Using Advanced Integration Concepts for Trans-sector Innovation – View and Status

H.J.M. (Harrie) Bastiaansen[1] and N.H.G. Baken[2]
[1]TNO Information and Communication Technology, the Netherlands
[2]KPN and Delft University of Technology, the Netherlands
e-mail:

Abstract - Through history, society has evolved from self-sustaining groups into a set of independent sectors. Today, ICT is pervasive in all sectors. Combine this with the trend that these sectors can no longer exist and innovate independently and we realize that a trans-sector approach is required. Information technology has an important role to play in trans-sector innovation. It literary and virtually connects the sectors, makes them interact and enables a wide spectrum of innovative trans-sector opportunities. But the decomposed society also raises a tremendous integration challenge in bringing together the different IT-environments of the various sectors. This puts a severe restriction on the trans-sector innovation power. Minimization of the integration complexity and effort becomes a key to success in trans-sector innovation. In this paper, we present a view on coping with the trans-sector integration complexity. This view builds upon deploying recent, state-of-the-art, developments in integration technology. As such, developments in model driven architectures, service oriented architectures, semantic web services and ontologies are considered. Along with the view, we identify various short-comings in the current status of these new integration technologies and present the challenges for bridging the gap between the view and reality.

Keywords: trans-sector integration, model driven architectures, service oriented architectures, semantic web services, ontologies.

Introduction — the Historic Perspective

Nomadic families started to settle down during the aftermath of the last ice-age and became colonists. The first villages emerged, hosting several families. During the tremendously long period before colonization, nomadic tribes were entirely self responsible, almost completely self sustaining cohesive clusters performing all necessary tasks in order to survive as a group. One may consider a tribe to have a collective nerve system guarding and controlling the prosperity and well-being of this relatively small "network".

The early colonist era was characterized by the introduction of agriculture and cattle breeding laying the foundation for the first wave [1], the agricultural revolution. People commenced, for sake of efficiency, to divide tasks among each other. They learned to specialize in a set of gradually and logically expanding specific tasks, such as farming, construction (houses, roads), mining, warfare, etc. Each requiring a rather unique knowledge and skill set.

The outsourcing of the tasks is based on a trust relation and supported by transactions, where people allowed other people to take over vital tasks they used to perform themselves. This transition initialized an important trend: progressing ***functional decomposition*** in economy and society. This enlarged efficiency and thereby the scale of economy and society, growing from local all the way to global. Direct implications are increasing complexity of economic and societal processes accompanied by a strong growth of the vocabulary of our language and the discovery of counting and abstract numbers providing the basis for the skill of writing.

This progress naturally invoked the secondary and tertiary wave (Fig. 1), i.e., the industrial and information (technology) revolution or rather the ICT-revolution, respectively. During all phases the functional decomposition progressed, and new entities, sectors, branches and companies emerged, having their own "partial rationale", e.g. healthcare, energy, construction, telecommunications, finance, industry, education, etc.

Summarizing, economic values gained importance and herewith the former tribal holistic nerve system gradually started to fade or even disappear to make place for partial rationalities of the emerging functional entities, such as sectors. The tight small-scale network of the tribe transformed in steps into a large scale network, comprising a set of sectors that in turn have become internationally organized [2]. Too strongly, they navigate on their own partial rationale for short term benefits. Presently, they resemble more the collection of opportunistically interacting isolated nodes instead of a strongly interactive network.

Have the individual sectors evolved too far? Is trans-sector innovation still

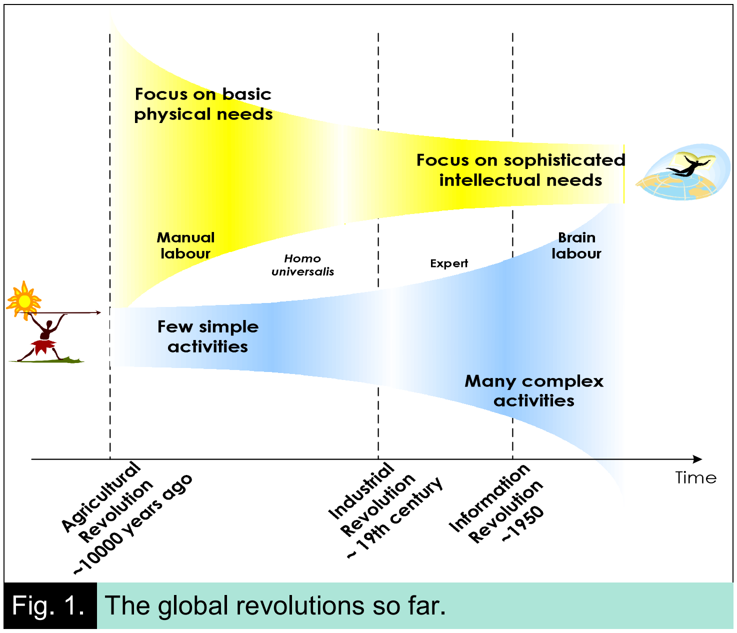
Focus on basic physical needs
Focus on sophisticated intellectual needs
Manual labour
Homo universalis
Expert
Brain labour
Few simple activities
Many complex activities
Time
Agricultural Revolution ~10000 years ago
Industrial Revolution ~ 19th century
Information Revolution ~1950

Fig. 1. The global revolutions so far.

feasible without excessive amount of effort? The answer to these questions contains organizational, commercial, cultural and technological aspects. In this paper we focus on the latter. We present a view on how integration technology may help in a distributed society and play a vital role as enabler for trans-sector innovation. It shows how integration complexity and efforts may be minimized in building bridges between the sectors.

Approach to Trans-sector Integration

From the above, we found a diverging trend through time, the *functional decomposition* of tasks, leading to a set of loosely coupled sectors. Re-unification of the sectors under a single authority is a utopian dream. Integration approaches will have to be deployed supporting efficient innovation in a distributed world with many sectors without central authority to impose a preferred information architecture onto all sectors. A distributed architecture for trans-sector integration is required with collaboration based on choreography rather than orchestration. Hence, the view as presented in this paper builds upon two basic assumptions:

- **Top-down development of trans-sector innovations while preserving autonomy of individual sectors.** The functionality to be realized must be leading in the development process for trans-sector innovations. Hence, its business requirements must be top-down translated into trans-sector processes and implementations. Such a top-down approach however must be able to cope with the autonomy of the individual sectors in which individual sectors keep their own (governance of) functionality, internal working, IT architecture and solutions.
- **Dynamic discovery and integration of a sector's functions.** Trans-sector processes require the integration of the functions provided by the individual sectors. Such integration currently takes much effort and is slow. An important cause is that the functions provided are usually only described on a syntactic basis. This leads to intense interaction to find out whether the function's details meet the requirements (discovery), realize data-mappings and agree upon service levels before interconnections can be made. Using richer and unambiguous semantic description of the functions can make the integration process a lot quicker, perhaps even fully automated.

The following sections describe a view on how both basic assumptions can be met. This will reduce the trans-sector integration effort considerably. The next section describes a top-down approach for the development of trans-sector processes with preservation of the sector's autonomy. The subsequent section considers the dynamic discovery and integration of the sector's functions.

Top-down Development of Trans-sector Processes Using MDA and SOA

The enormous advent of ICT in all sectors of society over the last decades has led to an urgent need for structural approaches for overcoming the integration challenge. Therefore, various new integration solutions, technologies, protocols and standards have been developed over the recent years, of which the concepts of Service Oriented Architecture (SOA) and Model Driven Architecture (MDA) have gained enormous momentum. They form the basis of the view on the top-down development of trans-sector processes as described in this paper. Both architectural concepts are briefly described. A concise overview on integration patterns can be found at [3].

Service Oriented Architecture

In the world of distributed IT-environments the concept of Service Oriented Architectures (SOA) has widely been adopted as means to realize an agile architecture. The essence of service orientation is about (re)designing the infrastructure around providing and consuming services, as opposed to the more traditional approach of interconnecting applications and their embedded processes. Service orientation reflects the everyday service oriented world. The goal is to design services as stable factors, while realizing agility through dynamic business processes that re-use the individual services. A clear separation in (stable) service-logic and (dynamical) process logic is envisioned, Fig. 2a.

SOA uses the principles of a distributed, interface-oriented architecture. These principles are not new. However, the emergence of web services (WS) technology has spurred the SOA-concept. Currently, a rich suite of WS-protocols and – standards has been developed by the OASIS organization and the W3C-consortium.

Model Driven Architecture

As said, the business requirements for the trans-sector innovations must lead the development of processes and implementations. The objective of MDA (an initiative of the Object Management Group - OMG) is to automate and provide consistency in the development process from business requirements to implementation. To this end, MDA uses formal description models at different levels of abstraction (Fig. 2b) with well-defined mappings and transformations for automatic translation between the formal description models of the various levels. This way, the top level business requirements (formally described,) can be automatically and consistently converted into implementations.

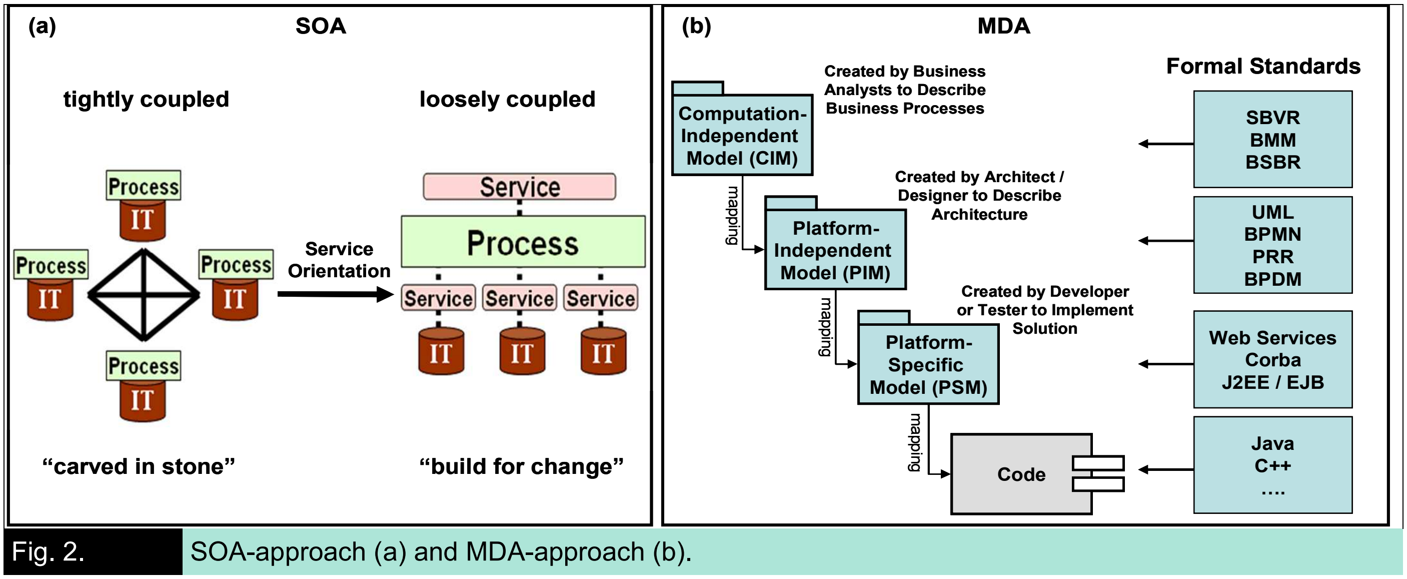

Fig. 2. SOA-approach (a) and MDA-approach (b).

Figure 2b illustrates the layered MDA approach. It shows the four levels of MDA. At each level a different formal model is used to describe the architecture at a different level of abstraction. The figure also enumerates some formal standards used in each level. In top-down direction, the models used at the various levels are:

- the Computational Independent Model (CIM) formally describing the business requirements, e.g., in common understandable language;
- the Platform Independent Model (PIM) describing the architecture in a manner that is independent of implementation technology;
- the Platform Specific Model (PSM), tailored to a specific implementation technology;
- the Code providing the actual implementation of the functionality.

In our view, both SOA and MDA have their own merits and role to play for trans-sector innovation. MDA provides the top-down approach that allows the trans-sector business requirements to be defined and to be translated into processes and implementations. SOA abstracts the functions and data of the individual sectors into re-usable (web) services to be used by the trans-sector processes while keeping the autonomy of the individual sectors.

However, it is to be noted that the MDA and SOA concepts can not readily be combined. SOA reflects a bottom-up reverse engineering approach abstracting available functionality and data in a sector. MDA on the other hand, provides a top-down approach deriving functionality for lower-level implementations. Hence, exploiting the merits of both approaches requires a unification method for the bottom-up SOA approach and the top-down MDA approach.

Several methods to unify the bottom-up SOA and the top-down MDA approach have been presented in literature. As such, the extensive work of British Telecom (BT) deserves special attention [4]. BT has elaborated a method using MDA in conjunction with component (service) based OSS standards to be deployed in a telco environment. They use (bottom-up) reverse engineering of the SOA-compliant standardized TMF NGOSS and OSS through Java (OSS/J) services into re-usable PSM's and PIM's. By exposing the component capabilities in model form, they can be used in new PSM's and PIM's through (top-down) forward engineering. They demonstrate the method for the case of a network inventory application with graphical interface for which the code has 100% automatically been generated. Their methods and the relation with the TMF NGOSS life cycle has further been demonstrated in various TMF Catalyst demonstration projects.

An alternative method is presented in [5]. It describes an intermediate layer for coupling the top-down MDA-approach and the bottom-up SOA-approach. The top-down MDA approach is used to generate a PIM and a SOA-specific PSM down to the lowest implementation part of the PSM that is represented by BPEL and WSDL files for the executable business processes and web services that can be invoked. Both a process-model and a data-model are included. Such an MDA approach is supported by the various initiatives for generating UML profiles for web services [6], [7] and the current OMG-initiative (as owner of the UML standards) for formal standardization hereof. Also in this method, SOA is used for bottom-up reverse engineering and abstracting available functionality and data in a sector. The intermediate layer maps and transforms functions between on one hand the web services created by the top-down MDA-approach and on the other hand the web services created by the bottom-up SOA-approach, Fig. 3. As part of its coupling, mapping and transformation function, the intermediate layer can:

- absorb unwanted changes from bottom-up to top-down, e.g., changing a data-base or application in a sector (resulting in a change of a low-level web-service) doesn't affect the high-level processes;
- hide semantic differences;
- integrate (results) of various low-level sector services into a single high-level trans-sector service.

Both the described methods for unifying the bottom-up SOA and the top-down MDA approach will have their own pro's and con's in the context of trans-sector integration as discussed in this paper. The former method seems to provide better opportunities for re-usability of PIM's and PSM's. But it also assumes relative stable and standardized low level services that are generated. Within a single sector with a strong standardization effort (such as for the telco case) this may be true and may therefore lead to less integration tax. In a trans-sector situation with many different sectors, non-standardized services and many alternative service suppliers this will

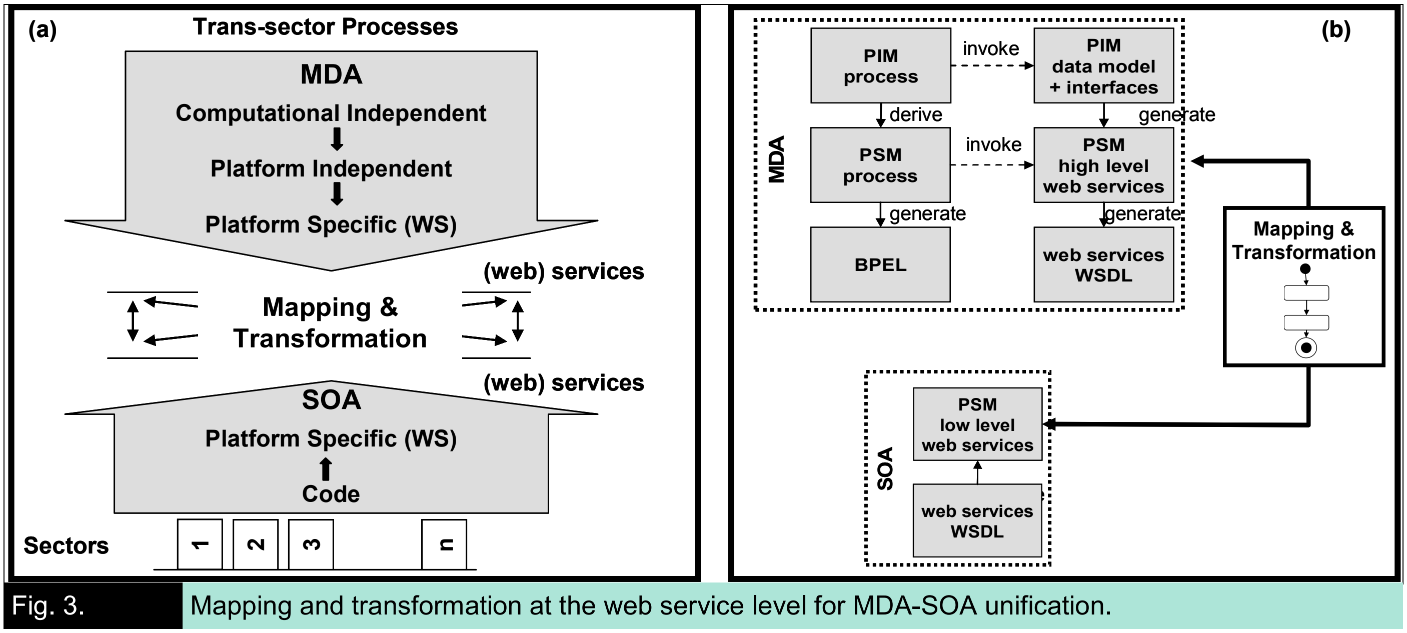

Fig. 3. Mapping and transformation at the web service level for MDA-SOA unification.

probably be less the case. In that case the latter method may prove to be more advantageous, although it results in (manual) mapping and transformation steps between top-down MDA-generated and bottom-up SOA-generated web services.

Dynamic Service Discovery and Integration Using Semantic Web Services

As discussed thus far, trans-sector innovation requires the integration of the (web) services that are provided by the sectors into trans-sector business processes. Dynamic discovery and integration of these services may contribute significantly in minimization of the trans-sector integration effort. Mere syntactic description of the services is therefore insufficient[1]. The unambiguous semantic description of the web services that are exposed is required for dynamic discovery and integration. In scientific literature various semantic web services methods have been presented. Reference [8] describes a telecommunications case study. It describes how semantic web service technology can be used so that service providers can dynamically integrate with a wholesale business-to-business gateway of an incumbent operator to allow them to integrate their operational support systems. Without such a system, a large amount of manual integration effort would be required. But such methods good enough to enable dynamic discovery and integration in the broader context of trans-sector innovation as described in this paper? What is (currently) feasible? Literature doesn't provide a direct answer. Hence, this section describes the state-of-the-art in semantic web services by considering the various (structure and process) components that together unambiguously define a service. Formal and unambiguous semantic description of all these components is necessary to facilitate the dynamic discovery and integration of such services.

The structure components of a service

The structure components describe what a service is. There are three structure components that are required and sufficient to unambiguously describe a service:

- The effect of a service. This describes the results of the processes and activities that a service performs.
- The message exchange on the service interface. This describes the messages that are sent and received over its interface when calling the service.
- The service level agreement. This describes a list of criteria which is used to measure performance of the service. It contains a consequences clause that determines what happens when the performance agreed upon is not realized.

An ontology provides the means to unambiguously define terminology. Therefore an ontology forms the basis of a semantic web service method. When an ontology unambiguously specifies knowledge with mathematical (logical) constructs it becomes possible (for machines) to reason about the contents of a service. A service can then be interpreted by a computer. If the three structure components of a service can be expressed by such an ontology, the service is unambiguously described.

The process components of a service

The process components describe the activities that are required for dynamic discovery and integration of a service, i.e.:

- **The discovery of a service.** A (potential) consumer formulates the service he wants to use. Subsequently, it is determined which suppliers can provide the requested service. The consumer receives a list of potential suppliers and selects one of them. In current, non-semantic, web service architectures mostly a relatively primitive method using key words attributed at service publication is used to search for services, e.g., by means of an UDDI directory. However, this is insufficient. Ontological service descriptions of semantic web services are more precise.
- **The parameter refinement for a service.** This determines specific values for state-parameters and additional details for providing the service after the service has already been discovered. Parameter refinement is re-

[1] With special thanks to Bart Vrijkorte for his work on the assessment of semantic web services as presented in this section.

quired when provisioning the service depends on the state of the consumer. This can be illustrated by means of a parcel delivery service. State-information such as “Is the parcel delivery address within the servicing area?”, “What is the timeslot for retrieving and delivering the parcel?”, “What is the volume of the parcel?” and “What is the price?” will not be available after the discovery process. Parameter refinement must result in specific parameter values that are satisfactory for both the consumer and the supplier of the service.

Currently, various methods for semantic web services are in development. The three most well-known and advanced methods are OWL-S, WSMO and SWSF. Table 1 shows how these methods are evaluated against the structure and process components as introduced in this section. The table indicates that the methods do not fulfill all requirements (yet).

As the table shows, none of the methods provides a sufficient solution on how to negotiate and enforce service level agreements in a service-oriented architecture, although this aspect is rapidly becoming of increasing importance.

Secondly, each method has a limited solution for service discovery as these methods do not take into account the state-changes (the effect) that a service will realize. Instead, a service is formulated as if it were an object in the world. This is not very precise. Again, this can be illustrated for the parcel delivery service. With the object-centric methods for discovery, the parcel delivery service is described as the object transport with attributes for the parcel-retrieving address, the parcel-delivery address and the price. In no way it can be derived what the essence of the service is, i.e., that the parcel changes location. It will be better to find a parcel delivery service by describing the object parcel and to search for a service that can change the current location of a parcel to the delivery address. This will make it possible to split this effect into several partial effects in which different parties will transport the parcel a part of the route. This advanced manner of searching is impossible with current object-centric methods.

Finally, there is no answer to the complex issue of parameter refinement. Often, a consumer wants to define specific pre- and post-conditions for a service (e.g. the parcel should change locations between specific addresses), whereas suppliers will use more generic pre- and post-conditions (a parcel delivery service in Europe). Checking whether these different views on pre- and post-conditions are in agreement is a difficult (logical) problem. Moreover, a service may require additional pre- and post-conditions with which the consumer may not be able or willing to comply (e.g., the parcel should weigh less than 1 kilogram or the receiver of the parcel should also pay an amount of money). Hence, a parameter refinement procedure is often inevitable, including negotiating the price of the service (which is currently not possible). This poses a major challenge on semantic web services.

But, although these considerations indicate that fully dynamic discovery and integration of services is not yet feasible, it is already feasible to speed up the integration process by using semantic information. This is under the condition that the organizations within the same sector develop a shared ontology for their sector. On the basis of this model, they should develop and describe their services. These descriptions can initially by informal (in a natural language). At a later stage, they can be formalized. In the telecommunications sector, the TMF’s Shared Information Data model (SID) provides a good basis for this purpose.

Conclusions - Bridging the Gap

In this article, we have described a view on how MDA, SOA and semantic web service technology can be used in trans-sector innovation for minimizing integration efforts. Although, these technologies are good steps forward towards the ultimate goal of highly-automated trans-sector process development and integration, still some gaps have to be bridged before manual efforts can completely be banished, especially on:

– unification of the top-down MDA-approach for the development of trans-sector business processes with the bottom-up SOA-approach for exposing sector-specific services.
– dynamic discovery and integration of services using semantic web service technology.

But although the ultimate goal may appear to be far away, it is to be noted that the current state of technology can already help considerably in minimizing the trans-sector process development and integration effort. With the view described in this paper, trans-sector integration can be reduced to mainly service discovery and mapping/ transformation of semantic (web) services exposing each sector’s functionality. Having to interconnect (by manual effort) deeply into the IT-systems of the individual sectors will no longer be necessary.

Furthermore, additional steps can be taken to further combine the concepts of

Table 1. Evaluation of semantic web service methods

Criterium	OWL-S	WSMO	SWSF
Representation of business process/effects	yes, non-composite services through pre- and post-conditions, composite services through a process model	yes, services are described through pre- and post-conditions	yes, non-composite services through pre- and post-conditions, composite services through a process model
Message exchange on service interfaces	yes, but only one single input- and output message per service	yes, every interaction patter is feasible	yes, with process-model for composite services
Representation of service level agreements	not in method	not in method	not in method
Service discovery	discovery based on object-centric approach	various proposals but not yet decided upon, not on the basis of state-changes	limited, based on the centrally defined objectives for a service
Parameter refinement	not in method	not in method	not in method

MDA and semantic web service discovery into a unified development process. To this end, it should be possible to embed semantic web service discovery processes as part of the MDA development methodology. The first ideas on how to realize this have already been published [9].

References

[1] A. Tofler, ”*The Third Wave*”, 1980.

[2] M. Castells, “*The Information Age: Economy, Society and Culture 1996-1998 and The Rise of the Network Society*”, 1996.

[3] A. Arsanjani, “*Toward a pattern language for Service-Oriented Architecture and Integration*”, http://www-128.ibm.com/developerworks/webservices/library/ws-soa-soi/.

[4] M. Azmoodeh, N. Georgalas, and S. Fisher, ”*Model-driven systems development and integration environment*”, BT Technology Journal, vol. 23, no 03, July 2005, pp. 96-110, ISSN 1358-3948, Springer, The Netherlands.

[5] T. Dirk Meijler, G. Kruithof, and N. van Beest, ”Top *Down Versus Bottom up in Service Oriented Integration: an MDA-based Solution for Minimizing the Technology Coupling*”, in Proceedings of The International Conference on Service Oriented Computing, Chicago 2006, to be published in Lecture Notes on Computer Science in 2007.

[6] D. Skogan, R. Groenmo, and I. Solheim, “*Web service composition in UML*” in Proceedings of the Eighth IEEE International Enterprise Distributed Object Computing Conference, (EDOC 2004) IEEE, 2004.

[7] F. Lautenbacher, “*UML Profile for and transformation rules for semantic web services*”, Report 2006-20, Institüt für Informatik, Augsburg, http://www.informatik.uni-augsburg.de/lehrstuehle/swt/vs/publikationen/reports/2006-20/.

[8] A. Duke, J. Davies and M. Richardson, “*Enabling a scalable service-oriented architecture with semantic web services*”, BT Technology Journal, vol. 23, no 3, July 2005.

[9] K. Yang, S. Ou, M. Azmoodeh, and N. Georgalas, ”*Model-based service discovery – prototyping experience of an OSS scenario*”, BT Technology Journal, vol. 24, no 2, April 2006, pp. 145-150, Springer, The Netherlands.