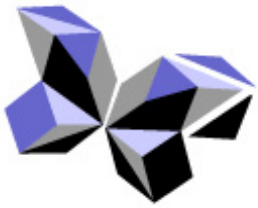

Digital Business Ecosystem

Contract n° 507953

Workpackage 34:
Exploitation & Sustainability

Deliverables D34.4.2:
Market Watch - Service Oriented Architecture and Semantic Middleware

Information Society Technologies

Project funded by the European Community under the “Information Society Technology” Programme

Contract Number: 507953

Project Acronym: DBE

Title: Digital Business Ecosystem

Deliverable N°: D34.4.2

Due dates: 12/2006

Delivery Date: 03/2007

Short Description:

Author: IBM

Partners contributed: IBM

Made available to: Public

Versioning

Version	Date	Author, Organisation
0.1	31/01/2007	Peter Stanbridge, Korora Ltd
0.2	31/03/2007	Tim Romberg, FZI

Quality check:

1st Internal Reviewer : N/A

2nd Internal Reviewer: N/A

Service Oriented Architecture and Semantic Middleware

Market Watch

Abstract. This final DBE market watch looks at the current trends in the areas of service oriented architectures (SOA) and current applications as well as limitations of semantics in the SOA area. Introduction

Introduction

One great challenge in today's digital business exchanges is the flexibility required for both customer customization of service offering and automated support for the diversity of interfaces that need to be integrated and with minimum effort. [Granebring et el]. Also, there appears to be a variety of competing paradigms for addressing the various aspects of this problem.

In the last market watch the challenge of service discovery was discussed, where the emergence of the social web was considered a potential contributor toward resolving the problems of rigidity in conceptual specifications, primarily given the dynamic and complex nature of a business eco-system. It was considered there to be a very promising approach to semantic descriptions of business organization and service descriptions.

The matter changes when one considers software interfaces where service connections, object instantiations, method calls and data formats must be not only machine readable, but unambiguously machine processible. This is the domain of service oriented architecture and web services (and other competing approaches such as Restful web services) and researchers in this area have looked to the semantic web community for the intellectual tools to address the semantic interoperability problem.

There exists yet another level of semantic descriptions required to enable the digital business eco-system. This level has been specifically addressed by the OMG in its SBVR specification. Contractual and business process obligations amongst a collection of interacting organizations can be extremely complex and need to be specifiable if software services are to be automated to reflect the transaction requirements of the underlying events within the business processes.

Together these represent huge topics that are on the current edge of both theoretical and practical based researches.

The aim of this market watch is to provide some background to these efforts and to the motivations that have made them popular and important areas of research. It will give some indication to current research and challenges as well as a connection with the efforts of the DBE.

The main conclusion is that the overall challenges of software service interoperability is a huge issue where it is still far from known how solutions would even look like let alone there being a clear path toward its solution. The DBE delivery is to be seen as an important step forward in understanding some of the critical business and technical issues surrounding software integration of this nature and it is to be hoped that partners working

within the DBE and those associated with it will be able to take the problem forward through follow up research within the universities and commercial R&D within the project's organizational partners.

Service-oriented architectures

Service-oriented architectures (SOA) has its place at the end a progression of computing architecture paradigms beginning with mainframe systems where applications were tightly coupled (or not coupled at all) progressing through the client server and distributed computing paradigms. Service-oriented architecture is therefore not a particular platform or software product but a genuine architecture instantiated by a number of different standards and software platforms – web services (WSDL, SOAP, UDDI) being and example.

The principle insight behind SOA is the idea of loosely coupled software services. It has been described in terms of object oriented software development. At the source code level, amongst other things OO is about inheritance as a means of specialization and generalization amongst software classes. SOA is about component composition at run time, where software components can call on the services of other software components to achieve a task.

A particularly important aspect about SOA is that it requires a different approach to software design and construction. Motivated in business terms by the increasing requirement for flexible and configurable software services SOA has promised to be an important architecture over the coming years. Its growing interest can be seen in the number of publications dedicated to it – with approximately 70 books with SOA or "Service Oriented Architecture" in the title and around 280 titles for web services.

Not to be beaten by the SOA boom the European Technology Platform on Networked Software and Services (NESSI) is talking in its NGG3 report already about going *beyond SOA* with the Service-Oriented Knowledge Utilities (SOKU) concept. SOA is still at the base of SOKU, but it adds knowledge ("SOKU services are knowledge-assisted ('semantic') to facilitate automation and advanced functionality, the knowledge aspect reinforced by the emphasis on delivering high-level services to the user", [NGG3 report pg 5]). It also adds utilities ("a utility is a directly and immediately useable service with established functionality, performance and dependability, illustrating the emphasis on user needs and issues such as trust. [NGG3 report pg 5]). But these sound very similar to SOA itself, with its emphasis on semantics and the requirements of utility. What might be different is "The primary difference between the SOKU vision and earlier approaches is a switch from a prescribed layered view to a multi-dimensional mesh of concepts, applying the same mechanisms along each dimension across the traditional layers" [NGG3 report pg 5]. The report clearly states that the idea of millions of interacting software components across a global network cannot be built on the client server model. It is too complex. The NGG3 report contains a SWOT analysis of research challenges for the EU in traversing a road map toward the SOKU vision. The main conclusions of the SWOT analysis are:

- Ontologies and semantic web technologies will be crucial to provide scalable support for complex, heterogeneous Grids middleware and applications.
- The strengths of the European telecommunications industry and the diversity of its market for electronic control systems have given Europe a leading position in the

areas of mobile and embedded technology. This is of particular relevance for the realization of the vision of a Grid as a pervasive, user-centered utility.

- The weakness in hardware and primary software products (e.g. commodity processors, server and desktop operating systems, programming languages, etc.) may hamper the development of a European leadership in Grids technologies.
- The convergence between Grids and web services provides a significant opportunity to move to a model of software development and service provision where the market dominance of particular OS vendors is no longer a major economic issue.
- The distinctive European vision of a Grids environment that operates from the level of devices to supercomputers, to serve communities ranging from individuals to whole industries, including data, information and knowledge and emphasizing resilience and scalability could have a significant economic and social impact far beyond the scope of existing compute and data Grids. This should be contrasted with the North American Grid vision of programmer-level meta-computing.
- It is vital that any European vision for the evolution of Grids is accompanied by a clear representation of that vision to the key standards bodies and technology providers worldwide.

From this list one can see the inclusion of many of the semantic web ideas and technologies and the convergence of the semantic web and web services. While this European vision is challenging, the NESSI ETP – taken alone - has already the support of currently 300 member organizations including, European SMEs and research institutions as well as major global IT vendors and European industry leaders. However, it can be estimated that many more European research and industry partners share the vision that Europe – as place of birth of the www - can also take a leading role to create software standards, platforms and products that will implement the vision underlying the next generation grid leading to a service oriented knowledge utility. To achieve this will require a considerable commitment to longer term foundation and action based research.

Web services

Web services are a cast of technology standards that supports the SOA architecture vision. Web services forms a core technology in the vision to interoperate software services – even though web services, unlike J2EE, CORBA and .NET, are not executable. They are better seen as XML-based interface technologies[1], although it would be wrong to consider web services as the only approach to SOA[2].

The two principle technologies belonging to the web services standard are WSDL and SOAP. These technologies are well understood in the DBE project and the DBE certainly takes these ideas forward with the MDA approach to modeling the interfaces.

[1] [Newcomer & Lamow2005] page 19

[2] E.g. RESTful approach to web services, which is suppose to be much simpler and faster because it avoids the overhead of SOAP. But for the purpose of this Market Watch where the emphasis is on semantics not performance, I do not think the distinctions are important.

Given the implementation choices of the DBE it is more useful to explore some of the other approaches to web services mentioned in the literature and to see whether alternatives to MDA, the UML driven BML and its replacement with SBVR are worth exploring.

Semantic Middleware

A key challenge is software interoperability within the context of e-business. This is a tricky issue even before one considers the eco-system vision of the DBE or the millions of independent software components vision of the Next Generation Grid report [NGG3 report].

It is often stated that the BML version 1 is an ontology driven approach, but ontology representation in the form of description logics can extend standard UML to a significant degree[3]. DL can be used, for example, to detect inconsistencies in UML models. However one sees the use or comparison between UML and ontologies, it is useful to understand two related challenges to the specification of interfaces using the type of semantics available in UML (or certainly the ontology approach adopted in the BML implementation but not restricted to ontologies in general).

The first (and oldest) is a problem described by Sowa in [Sowa 1984] quoting [Partee 1971] which he stated as “A notation by itself has no meaning”. This is a serious challenge because it says that labeling a box and generating a software interface artifact of itself isn’t at all semantics – it is machine readable or machine executable (depending on what it is) but it in itself (or its interfaces) is really syntax. In the terminology of standard first order theory, what is missing is a logic to operate on the syntax and this is an important element in the description of a software interface definition.

The second is closely related but further up the “value chain”. It claims that supplying a first order model to some syntax is insufficient for the purposes of ontology descriptions required for software and information descriptions. This challenge was first given by [Guarino & Giaretta 1995], [Guarino 1998] and other writings from Guarino and his collaborators. The basis of this challenge is that if one has a first order model of some situation in the world all one has is an extensional model of a particular configuration in time. A change to the configuration would require a new model and hence a new conceptualization (as described by the standard definition of an ontology as an “explicit specification of a conceptualization” (found in [Gruber 1995] who in turn is borrowing from [Genesereth & Nilsson 1987].

What is required, according to Guarino, is an intensional logic founded on alethic rules (possible worlds semantics – and getting closer to SBVR). Guarino proposes a very precise possible worlds semantics for his conceptualization and provides details of how ontologies would be considered to be a specification of such a conceptualization.

The key challenge from Guarino is to provide an implementation of a genuinely intensional semantics that would truly describe what is meant by the key syntactical terms (lexicon)

[3] As [Oberle 2005] page 35 points out, that ontologies differ from UML, ERD etc. in that (1) the primary goal of ontologies is to enable agreement on the meaning of specific vocabulary terms and, thus, to facilitate information integration across individual applications, (2) ontologies are formalized in logic-based representation languages and (3) the representational languages come with executable calculi enabling query and reasoning at run time.

used in a model. It would be insufficient to simply state, as is done by Genesereth, that block A was on top of block B, which are next to block C, or that software component A is dependent (in some way) on component B and that component C is also dependent on B while component D is not dependent on any other component.[4] The semantics needs to provide a specification that would determine in any given configuration what "on top of" and "beside" were and what software dependency between any component means so that if component D became dependent on component B, a new conceptualization would not be necessary.

In practice, the task of providing such semantics is extremely difficult and as with most of the SOA movement, still in research.

A significant contribution to the semantic aspirations of the DBE project is to be found in the work of the WonderWeb (Ontology Infrastructure for the Semantic Web) project, a European Union Information Society Technologies (IST) Future Emerging Technologies (FET) funded project.[5]

A key publication that indirectly came from the WonderWeb project is the book by Daniel Oberle [Oberle 2005] which provides a detailed analysis of the semantic requirements for middleware solutions that implement services similar to the DBE.

The book is significant in the progress it has made in the semantic area – it takes into consideration the intensional logics of Guarino; it utilizes foundational ontologies either directly or inspirationally[6] and it addresses many of the practical issues relating to languages, implementation details (the design of an ontology-based applications server, for example) and sets the work of SOA into the history of middleware development. Wisely Oberle is careful to indicate that the semantics provided for the middleware will go some way toward a **partial** automation of middleware management. The Holy Grail is still a long way off.

Oberle begins by asking the motivating research questions structured into a hierarchy, which are useful to repeat in full here, since they are worth comparing with the DBE middleware objectives.

Cardinal Question *Can ontologies be used to facilitate the development and management of middleware-based applications for developers and administrators?*
This in turn creates three main questions with sub questions.

Main Question 1. *How to find a good trade-off between modeling and management efforts?*
Question 1.1. *Who uses semantic descriptions?*
Question 1.2 *What are the semantic descriptions used for?*
Question 1.3 *When are the semantic descriptions used?*
Question 1.4 *Which aspects should be formalized into our ontology?*

[4] An example similar to that given in [Oberle 2005] page 36, to be discussed below.

[5] See http://wonderweb.semanticweb.org The project completed in 2004

[6] Potentially overcoming some of the criticisms Sowa has of the semantic web for moving in the opposite direction to AI – for which there are countless references of papers on the web (mainly in discussion groups) or examples like the very short http://www.dehora.net/journal/2003/04/semantic_web_crippled.html

Main Question 2 *How to build a suitable management ontology?*
Question 2.1 *Can an existing ontology be revised for our purposes?*
Question 2.2 *How to ensure high quality?*
Question 2.3 *How to decrease modeling efforts and enable reuse?*
Main Question 3 *How to realise semantic management of middleware?*
Question 3.1 *What is a suitable target platform?*
Question 3.2 *Who provides semantic descriptions?*
Question 3.3 *How to implement semantic management*?
Question 3.4 *how to reuse the ontology?*

These questions would have been very useful for the early stages of the DBE project and Oberle provides very detailed and technical answers to them throughout its 253 pages.

In the remainder of this section I will provide a brief sketch of the book, primarily as an indicator to how relevant his work is to fundamental research in this area.

The first part of the book presents fundamentals. The opening sections provide a historical map of middleware developments leading to web services. The history is not linear, but contains many branches that were not then directly used in later stages of development. For example, according to Oberle, Web Services transitively subsumes object brokers (through application services and object monitors). His conception of web services (and SOA) is similar to that already given above – the core principle being the looseness of the coupling.

Following the historical development is a section on ontologies. The manner of presentation is highly technical as Oberle wishes to distinguish the extensional definitions from the intensional ones, and makes a commitment to the intensional logics as being the most appropriate for the task.

The final chapter to the first part outlines the basic components of the middleware requirements making a distinction between modeling and management (it is the management that modeling is supposed to support). It is here that he lays out a careful argument for the importance of rigorous modeling.

The first chapter of the second part discusses several foundational ontologies with the purpose of selection for a foundational basis for the middleware semantics. It is interesting that this was the WonderWeb's starting point, and an argument is given that one should start by evaluating existing ontologies before considering the design of a specific ontology to support automation of the middleware management.

Ontologies presented by Oberle are OWL-S, BFO, DOLCE, OCHRE, OpenCyc and SUMO. They are all given penetrating analysis, evaluation and criticism. The DOLCE ontology is selected as the most appropriate foundational ontology since it meets all of the requirements set out before the evaluation.

The next chapter develops a middleware ontology based on DOLCE. A number of UML diagrams are presented to describe the ontologies and are worth comparing with the BML models. The first major difference is that the models developed by Oberle are very generic, while representing typical abstract real world conceptualizations; they are not tied to business organizational concepts. This is a significant difference to the DBE BML, which was an attempt to combine the semantics of the middleware and the semantics of the

business real world service discovery and description (this was the intent, the split between business modeling and SSL makes the comparison more complex than given here). A clear distinction is made between software and data along with a careful description of the importance as well as ambiguities associated with this distinction.
The analysis finishes with detailed logical descriptions and axiomatisation of the models, even to the detail of specifying API interfaces with a particular emphasis given on them enabling reuse – something we found difficult with the BML version 1.

The third part deals with the inference engine or processing side of the ontology – giving life to an otherwise passive object. It describes an implementation based on JBoss, although I am unsure as of writing whether this ontology system had been implemented and linked into a live ebusiness middleware installation.

In the final section, the book covers related work, which brings us to the next section of this Market Watch.

Semantic Middleware and MDA

The MDA has had a special place in the DBE. It is used reasonably literally in the SSL layers and rather metaphorically in the (delivered) BML models.

Is this a good place to work the basis of the semantics of middleware?

In some modeling exercise with the University of Central England, the author stumbled across many barriers of which two particular problems stuck out as stemming from the use of the MDA. The first was that what we were referring to as M1 models needed to be mixed – that is they needed to contain entities that would otherwise seem to need to appear on different levels. The second is the difficulty faced when attempting reuse models where for some organizations what would be at the M1 level was another organization's M0 level– the same real world entities (or the concepts applied to them) were at different levels for different organizations.[7]

But this is more related to the BML, which isn't under discussion here. For the situation of the semantics of middleware another viewpoint is necessary (since the SSL in the DBE is closest to the middleware management treated by Oberle). I quote in detail from [Oberle 2005] pg 227, 227:

> *"…However, in contrast to our work, MDA uses the models mainly to specify development aspects with focus on distinguishing platform-independent from platform-specific aspects, as well as on the definition of the transformation rules between them. While something similar can be done also for the management of middleware, the main use case of MDA is to generate an executable application out of the platform-independent model. Our approach focuses on run time relevant characteristics of component and service management, such as which version of an application interface requires which versions of libraries. We exploit the logic-based semantics of ontologies for*

[7] The common one given by way of illustration (because it is easy to understand) is that in the automotive industry one could consider new dealerships that take orders for a model (which is them manufactured), car rental firms that take orders by class and only allocate to an instance of a model later and second hand car dealers who receive orders on an individual car.

> *querying the inference engine in application server whether configurations are valid or whether further components are needed. Ontologies are best-suited for this purpose."*

Oberle then mentions the new developments of the Ontology Definition Metamodel (ODM) by the OMG and indicates that this may realize the WonderWeb ontology models into UML, but from what I have been able to find, this is still an open question.

Semantic Middleware and SBVR

The semantic middleware as described by Oberle and the WonderWeb project is not fully addressing the same space as SBVR except that the OMG vision for SBVR takes the middleware to much more advanced levels (beyond the technical interfaces for interoperability based on simple business activities to the support of very complex interactions of digital business events).

In this section a few comments on business rule logics will be briefly discussed, primarily followed by some cautions in its universality.
We begin with a quote from the business rules group.

> *Information Systems analysts have long been able to describe an enterprise in terms of the structure of the data the enterprise uses and the organization of the functions it performs. Unfortunately, there is often neglect of the rules (constraints and conditions) under which the enterprise operates. Frequently these rules are not articulated until it is time to convert them into program code. While rules that are represented by the structure and functions of an enterprise have been documented to a degree, others have not been articulated well, if at all. The Business Rules Group was organized to carry out that articulation*. http://www.businessrulesgroup.org/bra.shtml

In examining the background and some details of the SBVR standard (and earlier work such as that done by Sowa and Halpern (References for Sowa below and Halpern can be found via Google)) it appears as though full automation is going to require a much more comprehensive logic than what was originally conceived necessary within the DBE. However, it is one thing using deontic logic to reason about the nature of obligations within a philosophical or legal context, but it is another to believe that such a language (which must include alethic elements) is capable of expressing the human nuisances that go hand in hand with actual trading relationships. A strongly formal analysis of such business acts and obligations may be seen by some as being rather sterile and missing many of the vital points of human interaction and communication that play important parts in the real-world setting of business and subsequent processing transactions.

The next quote comes from the OMG RFP (when SBVR was called BSBR).

> *"The objective of this RFP is to allow business people to define the policies and rules by which they run their business in their own language, in terms of the things they deal with in the business, but in ways that are clear, unambiguous and readily translatable by Information systems experts into executable rules for many kinds of automated systems."*
> *~ (from the BSBR RFP)*

Many references on the web (Google search) show scepticism about SBVRs user friendliness and whether it would be truly possible to model business rules using an English language interface. I cannot pass judgement on this, but it represents a serious challenge to be able to represent complex real world scenarios (that often changes with shifts in people and political alliances) in sufficient detail and without ambiguity from an business English description.

Finally, the following quote from real rules blozine.

> *"A business vocabulary is defined to contain "all the specialized terms and definitions of concepts that a given organization or community uses in their talking and writing in the course of doing business". A business rule is defined as "a rule that is under business jurisdiction", which means that "the business can enact, revise, and discontinue business rules as it sees fit". Found in http://www.realrules.info/?q=node/43.*

I just question the feasibility of ever achieving this – both from completeness as well as a consistency point of view. Is business language stable enough to ever achieve this?

Conclusions

Once again the short research time made available to the Market Watch has made it difficult to provide a seriously detailed analysis of this big subject. With the help of the work undertaken by the WonderWeb project and meticulously documented by [Oberle 2005] and a host of other web based and book based resources, an interesting picture can be observed relating to the complexity and size of the interoperability problem. It is multi-faceted covering descriptions of businesses that are more promotional oriented to business descriptions that entail serious implications for business logic (that possibly needs to be executed on or supported from software services published within the digital business eco-system) as well as the descriptions of the software services themselves and the different requirements on the data structures representing the business data being exchanged as part of the service.

This is an extremely ambitious undertaking that is really too much for a single project to take on. Semantic interoperability is work in progress and experimentation and an open mind regarding the frameworks and methods used to address it will be necessary as current technologies may lead development into blind alleys that in itself may lead to new solutions (for example, as the ODM and SBVR might from the initial attempts to work within the MDA).

[Genesereth & Nilsson 1987] Genesereth, Michael R. and Nilsson, N.J. *Logical Foundations of Artificial Intelligence,* Morgan Kaufmann, 1987

[Granebring et el] Granebring Annika, Thilénius Peter, Révay Peter; SOA is a driver for Personalization, found at http://www.eki.mdh.se/personal/agg01/ECITE06-SOA%20is%20a%20driver%20for%20Personalization.pdf last accessed 12 March 2007.

[Gruber 1995] Gruber, Thomas R. Toward Principles for the Design of Ontolgoies Used for Knowledge Sharing. *International Journal of Human Computer Studies,* 43(5-6):907-928, 1995

[Guarino 1998] Guarino, N. Formal Ontology in Information Systems. In Guarino, N,. Editor, *Formal Ontology in Information Systems.* Proceedings of FOIS'98, Trento, Italy, June 6 – 8, 1998, pages 3-15. IOS Press

[Guarino & Giaretta 1995] Guarino, N. and Giaretta, P. Ontologies and Knowledge Bases: Towards a Terminological Clarification. In Mars, N., editor, *Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing*, pages 25-32, IOS Press

[Marks & Werrell 2003] Marks Eric A. Werrell Mark J. *Executive's Guide to Web Services*, Wiley, 2003

[NGG3 Report]

[Newcomer Eric, Lomow Greg, *Understanding SOA with Web Services,* Addison-Wesley, 2005.

[Oberle 2005] Oberle, Daniel *Semantic Management of Middleware* Springer, 2005

[Partee 1971] Partee, Barbara Hall "Linguistic metatheory" reprinted in G. Harman, ed./ *On Noam Chomsky: Critical Essays*, Anchor Books, 1971, 301 - 315

[Sowa 1984] Sowa, John *Conceptual Structures: Information Processing in Mind and Machine* Addison-Wesley, 1984