systems and high performance at the level of controlled failure systems. Distributed software paradigms, such as service orientation, have more places where faults can take place, but also have more potential for fault tolerance.

Through the innovative concept of architectural hybridization, several components with the capacity to tolerate vulnerabilities, attacks and intrusions of different kinds and levels of severity are simultaneously supported. For example, part of the system might be assumed to be subject to malicious attacks, whereas other parts are specifically designed in a different manner, to resist different sets of attacks. These hybrid failure assumptions are, in fact, enforced in specific parts of the architecture, by system component construction, and are thus substantiated. In particular, trusting an architectural component doesn't mean making possibly naive assumptions about what a hacker can or can't do to that component. Instead, the component is specifically constructed to resist a well-defined set of attacks.

Resilience engineering in service architectures should rely on understanding safety and why accidents happen, of which two significant insights became perceptible:

- Adverse events are more often due to an unfortunate combination of a number of conditions, than to the failure of a single function or component;
- Failures are the reverse side of successes, meaning if one doesn't hold the other will, undesirably, take its place.

Both of these have their origin in performance variability and complement each other. The failure mechanisms behind adverse events are simply not known beforehand, otherwise damage would have been avoided (hindsight approach). If we think of resilience as the ability to recover from or adjust easily to misfortune or change, as well as the capacity of systems to anticipate and adapt to the potential for surprise and failure, it becomes clear this notion had gradually emerged as the logical way to overcome the limitations of existing approaches of risk assessment and system safety. Resilience Engineering in service architectures is a paradigm that focuses on how to help people cope with complexity of these architectures under pressure to achieve success, strongly contrasting with what is typical today, i.e., a paradigm of tabulating an error as if it were a thing, followed by interventions to reduce this count. One measure of resilience is therefore the ability to create foresight in order to anticipate the changing shape of risk before failure and harm occurs.

Since the service oriented infrastructure is going to be owned and managed by different stakeholder, each of its constituents could interact in many different and, sometimes, unpredictable ways, in a major extent due to the concurrency existing between their functions and the intrinsic and surrounding events of the system. Modelling the accidents occurring in these systems, in a way to understand their causes and gather knowledge to be further applied in a retrospective approach in order to quickly react and avoid falling in the same consequences, is not enough if we seek to provide the system with a much powerful capacity, which is, to adapt itself in face of major stress or misfortune. Therefore, we shall also be able to pro-actively perform some measure of risk assessment for service oriented infrastructures in a way to predict and avoid accidents from happening, or at least to minimize their occurrence as much as possible.

In the European project Deserec for example, a combination of three dimensions has been selected to deal with resilience in critical infrastructures: modelling and simulation approach, detection mechanisms and response including reconfiguration. A similar combination could be applied to resilience in service infrastructures. Runtime reconfiguration is also considered in FP6 project Serenity. In the IRRIS project, on the other hand, the focus is on middleware components that would help to increase dependability, survivability and resilience of infrastructures. While resilience in service oriented infrastructures has been investigated in the above mentioned projects, having ICT infrastructures as a part of critical infrastructures, resilience engineering for services has not been sufficiently addressed yet. The participants of the network of excellence Resist are working currently with partners from the ESFORS project on the common position paper that targets this issue. Resilient services should meet 4 criteria: availability (Is this service running right now?), reliability (How long can it run without failure?) safety (What will happen if the service fails?) and maintainability (How easy is to fix the service?). One of the important issues will be classification of faults in services and service oriented infrastructures (transient faults, intermittent faults, permanent faults) which lead to errors and failures in service oriented software systems. Equally important issue is the business process resilience that relates to a group of services, and where design issues have to consider trade off between points of failures and efficiency. Service and process resilience is also considered in HIDENETS in analysis of end-to-end resilience solutions for distributed applications and mobility-aware services in ubiquitous communication scenarios, such as ad-hoc car-to-car communication with infrastructure service support. The use of off-the-shelf services and communication links will dramatically decrease the costs of market entry and hence make such ubiquitous scenarios commercially feasible, but, as HIDENETS correctly mentions, if services and other software components are unreliable, even if we have reliable service oriented infrastructures, the end-to-end system-level resilience solutions will not be satisfactory.

3.3.2.4 Performance Non-Functional Properties

3.3.2.5 Behavioural Non-Functional Properties

3.3.2.6 Deployment Non-Functional Properties

3.3.2.7 Business Non-Functional Properties

3.3.3 Infrastructure Non-Functional Properties

In the following we list the main challenges regarding non-functional issues in service-oriented infrastructures⁵⁵.

3.3.3.1 Dependability

In today's systems dependability requirements are typically satisfied with a static solution and most often only at the level of infrastructure resources (e.g. by assigning a number of redundant nodes or network links to a particular activity). The dependability characteristics of the solution are not directly linked to application (or business process) level requirements and consequently the economic costs and benefits of a particular approach to providing dependability are hard to quantify.

Future systems need dependability levels appropriate to the needs of individual customers. Consequently, the most important research challenge in this area is the automatic derivation of hardware configuration from customer requirements and the aggregation of the resulting costs so that different approaches to providing dependability can be compared.

Another important challenge is about resilience. This implies that node failures (or withdrawal of nodes/sites, congestion or reconfiguration of network connectivity etc) have minimal impact on the overall performance.

Automated system management based on the so-called self-* properties (e.g. self-configuring, self-healing, self-tuning) aim at ensuring maximum service availability and performance when some components of the infrastructure or application fail or become disconnected (the normal state of any large Grid deployment). Exploiting both physical and virtual machine based technologies for supporting these properties is required. Flexible monitoring across multiple resource types should be used to respond rapidly to failures or even to identify impending failures before they happen. Once identified, mechanisms for automated fault mitigation – making good use of resources that are still functioning correctly – and recovery are needed.

Special attention must be paid to scenarios where resources are provided by mobile devices. These are inherently less dependable components as a result of restricted or intermittent network connectivity and limitations associated with battery power. Predicting the behaviour of mobile nodes could be supported by the ability to monitor connectivity and other properties of mobile nodes in a consistent way.

3.3.3.2 Security

At the moment, Grid security mechanisms target traditional infrastructure resources (e.g. compute nodes, files, devices, .etc.) in a rather static way and do not consider the overlaying organizational structure. Enterprise security mechanisms are focused on organisational and user roles but make (implicit or explicit) assumptions about the underlying infrastructure topology which is subject to change in grid environments.

Future systems need to provide security mechanisms for possibly virtualised infrastructure resources (e.g. a virtual node or a distributed grid job) with a granularity and flexibility that allows the reflection of higher-level security zones such as a process group, a business process or organizational entities including departments, customers and

⁵⁵ From NESSI-Grid

partners. Consequently, Future systems require advanced security models and policies that can accommodate variability in both organisational structures and infrastructure topologies. This is most important for hosting scenarios, where systems process applications on behalf of competitors within a single operational domain. In this context, the ability to achieve complete isolation at any level (application to infrastructure), wherever required, is necessary. The dynamic nature of Grids means that the separation of resources into clear security layers is a major challenge. Possible conflicts between policies relating to local and remote resources albeit within a single logical layer will need to be addressed.

The security properties and performance experienced by a customer should be independent of any other services the provider may be offering to other customers. The scale and the scope of Grids necessitate security models that can provide certain security assurances even in the presence of adversaries. They should be reactive and protective in nature yet adaptive to the situations where menaces are unavoidable. Development of low-level security mechanisms at the infrastructure level could be an effective way of addressing external threats (e.g. from the Internet). Such deeply integrated mechanisms may support traditional techniques such as network-level firewalls or might even make them obsolete.

Access control in dynamic virtual organisations and virtual hosting environments is an important issue. Considerable work has been done on authentication mechanisms for Grids. Effective authentication solutions are now available for Grids. However, little research has been done on improving authorisation mechanisms for dynamic virtual organisations, i.e. where users and resources come and go dynamically. Most of the existing mechanisms can be classified as static and/or coarse grained. One promising approach for improving authorisation in virtual organisations is usage control. Usage control provides a model for fine grained and continuous control in Grids. The ability to check compatibility between user and provider usage control policies is required in order to support resource brokering.

Data security is an important element of any Grid service. It may make use of encrypted data handling and require scalable key management solutions for access control. Security features are required even for the protection of non-confidential data. Such data still requires its integrity to be maintained.

Research is also needed to develop forensics models for Grids to identify the causes of failures. This aspect of security modelling is important not only for the determination of the losses and liabilities but also for the determination of the preventive steps to be taken to avoid the similar breaches in the future.

Business applications require quantitative measures of the quality of protection. It is important to develop suitable metrics so that quality of security services (QoSS) can be included in SLAs.

Reputation and trust are important issues in any scenario that involves more than one administrative domain. Dynamic, lightweight interactions will need to be based on trust relationships, possibly supported by reputation rather than complex contracts. Mechanisms for distinguishing between providers on the basis of the quality of their delivery are required. These could be based on the shared experience of others as a means of establishing reputation at least for an unknown provider. Techniques for quantifying and managing the risk associated with dealing with unknown entities represent an important challenge. Solving this could allow appropriate security mechanisms to be selected at runtime for each interaction, balancing the risk associated with low levels of security against the performance overhead associated with highly secure operation.

3.3.3.3 Performance

Virtualisation involves hiding the differences between heterogeneous resources, across different resource types, between vendors and across time. It allows decoupling of computing hardware from applications and management systems. Hardware and software can then follow independent evolution paths since changes to one can have minimal effect on the other. The ability to rapidly configure and reconfigure virtual computing resources can be used to support a flexible resource pool for a wide range of applications. There is currently still a performance penalty associated with the use of virtualised computing resources although this is reducing as the technology improves. A more significant issue is that use of a virtualised infrastructure increases management complexity because the way applications are mapped to hardware can now be variable. This makes it much more difficult to reliably predict

specific performance metrics of the system in advance. This results in unpredictable costs for a given performance level, as the resources required to meet a SLA can only be estimated very roughly.

Scheduling in current Grid systems is typically carried out for a number of different kinds of resource (e.g. processing, storage, network transport) with a low degree of coordination between them. The consequence is that decisions made by one scheduler may conflict with the decisions of another. An integrated approach which enables coordinated scheduling would reduce execution time, balance workload more efficiently across available resources and provide a wider range of management options in case of failure or degradation of performance. Dynamic adaptation and scalable scheduling are particularly relevant to several of the more complex enterprise Grid scenarios.

Performance prediction is necessary in order to apply advanced scheduling techniques. This needs to take into account the issue of predicting the performance of the overall system based on the characteristics of its constituent parts and also where resources are shared by many independent applications.

Interactions between applications sharing the same resources are unpredictable in general and in a virtualised environment it will be infeasible to test all possible combinations in advance. Support for predictable isolation at the resource level is therefore required to achieve predictable performance. Combining monitoring with performance prediction and using the results to determine (re-)scheduling decisions will allow dynamic changes in workflows (or business processes) to be supported.

3.3.3.4 Interoperability

Today, organisations are adopting Grid solutions and achieving significant benefits even without having all the standards in place for interoperability. This is particularly true for organisations running a single version of one of the popular grid middleware software products. As Grid solutions become more widely adopted, the need for interoperability and standards increases. Interoperability becomes more critical as organisations connect Grids within their organisations or with other organisations that utilise different Grid middleware.

Current Grids offer restricted interoperability for two main reasons. Many applications, particularly business applications, will not run on a Grid without modification. Even if the executable software components are compatible with Grid (e.g. batch jobs), schedulers generally require an associated description of each component to be provided so that each component can be run on a compatible computer with the right parameters. Each scheduler typically has its own format for this description. Second, there are several different Grid middleware systems (and research infrastructures) which cannot easily be combined within a single Grid as they are based on different standards or have inconsistent behaviour.

Dynamic reconfiguration is another area where the heterogeneity of current systems limits interoperability. In the context of live migration of virtual machines, for example, the challenges include migration between subnets, between hosts that do not share access to common storage, and between hosts with different configurations (e.g. configuration of virtual switches).

Future systems should address the prospects running business applications on a Grid infrastructure without modification. In addition, development and adoption of open standards that support the interoperability of Grids based on different middleware is required. This is especially important for the hierarchical and extended enterprise scenarios where independently developed Grids based on different middleware may need to be integrated.

A particular challenge associated with interoperability is the need to go beyond harmonisation of message exchanges, which should be relatively straightforward. Mediation between different semantic models for non-functional aspects such as security, dependability, management information (e.g. monitoring data and policies) and performance is essential if the behaviour of the combined Grid is to be predictable and acceptable.

Open standards will be an important means of achieving interoperability between different Grids. There is already considerable work underway in a number of standards development organisations aimed at solving technical problems and defining standards for Grid interoperability. This needs to be better coordinated and supported by robust test methodologies so that users can have confidence that interoperability between different Grids across

organisations can be achieved. Reference implementations will also have a valuable role to play in promoting the use of open standards and interoperable solutions.

3.3.3.5 Manageability

Management and system administration are still major cost factors in today's business systems. In particular, many management tasks are not yet extensively automated. Grid systems provide some support for automated management of infrastructure resources but typically follow a narrow perspective that simply covers the management of resources as assets rather than looking at how they fit into overall business solutions.

Infrastructure architectures and their management systems need to have the flexibility to incorporate new kinds of server and special-purpose devices alongside existing hardware.

Virtualisation technologies applied across many resource types can provide the abstractions and interfaces required for an integrated and automated approach to management. Realising this promise is a significant research challenge.

The traditional approach to systems management has relied on human intervention to manage complex situations. However, this is now becoming untenable in large enterprise deployments and will be even more so in future.

There is a need to monitor and control on timescales relevant to individual application sessions whose duration could be on the order of seconds. Control decisions, particularly in a virtualised infrastructure must be made more rapidly than manual systems can achieve. This may be particularly important in mobile scenarios where the environment of mobile nodes is subject to change on short timescales.

Some improvements are anticipated in management of a single administrative domain – lightweight mechanisms for collecting information about the state of the managed Grid and controlling it as an integrated infrastructure.

However, management information will not, in general, be completely shared across administrative boundaries. Dealing with this fact represents a major research challenge. There will be no central point where management decisions can be made with full knowledge of the system state. Automated and decentralised management decision-making is required to achieve the required speed of response, particularly in a large system. Decisions will need to be made based on incomplete information – whether driven by the need to respond quickly or because access is restricted. Linkage to SLAs is important so that the impact of a particular management decision on customers can be determined.

Tools for modelling and simulating large Grids are required in order to ease the development and evaluation of automated management mechanisms and policies.

Appropriate analysis and presentation tools are required to provide visibility of the business performance on appropriate timescales for human decision makers.

3.3.3.6 Governance

Enforcement of governance policies, whether defined within a business or imposed by regulation or legislation, is poorly supported in current systems and requires manual intervention by administrators. In particular, support for complex organisational structures (such as hierarchical enterprise) is almost non-existent.

An important research challenge is therefore the enforcement and tracking of business policies at the infrastructure level. To achieve this, techniques for logging, tracking, and auditing of resources are necessary. In addition, (automatic) translation of business policies to configuration of the infrastructure needs to be addressed. A third issue is on automated mechanisms for enforcement and conflict resolution.

These requirements are essentially the same as those for automating management driven by SLAs. The major difference here is the need to demonstrate compliance via audit trails. Robust audit trails in turn rely on secure,

reliable data management so that actions taken to comply with governance policies are recorded in a durable form and it can be demonstrated that records have not been tampered with.

3.3.3.7 Flexibility

Typically, enterprise IT systems are implemented to meet specific requirements with little attention paid to building in the flexibility to support unanticipated future changes. This is understandable if the focus is on satisfying known requirements at the lowest cost. The result is a close association of software with dedicated servers and network infrastructure. This leads to inefficiency across the IT estate and also means that modifying IT systems to reflect changing business circumstances are expensive and time consuming.

Future systems require 2 main characteristics: the automation of information flows and management processes from business process to infrastructure, and the dynamic reconfiguration of infrastructure resources. As management automation issues are discussed in other sections we focus on flexible resources here.

This requires research in dynamic allocation and release of resources. Migration of software between servers is also required to support load balancing and scaling, both horizontal (changing the size of a resource pool) and vertical (migration to a resource with different capacity). Business applications are often multi-tier (e.g. presentation, business logic, data) so migration is more complex than moving a self-contained software component, since relationships between different parts of the application need to be managed. Migration of an application or its data may be subject to licensing, regulatory or legal constraints which need to be enforced by any automated management system. Management of the scheduling and migration of applications within a flexible infrastructure is a major challenge. An application may be represented as an abstract workflow, independent of the details of the deployment environment. This is invariant to changes in the underlying infrastructure. Mapping of abstract workflow to a concrete workflow (configured on specific resources) will need to be done when the application is first deployed and subsequently whenever a change is required. Semantic descriptions of applications and infrastructure are expected to be important aspects of automating this mapping process.

There are particular challenges associated with large scale flexible infrastructures. Future systems may involve very large and geographically distributed populations of users. They will also support very large numbers of applications on shared resources. Users and applications will be added and removed on short timescales. The dynamic nature of the infrastructure will make it more difficult to keep track of the association between user sessions and physical infrastructure, with implications for fault handling and impact assessment.

Managing security in a dynamic environment is also a challenge. Mechanisms such as firewalls which rely on protecting a boundary between physical resources will not be appropriate. Isolation of individual application sessions from one another is required, even if the resources used change during the session. This includes management of dynamic VOs or VPNs where users or resources can join or leave on short timescales and security policies must be maintained.

4 Core Principles

[Work approaches (e.g. openness) and Non Functional properties (e.g. domain independence) which cross the whole framework.]

One of the core objectives of NESSI is to define a **pervasive(?)** open service framework, codenamed NEXOF, which is to be provisioned as a utility. The deployment model is dynamic, based on multiple vendors (mostly service providers) deploying their own instantiation of NEXOF, with full interoperability and connectivity. There will not be a single large instantiation of NEXOF. The NESSI deployment model is inspired by the telecommunication world: wired and wireless service providers deploy their own networks and compete with one another, but the experience of the subscriber is a seamless complete connectivity: any phone number can be reached, regardless of the provider. Similarly, a user must be able to identify, locate, configure, and use a service independently of the framework instantiation which supports it.

The above objectives call for a rigorous approach in the definition of NEXOF, which must exhibit consistence, with both the angle of coherence (reasonable or logical harmony between parts) and constancy (remaining the same despite change or variation in other things). Yet, evolvability, defined as the ability to respond to changing needs and to stay competitive, is an essential attribute for survivability. This creates a tension between static and dynamic approaches, between the need for well characterized and well defined behaviors and the ability to evolve and change on the other.

The objective of the SRA is to define the key research topics which correspond to gaps for building or exploiting NEXOF. This volume focuses on the technology foundations for the Service Economy, as such the results of this research are expected to enrich NEXOF or, at a minimum, interface with NEXOF.

Our approach is to define core principles and common structures as the basis to reach our (conflicting) objectives of consistence and adaptability.

The core principles⁵⁶ are pervasive and fundamental to the processes used in the creation of NEXOF, whereas requirements capture the expected behaviour of the resulting system. By their nature, requirements are very specific and far outnumber principles, but only principles describe the philosophy which guides the construction of the system and, as such, drive the consistence of the system.

Previously published NESSI documents, including the NESSI Vision and the NESSI SRA Volume 1 “Framing the future of the Service Oriented Economy”, identify some core principles for NESSI [*to be revisited/localized/focused for Vol. 2*]:

- **Trust** – this is a fundamental principle on which the whole NESSI approach is based.
- **Dependability** – as in current ICT practices, new services-based system will be implemented in such a way as to guarantee predictable functioning, and that their usage should not harm humans or the environment.
- **Comprehensive View** – regardless of the complexity of its technologies, from the end-user point of view NESSI technologies will be perceived as utilities (e.g. electricity) without requiring specific knowledge to use them effectively.
- **Open Standards** – this principle is at the core of the approach to wide adoption of NESSI technologies without creating barriers to business actors benefiting from NESSI results. It also represents an important contributor to the overall interoperability of service-based business applications.
- **Open Source** – fostering Open Source development not only allows a widely adopted NESSI reference implementation to be built, but also increases market competitiveness and transparency.
- **Quality** – The implementation of NESSI solutions as well as service-based systems will consider quality as a design-guide principle. At this level quality has to be considered as a twofold property: from one side it pertains to the quality of the delivered systems as such, from the other it pertains to all aspects related to Service Level Agreement (SLA) management (definition, implementation, and assurance). Indeed the adoption of a specific service by a specific consumer is not only guided by what the service does, but certainly also which are the SLAs at the basis to the service (costs, performance, availability, etc.).

⁵⁶ Core principles defined here are a subset of what the literature labels as principles. Core principles address structuring characteristics of what is being built, whereas the literature more widely defines principles as being invariant characteristics of the process and methodology used to build things and their outcome.

- **Connectivity** – According to the “Comprehensive View” above and considering the size of the environment, needs in terms of flexibility, scalability and potential coverage in terms of geographical area, the whole infrastructure can be seen as a growing network made of a number of self-consistent instances able to interoperate in order to provide services to the users.
- **Federation** – NESSI will not be built as a single instance or infrastructure. Multiple different instances of NESSI will be able to cooperate and collaborate as needed. In this context, scalability of NESSI-based solutions (from the very small to the very large) is a crucial property.

With respect to the above, specific architectural principles are:

- adaptability with respect to domain, technology and market segment;
- interoperability (as reinforced by Federation);
- soa principles (e.g. loosely coupling – ref. OASIS?);
- ...

Future drafts of this document should confirm the above, enhance the list, and fine-tune the definitions. This should be done keeping in mind the important difference between a principle and a requirement, as spelled out above

5 References

5.1 Volume 2a

[Aho06] Aho, E, Cornu, J. Georghiou, L. and Subirá, A (2006) Creating an Innovative Europe - Report of the Independent Expert Group on R&D and Innovation appointed following the Hampton Court Summit, January 2006, European Commission, Brussels, Belgium.

[Hamdani00] Hamdani, D. (2000), ‘Measuring novelty of innovation: evidence from the Canadian Services Innovation Survey’, paper presented at The Economics and Socio-Economic of Services: International Perspectives Conference, Lille, 22-23 June 2000.

[Kanerva06] Kanerva, M. Hollanders, H. and Arundel, A. (2006) 2006 Trendchart Report: Can We Measure and Compare Innovation in Services? European Trend Chart on Innovation, Commission of the European Communities, Brussels, 17.

[VanArk03] Van Ark, B. Broersma, L. and Den Hertog, P (2003) Service Innovation, Performance and Policy: A Review, Ministry of Economic Affairs, The Hague.

5.2 Volume 2b

[ABD98] S. Araki, A. Bilas, C. Dubnicki, J. Edler, K. Konishi, and J. Philbin. User-space communication: A quantitative study. In *Proc. of The 1998 Supercomputing Conference on High Performance Networking and Computing (SC97)*, Orlando, Florida, November 1998.

[Akal05] Fuat Akal, Can Türker, Hans-Jörg Schek, Yuri Breitbart, Torsten Grabs, Lourens Veen: Fine-Grained Replication and Scheduling with Freshness and Correctness Guarantees. VLDB 2005: 565-576

[Akman97] V. Akman, M. Surav: The use of situation theory in context modeling. Computational Intelligence 13, 3, 427–438, 1997.

- [Alonso04] G. Alonso, F. Casati, H. Kuno, V. Machiraju: Web Services Concepts, Architectures and Applications, in: Springer, 2004.
- [Alvisi01] L. Alvisi, T. C. Bressoud, A. El-Khashab, K. Marzullo, D. Zagorodnov. Wrapping Server-Side TCP to Mask Connection Failures. INFOCOM 2001. pp. 329-337.
- [Amir02] Yair Amir, Ciprian Tutu: From Total Order to Database Replication. ICDCS 2002.
- [Amza03] Cristiana Amza, Alan L. Cox, Willy Zwaenepoel: Distributed Versioning: Consistent Replication for Scaling Back-End Databases of Dynamic Content Web Sites. Middleware 2003: 282-304
- [Amza05] Cristiana Amza, Alan L. Cox, Willy Zwaenepoel: A Comparative Evaluation of Transparent Scaling Techniques for Dynamic Content Servers. ICDE 2005: 230-241
- [Bennani05] N. Bennani, M. and A. Menasce, D. 2005. Resource Allocation for Autonomic Data Centers using Analytic Performance Models. In ICAC'05. pp. 229-240.
- [Bertalanffy68] L. von Bertalanffy: General System theory: Foundations, Development, Applications, New York, 1968: George Braziller, revised edition 1976: ISBN 0-8076-0453-4
- [Bolch06] G. Bolch, S. Greiner, H. de Meer, and K. S. Trivedi. Queueing Networks and Markov Chains. Wiley-Interscience. 2006.
- [Breitbart99] Yuri Breitbart, Raghavan Komondoor, Rajeev Rastogi, S. Seshadri, Abraham Silberschatz: Update Propagation Protocols For Replicated Databases. SIGMOD Conference 1999: 97-108
- [Camarinha-Matos99] Camarinha-Matos, L.M., Afsarmanesh, H., "The Virtual Enterprise Concept", in Infrastructures for Virtual Enterprises – Networking Industrial Enterprises, Kluwer Academic Publishers, IFIP Vol. 153, September 1999.
- [Capra03] Capra, L.; Emmerich, W.; Mascolo, C. CARISMA: context-aware reflective middleware system for mobile applications. IEEE Transactions on Software Engineering. V. 29, N. 10, pp. 929 – 945. Oct. 2003.
- [Cardellini02] Cardellini, V., Casalicchio, E., Colajanni, M., and Yu, P. S. 2002. The state of the art in locally distributed Web-server systems. ACM Comput. Surv. 34, 2 (Jun. 2002), 263-311.
- [CCRP02] J. L. Crowley, J. Coutaz, G. Rey and P. Reignier: Perceptual Components for Context Aware Computing. International Conference on Ubiquitous Computing, Goteborg, Sweden, September 2002.
- [Chen01] Chen, X., Funk, J., Madnick, S., Wang, R.: Corporate Household Data: Research Directions, Proceedings of the Americas Conference on Information Systems, Boston, August 2001
- [Chen06] Jin Chen, Gokul Soundararajan and Cristiana Amza. Autonomic Provisioning of Backend Databases in Dynamic Content Web Servers, ICAC 2006
- [Cecchet03] Emmanuel Cecchet, Julie Marguerite, Willy Zwaenepoel: Partial Replication: Achieving Scalability in Redundant Arrays of Inexpensive Databases. OPODIS 2003: 58-70
- [Chtcherbina03] E. Chtcherbina, M. Franz: Peer-to-peer coordination framework (p2pc): Enabler of mobile ad-hoc networking for medicine, business, and entertainment. In Proceedings of International Conference on Advances in Infrastructure for Electronic Business, Education, Science, Medicine, and Mobile Technologies on the Internet, L'Aquila, Italy, 2003.

[CMD99] K. Cheverst, K. Mitchell, N. Davies: Design of an object model for a context sensitive tourist GUIDE. *Computers and Graphics*, Vol. 23, No. 6, pp. 883-891, 1999.

[Crowley02] J. L. Crowley: Context Aware Observation of Human Activities. IEEE International Conference on Multimedia and Expo, Lausanne, August 2002.

[Crowley03] J. L. Crowley: Context Driven Observation of Human Activity. European Symposium on Ambient Intelligence, Amsterdam, November 2003.

[Daudjee05] Khuzaima Daudjee, Kenneth Salem: A Pure Lazy Technique for Scalable Transaction Processing in Replicated Databases. *ICPADS (1) 2005*: 802-808

[Daudjee06] Khuzaima Daudjee, Kenneth Salem: Lazy Database Replication with Snapshot Isolation. *VLDB 2006*: 715-726

[DBL97] C. Dubnicki, A. Bilas, K. Li, and J. Philbin. Design and implementation of Virtual Memory-Mapped Communication on Myrinet. In *Proc. of the 11th International Parallel Processing Symposium (IPPS97)*, pages 388-396, April 1997.

[Dey00] Dey, A.K. Abowd, G.D. Towards a Better Understanding of Context and Context-Awareness. CHI 2000 Workshop on the What, Who, Where, When, and How of Context-Awareness (2000)

[DMTF-CIM 07] DMTF: Common Information model. URL: <http://www.dmtf.org/standards/cim/>

[DMTF-SVPC 07] DMTF: System Virtualization, Partitioning, and Clustering Working Group. August 2007; URL: <http://www.dmtf.org/about/committees/SVPCCharter.pdf>

[Elnikety06] S. Elnikety, S. Dropsho and W. Zwaenepoel. Predicting Replicated Database Speedup from Standalone Database Measurements: Single-Master vs. Multi-Master. Technical report. EPFL. Nov. 2006.

[ESKS06] S. Eichler, C. Schroth, T. Kosch, M. Strassberger: Strategies for Context-Adaptive Message Dissemination in Vehicular Ad Hoc Networks. Proceedings of the IEEE Second International Workshop on Vehicle-to-Vehicle Communications (V2VCOM), San Jose, CA, USA, July 2006.

[Fielding00] Fielding, T. R.: Architectural styles and the design of network-based software architectures, Ph.D. thesis, University of California, Irvine, 2000

[Forrester06] Web Services Specifications, Dec 2006/Jan 2007. www.forrester.com

[Frost98] Frost, P., Frost, S.: Component-based development for enterprise systems: applying the SELECT perspective, in: Cambridge University Press, 1998

[Gançarski07] Stéphane Gançarski, Hubert Naacke, Esther Pacitti, Patrick Valduriez: The leganet system: Freshness-aware transaction routing in a database cluster. *Inf. Syst.* 32(2): 320-343 (2007)

[Giunchiglia97] F. Giunchiglia: Contextual reasoning. *Epistemologica - Special Issue on I Linguaggi e le Macchine* 16 (1993), 345–364. Also IRST-Technical Report 9211-20, IRST, Trento, Italy, 1993.

[Gray96] Gray, J., Helland, P., O'Neil, P., and Shasha, D. 1996. The dangers of replication and a solution. *ACM SIGMOD* 1996. pp. 173-182.

[Gokhale04] A. S. Gokhale, B. Natarajan, D. C. Schmidt, J. K. Cross. Towards Real-Time Fault-Tolerant CORBA Middleware. *Cluster Computing* 7(4), pp. 331-346. 2004.

[Halpin01] T. A. Halpin: Information Modeling and Relational Databases: From Conceptual Analysis to Logical Design. Morgan Kaufman Publishers, San Francisco, 2001.

[HBS02] A. Held, S. Buchholz, A. Schill: Modeling of context information for pervasive computing applications. Proceedings of the SCI2002/ IAS 2002, 2002.

[HIR02] K. Henricksen, J. Indulska, A. Rakotonirainy: Modeling context information in pervasive computing systems. In LNCS 2414: Proceedings of the 1st International Conference on Pervasive Computing, Zurich, Switzerland, 2002.

[HIR03] K. Henricksen, J. Indulska, A. Rakotonirainy: Generating Context Management Infrastructure from High-Level Context Models. Proceedings of the 4th International Conference on Mobile Data Management, Australia, 2003.

[Hoegg06] Hoegg, R., Meckel, M., Stanoevska-Slabeva, K., Martignoni, K. (2006): Overview of business models for Web 2.0 communities, Proceedings of GeNeMe 2006, p. 23-37, Dresden, 2006

[Hof07] Hof, R.D., Mix, Match, And Mutate, Business Week Magazine, 2005, available online at: http://www.businessweek.com/@@76IH*ocQ34AvyQMA/magazine/content/05_30/b3944108_mz063.htm, last seen: February 23, 2007

[Holliday02] JoAnne Holliday, Divyakant Agrawal, Amr El Abbadi: Partial Database Replication using Epidemic Communication. ICDCS 2002: 485-493

[Holliday03] JoAnne Holliday, Robert C. Steinke, Divyakant Agrawal, Amr El Abbadi: Epidemic Algorithms for Replicated Databases. IEEE Trans. Knowl. Data Eng. 15(5): 1218-1238 (2003)

[Hu06] B. Hu: Contextual Computing. In: 1st International Symposium on Pervasive Computing and Applications, pp. 10-11, IEEE Press, 2006.

[IRRH03] J. Indulska, R. Robinson, A. Rakotonirainy, K. Henricksen: Experiences in using cc/pp in context-aware systems. In LNCS 2574: Proceedings of the 4th International Conference on Mobile Data Management, Australia, 2003.

[JBoss] <http://labs.jboss.com/projects/docs/>

[Jimenez02] R. Jiménez-Peris, M. Patiño-Martínez, G. Alonso. *An Algorithm for Non-Intrusive, Parallel Recovery of Replicated Data and its Correctness*. 21st IEEE Int. Conf. on Reliable Distributed Systems (SRDS 2002), pp. 150-159. Oct. 2002, Osaka, Japan.

[Jimenez03] R. Jiménez-Peris, M. Patiño-Martínez, G. Alonso, and B. Kemme. *Are Quorums an Alternative for Data Replication?* ACM Transactions on Database Systems (TODS), Vol. 28, N. 3, Sept. 2003, pp. 257-294, ACM Press.

[Jimenez07] Ricardo Jimenez-Peris, Marta Patino-Martinez, Damian Serrano, B. Kemme. Towards Large-Scale Replicated Databases. Workshop on Large-Scale Distributed Systems and Middleware (LADIS). Haifa (Israel). 19-21 March 2007.

[KAESS06] T. Kosch, C. Adler, S. Eichler, C. Schroth, M. Strassberger: The Scalability Problem of Vehicular Ad Hoc Networks and How to Solve it. IEEE Wireless Communications, Volume: 13, Number: 5; October 2006.

[Kemme00] Bettina Kemme, Gustavo Alonso. Don't Be Lazy, Be Consistent: Postgres-R, A New Way to Implement Database Replication. VLDB 2000. pp. 134-143.

[Kemme01] B. Kemme, A. Bartoli, and O. Babaoglu. Online Reconfiguration in Replicated Databases Based on Group Communication. In *Proc. of the Int. Conf. on Dependable Systems and Networks (DSN)*, 2001.

[Killijian00] Killijian, M.-O.; Fabre, J.C.. Implementing a reflective fault-tolerant CORBA system. The 19th IEEE Symposium on Reliable Distributed Systems (SRDS), pp. 154 – 163. 2000.

[KLS 06] R. Krummenacher, H. Lausen, T. Strang: On the Modeling of Context-Rules with WSML. In: *Proceedings, Workshop on Contexts & Ontologies: Theory, Practice and Applications at ECAI'06*, Riva del Garda, Italy, 2006.

[Kolbitsch 06] Kolbitsch, J., Maurer, H.: The Transformation of the Web: How Engineering Communities Shape the Information We Consume, *Journal of Universal Computer Science*, Vol. 12, No. 2, 2006

[Kon02] Kon, F., Costa, F., Blair, G., and Campbell, R. H. The case for reflective middleware. *Commun. ACM* 45, 6, pp. 33-38. June 2002.

[Kounev 2005] Samuel Kounev: Performance Engineering of Distributed Component-based Systems. PhD thesis, University of Darmstadt, May 2005

[Krummenacher07] R. Krummenacher, T. Strang,: Ontology-Based Context Modeling. In: *Proceedings, Context Awareness for Proactive Systems (CAPS 2007)*, Guildford, United Kingdom, 2007

[Liebermann00] H. Lieberman, T. Selker: Out of context: Computer systems that adapt to, and learn from, context. *IBM Systems Journal*, Vol. 39 (3&4), 2000.

[Lin05] Y. Lin, B. Kemme, M. Patiño-Martínez, R. Jiménez-Peris. Middleware based Data Replication providing Snapshot Isolation. *ACM SIGMOD Int. Conf. on Management of Data*. Baltimore, Maryland, USA. Jun. 2005.

[MacKenzie06] M. MacKenzie, K. Laskey, F. McCabe, P. F. Brown, R. Metz: OASIS - Reference Model for Service Oriented Architecture 1.0, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=soa-rm, 2006

[Malone 01] Malone, T.: “The Future of E-Business”, *Sloan Management Review*, Vol. 43, No. 1, p. 104, 2001

[Matick03] Matick, R. E. 2003. Comparison of analytic performance models using closed mean-value analysis versus open-queueing theory for estimating cycles per instruction of memory hierarchies. *IBM J. Res. Dev.* 47, 4 (Jul. 2003), 495-517.

[MPB07] M. Marazakis, V. Papaefstathiou, and A. Bilas. Optimization and Bottleneck Analysis of Network Block I/O in Commodity Storage Systems. In *Proc. of the 21st ACM International Conference on Supercomputing (ICS07)*, June 2007. To appear.

[McAfee05] Will Web Services Really Transform Collaboration, in: *MIT Sloan Management Review*, Vol.46, No.2, 2005

[McAfee07] A. McAfee: Enterprise 2.0 vs. SOA, in: *Harvard Business School Faculty Blog*, http://blog.hbs.edu/faculty/amcafee/index.php/faculty_amcafee_v3/enterprise_20_vs_soa/, last seen February 28, 2007

[McCarthy93] J. McCARTHY: Notes on formalizing contexts. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, San Mateo, California, USA, 1993.

[Merrick03] Merrick, P.: “webMethods CEO eyes Web services innovation - Semantic integration, SOA hold key to next-generation integration products”, Interview by Mark Jones, InfoWorld, 24 Nov 2003.

[Moran01] Thomas P. Moran, Paul Dourish Special Issue of Human-Computer Interaction, Volume 16, 2001

[MXP06] M. Marazakis, K. Xinidis, V. Papaefstathiou, and A. Bilas. Efficient Remote Block-level I/O over an RDMA-capable NIC. In *Proc. of the 20th ACM International Conference on Supercomputing (ICS06)*, June 2006.

[Narasimhan02] P. Narasimhan, L. E. Moser and P. M. Melliar-Smith, Eternal - A Component-based Framework for Transparent Fault-Tolerant CORBA Software Practice and Experience, Volume 32, Issue 8, pp. 771-788. July 2002.

[NESSI-Grid 07] NESSI-Grid: Grid Vision and Strategic Research Agenda. Version 2. October 2007. URL: <http://www.nessi-europe.com/Nessi/> -> About NESSI -> NESSI-Grid

[OCG-ITIL 07] Office of Governance Commerce: IT Infrastructure Library. URL: <http://www.itil.org>

[Oliveira06] A. Sousa, Afrânio Correia Jr., F. Moura, J. Pereira, Rui Oliveira: Evaluating Certification Protocols in the Partial Database State Machine. ARES 2006: 855-863

[OMG-MOF 07] Object Management group: Meta Object Facility, URL: <http://www.omg.org/mof/>

[OMG-UMLSPT 05] Object Management group: UML profile for Schedulability, Performance, and Time Specification. Version 1.1, January 2005; URL: <http://www.omg.org/technology/documents/formal/schedulability.htm>

[OMG-MARTE 07] Object Management group: Modeling and Analysis of Real-time and Embedded Systems. Beta 1 specification, August 2007; URL: <http://www.omg.org/cgi-bin/doc?ptc/2007-08-04>

[Oracle] Oracle9iAS. <http://www.oracle.com/technology/deploy/security/oracle9ias/index.html>

[Oreilly05] O’Reilly, T.: “What is Web 2.0”, <http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>

[Pacitti05] Esther Pacitti, Cédric Coulon, Patrick Valduriez, M. Tamer Özsu: Preventive Replication in a Database Cluster. *Distributed and Parallel Databases* 18(3): 223-251 (2005)

[Pape04] Cécile Le Pape, Stéphane Gançarski, Patrick Valduriez: Refresco: Improving Query Performance Through Freshness Control in a Database Cluster. *CoopIS/DOA/ODBASE* (1) 2004: 174-193

[Parnas 72] Parnas, D.L.: On the criteria to be used in decomposing systems into modules, in: *Commun. ACM*, Vol.15, No.2, ACM Press, 1972.

[Patino05] M. Patiño-Martinez, R. Jimenez-Peris, B. Kemme, G. Alonso. Consistent Database Replication at the Middleware Level. *ACM Transactions on Computer Systems (TOCS)*. n. 4, v. 23, pp. 375-423, Nov. 2005.

[Pedone03] Fernando Pedone, Rachid Guerraoui, André Schiper: The Database State Machine Approach. *Distributed and Parallel Databases* 14(1): 71-98 (2003)

[Perez06] F. Perez, J. Vuckovic, M. Patiño-Martínez, R. Jiménez-Peris. Highly Available Long Running Transactions and Activities for J2EE Applications. 26th IEEE Int. Conf. on Distributed Computing Systems (ICDCS), Lisboa, Portugal, July 2006.

[PKK07] Stavros Passas, George Kotsis, Sven Karlsson, and Angelos Bilas. Multiedge: An edge-based communication subsystem for scalable commodity servers. In *Proc. of the 21th International Parallel and Distributed Processing Symposium (IPDPS2007)*, March 2007.

[Plattner06] Christian Plattner, Gustavo Alonso, M. Tamer Özsu: DBFarm: A Scalable Cluster for Multiple Databases. *Middleware 2006*: 180-200

[Porter01] M. Porter, *Strategy and the Internet*, Harvard Business Review, 2001

[Ren03] J. Ren, D. E. Bakken, T. Courtney, M. Cukier, et al. AQUA: An Adaptive Architecture that Provides Dependable Distributed Objects. *IEEE Trans. Computers* 52(1): 31-50 (2003)

[ROC] Brown, A. A Recovery-Oriented Approach to Dependable Services: Repairing Past Errors With System-Wide Undo, UC Berkeley Computer Science Division Technical Report UCB//CSD-04-1304, December 2003.

[Rohm02] Uwe Röhm, Klemens Böhm, Hans-Jörg Schek, Heiko Schuldt: FAS - A Freshness-Sensitive Coordination Middleware for a Cluster of OLAP Components. *VLDB 2002*: 754-765

[Salas06] J. Salas, F. Perez, M. Patiño-Martínez, R. Jiménez-Peris. WS-Replication: A Framework for Highly Available Web Services. *ACM WWW Conf.*, Edinburgh, Scotland, May 2006.

[SAW94] B. N. Schilit, N.L. Adams, R. Want: Context-aware computing applications. In *IEEE Workshop on Mobile Computing Systems and Applications*, Santa Cruz, CA, US, 1994.

[SBG99] A. Schmidt. M. Beigl, H.-W. Gellersen: There is more to context than location. *Computers and Graphics*, Vol. 23, No.6, pp. 893-901, 1999.

[Schmidt06] A. Schmidt: Ontology-based user context management: The challenges of dynamics and Imperfection. *Proceedings of ODBASE 2006, On the Move Federated Conferences (OTM)*, Montpellier, France. LNCS, Springer, 2006.

[SHC00] K. Schwerdt, D. Hall, J. L. Crowley: Visual Recognition of Emotional States. *The Third International Conference on Multimodal Interfaces, ICMI-2000*, Beijing China, October 2000.

[Sheiko06] Dmitry Sheiko. *Aspect-oriented Software Development and PHP*, 2006.

[SLF03a] T. Strang, C. Linnhoff-Popien, K. Frank: CoOL: A Context Ontology Language to enable Contextual Interoperability. Springer LNCS 2893: *Proceedings of 4th IFIP WG 6.1 International Conference on Distributed Applications and Interoperable Systems (DAIS2003)*, November 2003.

[SLF03b] T. Strang and C. Linnhoff-Popien and K. Frank: Applications of a Context Ontology Language. *Proceedings of International Conference on Software, Telecommunications and Computer Networks (SoftCom2003)*, October 2003.

[Soundararajan06] Gokul Soundararajan, Cristiana Amza, Ashvin Goel. Database Replication Policies for Dynamic Content Applications. *ACM SIGOPS EuroSys 2006*

[Strang04] T. Strang, C. Linnhoff-Popien: A context modeling survey. *UbiComp 1st International Workshop on Advanced Context Modelling, Reasoning and Management*, pp. 34-41, Nottingham, 2004.

[TheEconomist05]: Mashing the Web, The Economist 376 (8444), Special section, p. 4, 2005

[W3C-SML 07] World Wide Web Consortium: Service modelling language. URL:
<http://www.w3.org/XML/SML/>

[WebLogic] WebLogic Clustering. <http://edocs.bea.com/wls/docs81/cluster/overview.html>

[Weill04] Weill, P., Ross, R.W.: IT Governance: How Top Performers Manage IT Decision Rights for Superior Results, Harvard Business School Press, May 2004

[Wu05] H. Wu, B. Kemme. Fault-tolerance for Stateful Application Servers in the Presence of Advanced Transactions Patterns. IEEE SRDS. pp. 95-108. 2005.

[Zhang05] Q. Zhang, A. Riska, W. Sun, E. Smirni, and G. Ciardo. Workload-aware load balancing for clustered web servers. IEEE Trans. Parallel Distributed System, 16, 2005.

6 Appendix A – SOA standards table

The following draft table is here provided as a reference to sketch the complexity and fragmentation of the standards world particularly when addressing SOA.

<u>Standard or Specification & Version</u>	<u>Source</u>	<u>Status</u>
WEB Services		
Core Web services messaging:		
SOAP 1.1	W3C	spec
SOAP Messages with Attachments	W3C	spec
SOAP 1.2 (Parts 0, 1, 2)	W3C	standard
Conforms to SOAP 1.2 Test Suite, 2003 06 24	W3C	standard
XOP (XML-binary Optimized Packaging)	W3C	standard
MTOM (SOAP Message Transfer Optimization Mechanism)	W3C	standard
RRSHB (Resource Representation SOAP Header Block)	W3C	standard
SOAP 1.1 Request Optional Response HTTP Binding	W3C	spec
SOAP 1.2 Message Normalization	W3C	spec
SOAP 1.2 Email Binding	W3C	spec
WSDL 1.1	W3C	spec
WSDL 2.0 (Parts 0, 1, 2)	W3C	in process
WSDL 2.0 SOAP 1.1 Binding	W3C	in process
WSDL 2.0 RDF Mapping	W3C	in process
WS-I Basic Profile 1.0	WS-I	standard
WS-I Basic Profile 1.1	WS-I	standard
Simple SOAP Binding Profile 1.0	WS-I	standard
WS-I Attachments Profile 1.0	WS-I	standard
Fast Infoset	ITU/ISO	standard
Registry, metadata, and resource management:		
UDDI 2.0	OASIS	standard

CONFIDENTIAL DRAFT – FOR DISTRIBUTION TO NESSI MEMBERS ONLY

UDDI 3.0.2	OASIS	standard
Schema Centric XML Canonicalization 1.0	OASIS	spec
WS Policy 1.5 - Framework	W3C	in process
WS Policy 1.5 - Attachment	W3C	in process
Semantic Annotations for WSDL	W3C	in process
WS-MetadataExchange	multiple	vendor spec
WS Discovery	multiple	vendor spec
WS-Enumeration	W3C	in process
WS-Transfer	W3C	in process
WS-ResourceFramework 1.2 (WSRF)	OASIS	standard

Security:

WS-Security: SOAP Message Security 1.1	OASIS	standard
WS-Security: Username Token Profile 1.1	OASIS	standard
WS-Security: SAML Token Profile 1.1	OASIS	standard
WS-Security: X.509 Token Profile 1.1	OASIS	standard
WS-Security: Kerberos Token Profile 1.1	OASIS	standard
WS-Security: REL Token Profile 1.1	OASIS	standard
WS-Security: SOAP with Attachments Profile 1.1	OASIS	standard
WS-SecureConversation 1.3	OASIS	in process
WS-Trust 1.3	OASIS	in process
WS-SecurityPolicy 1.2	OASIS	in process
XACML 1.1	OASIS	standard
XACML 2.0	OASIS	standard
SPML 2 (Service Provisioning Markup Language)	OASIS	standard
SAML 1.0	OASIS	standard
SAML 1.1	OASIS	standard
SAML 2.0	OASIS	standard
WS Federation	multiple	vendor spec
WS-I Basic Security Profile 1.0	WS-I	in process
WS-I Basic Security - Username Token Profile	WS-I	in process
WS-I Basic Security - X.509 Cert Token Profile	WS-I	in process
WS-I Basic Security - SAML Token Profile	WS-I	in process
WS-I Basic Security - Kerberos Token Profile	WS-I	in process
WS-I Basic Security - REL Token Profile	WS-I	in process
WS-I Reliable Secure Profile	WS-I	in process

Delivery control and process flow:

WS-Addressing 1.0 - Core	W3C	standard
WS-Addressing 1.0 - SOAP Binding	W3C	standard
WS-Addressing 1.0 - WSDL Binding	W3C	in process
WS-Reliability 1.1	OASIS	standard
WS-ReliableMessaging 1.1	OASIS	in process
WS-RM Policy 1.1	OASIS	in process
WS-BPEL 2.0	OASIS	in process
WS-BPEL Extension for People	multiple	vendor spec
WS Choreography Description Language 1.0	W3C	in process
XPDL 2.0 (XML Process Definition Language)	WfMC	standard
Wf-XML 2.0	WfMC	in process
ASAP 1.0 (Asynchronous Service Access Protocol)	OASIS	in process

CONFIDENTIAL DRAFT – FOR DISTRIBUTION TO NESSI MEMBERS ONLY

WS Eventing	W3C	in process
WS Notification 1.3	OASIS	in process
Management:		
WSDM-MOWS 1.0	OASIS	standard
WSDM-MUWS 1.0 (Parts 1, 2)	OASIS	standard
WSDM-MOWS 1.1	OASIS	standard
WSDM-MUWS 1.1 (Parts 1, 2)	OASIS	standard
WS Management	DMTF	in process
Transactions, frameworks, and packaging:		
Liberty ID-WSF 1.1 (Web Services Framework)	Liberty	standard
Liberty ID-WSF 2.0 (Web Services Framework)	Liberty	in process
Liberty ID-WSF Data Services Template 2.0	Liberty	standard
Liberty ID-SIS 1.0	Liberty	in process
WS-CAF (Composite Application Framework)	OASIS	in process
WS-Coordination 1.1	OASIS	in process
WS-AtomicTransaction 1.1	OASIS	in process
WS-BusinessActivity 1.1	OASIS	in process
Service Component Architecture	OSOA	vendor spec
Service Data Objects	OSOA	vendor spec
Java Business Integration	JCP	standard
Modeling and development		
Software Services Profile and Metamodel	OMG	in process
UML Profile and Metamodel for Services (UPMS)	OMG	in process
UML Profile for Modelling Quality of Services and Fault Tolerance Characteristics and Mechanisms	OMG	standard
BPMN	OMG	standard
Standard services and miscellaneous:		
Digital Signature Services	OASIS	in process
Translation Web Services	OASIS	in process
WSRP 1.0	OASIS	standard
WSRP 2.0	OASIS	in process
Semantic Execution Environment	OASIS	in process
WS Quality Model (WSQM)	OASIS	in process
Core supporting specs:		
XML 1.0	W3C	standard
XML 1.1	W3C	standard
XML Schema 1.0 (Parts 1, 2)	W3C	standard
XML Schema 1.1 (Parts 1, 2)	W3C	in process
XML Schema Patterns for Common Data Structures 1.0	W3C	in process
XSLT 1.0	W3C	standard
XSLT 2.0	W3C	in process
XPath 1.0	W3C	standard
XPath 2.0	W3C	in process
XQuery 1.0	W3C	in process
XProc	W3C	in process

CONFIDENTIAL DRAFT – FOR DISTRIBUTION TO NESSI MEMBERS ONLY

XML Signature (standalone usage)	W3C	standard
XML Encryption (standalone usage)	W3C	standard
ebXML:		
ebXML BPSS 2.0.3 (ebBP)	OASIS	in process
ebXML CPPA 3.x	OASIS	in process
ebXML conformance and interoperability tests	OASIS	in process
ebXML Messaging Services 3.0 (ebMS)	OASIS	in process
ebXML Registry Information Model 3.0	OASIS	standard
ebXML Registry Services 3.0	OASIS	standard
ebSOA	OASIS	in process
Other specifications & standards:		
Atom Syndication Format	IETF	in process
Atom Publishing Protocol	IETF	in process
WADL	TBD	in process
XACML 1.0	OASIS	standard
UBL V2.0		
OpenJXP		
Java	JCP	Various
Http, https		
BTP (Business Transaction Protocol)	OASIS	
OWL (Ontology)		
OWLs (Semantics)		
Semantic Annotation for WSDL (SAWSDL)		
WS-MO		
WS-ML		
BPEL (Service orchestration)		
WS-Metadata Exchange		
WS-Federation		
SAML2		
WS-Policy		
WS-Addressing		
WS-Reliable Exchange		
WS-I BSP 2.0 (Interoperability)		
SOAP 2.0		
WSDL 2.0		
OSGI		
JSR88		
Enterprise service bus		
WS-Context		
WS-Coordination		
WSS	OASIS	

