A Semantically Enabled Service oriented Architecture

Darko Anicic[1], Michael Brodie[2], Jos de Bruijn[1], Dieter Fensel[1], Thomas Haselwanter[1], Martin Hepp[1], Stijn Heymans[1], Jörg Hoffmann[1], Mick Kerrigan[1], Jacek Kopecky[1], Reto Krummenacher[1], Holger Lausen[1], Adrian Mocan[1], James Scicluna[1], Ioan Toma[1], and Michal Zaremba[1]

[1] Digital Enterprise Research Institute (DERI)
University of Innsbruck, Austria
{firstname.lastname}@deri.org
[2] Verizon Communications
USA
michael.brodie@verizon.com

Abstract. The researchers in DERI Innsbruck have been building an execution infrastructure for the Semantic Web Services (SWS) based on the Services Oriented Architecture (SOA) paradigm of loosely coupled components. While SOA is widely acknowledged for its potential to revolutionize the world of computing, that success depends on resolving several fundamental challenges, and especially in the case of open SOA environment the existing specifications do not address several issues. We aim in DERI Innsbruck to define a skeleton of the SWS system and implement the overall infrastructure with the aim of automating service discovery, negotiation, adaptation, composition, invocation, and monitoring as well as service interaction requiring data, protocol, and process mediation. We call this infrastructure a Semantically Enabled Service oriented Architecture (SESA). While there are already several specifications in the space for Web Services there are still elements missing, for example there is no specification describing how the particular components/services of the SWS infrastructure would work together. That work is carried out by DERI researchers in standardization bodies such as OASIS and W3C. In the near future a service-oriented world will consist of an uncountable number of services. Computation will involve services searching for services based on functional and non-functional requirements and an interoperating with those that they select. Services will not be able to interact automatically and SOAs will not scale without signification mechanization of a fixed set of components/services. Hence, machine processable semantics are critical for the next generation of computing, services and SOAs, to reach their full potential. The contribution of DERI Innsbruck is to define and implement the fixed set of services of an infrastructure that must be provided to enable a dynamic discovery, selection, mediation, invocation and inter-operation of the Semantic Web Services to facilitate the SOA revolution towards open environments. We recognize in DERI Innsburck that SOA outside of tightly controlled environment cannot succeed until/unless the semantics issues are addressed. Only with semantics can critical subtasks can be automated leaving humans to focus on higher level problems.

1 Introduction

The most important issue in todays design of software architectures is to satisfy increasing software complexity as well as new IT needs, such as the need to respond quickly to new requirements of businesses, the need to continually reduce the cost of IT or the ability to integrate legacy and new emerging business information systems. In the current IT enterprise settings, introducing a new product or service and integrating multiple services and systems present unpredicted costs, delays and difficulty. Existing IT systems consist of a patchwork of legacy products, monolithic off-shelf applications and proprietary integration. It is even todays reality that in many cases users on the spinning chairs manually re-enter data from one system to another within the same organization. The past and existing efforts in Enterprise Application Integration (EAI) dont represent successful and flexible solutions. Several studies showed that the EAI projects are lengthy and the majority of these efforts are late and over budget. It is mainly costs, proprietary solutions and tightly-coupled interfaces that make EAI expensive and inflexible.

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 SOAP protocol. The Business Process Execution Language (BPEL) allows 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 users needs, as well as resolving heterogeneity in services capabilities and interfaces will again be a lengthy and costly process. For this reason, machine processable semantics should 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.

In [3], the vision of serviceware as the next natural step beyond hardware and software is introduced: After four decades of rapid advances in computing, we are embarking on the greatest leap forward in computing that includes revolutionary changes at all levels of computing from the hardware through the middleware and infrastructure to applications and more importantly in intelligence. There we refine this towards a comprehensive framework that constitutes the bases for the technologies developed by DERI researchers, which integrates two complimentary and revolutionary technical advances, namely Service-Oriented Architectures (SOA) and Semantic Web, into a single computing architecture, that we call SESA. While SOA is widely acknowledged for its potential to revolutionize the world of computing, that success is dependent on resolving two fundamental challenges that SOA does not address, namely integration, and

search or mediation. In a service-oriented world, millions of services must be discovered and selected based on requirements, then orchestrated and adapted or integrated. SOA depends on but does not address either search or integration.

This report is structured as follows. Section 2 is providing a general summary of research goals pursued by DERI Innsbruck. We present DERI technologies in the following Section 3. Section 5 summarizes this report.

2 Objectives

Within DERI Innsbruck each of our researchers is working on a research goal. Such a research goal usually combines a research area, i.e., a major research challenges in SWS and SESA together with an implementation effort related to it. A research goal typically has a corresponding architectural component, and vice versa. A tight coupling between research goals and architectural components is desirable. The WSMX platform [5] will provide a SESA environment, which facilitates prototype development.

We distinguish 4 different types of elements of an overall SESA where each element type is composed by some sub-functionalities:

- The problem-solving layer which consists of (1) Ontologies, (2) Applications (e.g., e-tourism, e-government) and (3) Developer tools (GUI tools such as those for engineering ontology/web service descriptions; generic developer tools such as language APIs, parsers/serializers, converters, etc.).
- The broker layer which consists of (4) Discovery, (5) Adaptation (including selection and negotiation), (6) Composition (web service composition techniques such as planning), (7) Choreography, (8) Mediation ((a) Ontology mediation: techniques for combining Ontologies and for overcoming differences between Ontologies; (b) Process mediation: overcoming differences in message ordering, etc.), (9) Grounding, (10) Fault Handling (Transactionality, Compensation, etc.), and (11) Monitoring.
- The base layer that is providing the exchange formalism used by the architecture, i.e., (12) Formal languages (static ontology and behavioral, i.e., capability, choreography, orchestration languages, connection between higher-level descriptions, e.g., WSML), (13) Reasoning (techniques for reasoning over formal descriptions; LP, DL, FOL, behavioral languages, etc.) and (14) Storage and Communication.
- Finally, vertical services such as (15) Execution management and (16) Security (authentication/authorization, encryption, trust/certification).

Figure 1 presents the current status of the WSMX and SESA infrastructure.

Currently, DERI Innsbruck focuses on the following essential components to bootstrap the overall approach: (1) Ontologies, (2) Applications, (3) Developer tools, (4) Discovery, (5) Adaptation, (6) Composition, (7) Choreography, (8) Mediation, (9) Grounding, (12) Formal languages, (13) Reasoning, (14) Storage and Communication, and (15) Execution management. There are no concrete plans yet for (10) Fault Handling, (11) Monitoring, and (16) Security. Some of this work may be provided by external DERI cooperation partners.

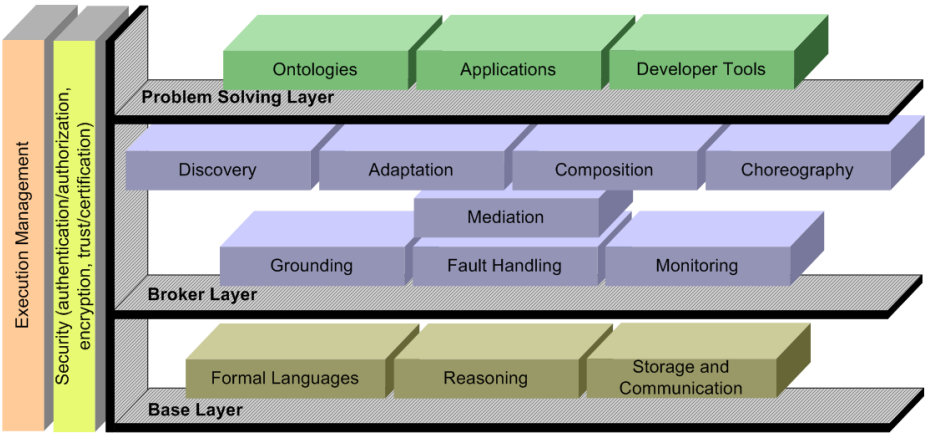

Fig. 1. SESA Infrastructure

While some of these functionalities are provided as services, the others remain the entities required to let the overall system to function, but they are not services in terms of a Service oriented Architectures. These results in SESA being called an infrastructure, not just an architecture. While the Base layers (without Formal Languages) builds SESA architecture in terms of the services, the Problem Solving layer adds the set of tools and entities, which causes that SESA becomes a complete Semantic Web Services oriented infrastructure.

SOAs typically consist of a set of services, and a coordinator that combines the services and puts them to use. Talking about SOA in the context of SESA can some-

times be misleading since SESA is a SOA and at the same time it is the coordinator of another, larger SOA. The SESA differentiates two types of services: platform services (such as Discovery, Choreography, Data and Process mediation etc.) and user services (e.g. back-end applications services). Platform services are necessary to enable the infrastructure to deliver its functionality as defined by its execution semantics. User services are exposed by information system external to the SESA infrastructure, but they are still coordinated using SESA platform services. The SESA recommendation defines the scope of particular platform services in terms of their functionality, while it remains silent about the scope and the functionality of user services.

The SESA infrastructure consists of several decoupled services. This enables independent refinement of these services, so each of them can have its own architecture, without hindering the overall SESA infrastructure. Following the SOA design principles, the SESA infrastructure separates the concerns of the individual components thereby separating the service descriptions and their interfaces from the implementation. This adds flexibility and scalability for upgrading or replacing the implementation of the services provided by the components that adhere to the required interfaces.

The SESA recognizes **vertical** and **horizontal** services. Vertical services remain invisible to horizontal services, and during its execution, the horizontal services remain unaware that vertical services are executed. This type of vertical service is provided through the inversion of control.

3 Technologies/SESA Services

As presented in section 2, we distinguish four different types of elements of an overall SESA where each element type is composed by some sub functionalities:

- The problem-solving layer,
- The broker layer,
- The base layer,
- The vertical layer

This section describes in more detail functional components that play a role in the SESA architecture.

3.1 The Problem-Solving Layer

The problem-solving layer consists of Ontologies, Applications (e.g., e-tourism, e-government) and (3) Developer tools (GUI tools such as those for engineering ontology/web service descriptions; generic developer tools such as language APIs, parsers/serializers, converters, etc.).

Ontologies (chair: Martin Hepp) In this research topic, the working group advances the state of the art in the use of ontologies for advancement of the automation of business processes [7, 13, 12]. Ontologies in our understanding are community contracts about a representation of a domain of discourse. Representation in here includes (1) formal parts

that can be used for machine reasoning, and (2) informal parts like natural language descriptions and multimedia elements that help humans establish, maintain, and renew consensus about the meaning of concepts.

Our contributions address the following two maindimensions of using ontologies for real application:

1. *Maturing Semantic Web foundations*, so that they become compatible with the real world complexity and scale. This includes three main branches of research:
   - *Ontology Engineering* Methodologies for and prototypes of industry-strength business ontologies, e.g. the gen/tax methodology for deriving ontologies from existing hierarchical standards and taxonomies (UNSPSC, eCl@ss, ...) and eClassOWL, the first serious attempt of building an ontology for e-business applications; and in general advancing the state of the art in e-business data and knowledge engineering, including metrics for content.
   - *Community-driven Ontology Building* For quite a while, we have been trying to hand back control over the evolution of ontologiesto the user community, including semi-automated approaches and OntoWiki, a Wiki-centric ontology building environment. In this segment also fall quantitative comparisonsof community-centric and engineering-based ontology building.
   - *Economic Aspects of Ontology Building and Usage*
     Building ontologies consumes resources, and in an economic setting, these resources are justified and will be spend (by rational economic actors, at least) only if the effort needed to establish and keep alive a consensual representation of a domain of discourse is outweighed by the business gain, either in terms of cost, added value, or strategic dimensions, e.g. process agility. This research branch is rather young and underdeveloped, but an important piece of understanding and fueling the use of ontologies in business applications.
2. *Building actual ontologies for core challenges of Information Systems* in order to realize and evaluate the business benefit, and to identify the open research challenges. We currently focus on five specific application domains:
   - Semantics-supported Business Process Management, i.e. the idea to mechanize Business Process Management by using Semantic Web techniques and especially Semantic Web Services. There is a first vision paper and a Working Group being founded.
   - Semantic Web services, especially WSMO/WSML/WSMX, i.e. the use of ontologies and related technology for the automation of Web services discovery, composition, execution, and monitoring.
   - Electronic Markets and Electronic Procurement, including a reference framework for ontology-supported electronic procurement and an analysis of the true complexity of business matchmaking.
   - eTourism, e.g. the automation of the discovery and booking of tourism offerings.
   - Financial reporting, e.g. the automated mediation between financial 2data (e.g. XBRL data) so that balance sheets and other documents from multiple sources can be integrated on the fly.

Applications (chair: Michal Zaremba) The mission of the Application working group is to develop a common understanding of various technologies intended to facilitate the use of other services of SESA. This working group develops (1) use case scenarios that help validate the real-world fitness of SESA components and (2) domain-specific implementations which will be used for testing of SESA services.

Semantic Web Services challenge[3] has been the first attempt to provide a test-bed for the future Application working group [25]. The goal of the SWS Challenge is to develop a common understanding of various technologies intended to facilitate the automation of mediation, choreography and discovery for Web Services using semantic annotations. The challenge explore the trade-offs among existing approaches. Additionally we would like to figure out which parts of the problem space may not yet be covered. The challenge aims to provide a forum for discussion based on a common application. This Challenge seeks participation from industry and academic researchers developing software components and/or intelligent agents that have the ability to automate mediation, choreography and discovery processes between Web services.

Developer Tools (chair: Mick Kerrigan) The mission of the Developer Tools working group is to produce high quality tools related to Semantic Web Services that can be used by users of all competency levels. To this end DERI Innsbruck provide a large number of tools that can be used by users with different skill sets. Members of the working group are working on tools for managing WSMO ontologies, web services, goals and mediators, for creating mappings between WSMO ontologies for runtime mediation, for executing WSDL web services and managing WSMO execution environments.

The developer tools implemented with DERI are broken down into a number of plug-ins for Eclipse. The plug-ins are bundled together as two different products, namely the Web Services Modeling Toolkit (WSMT) and the DERI Ontology Management Environment (DOME). The WSMT [18, 17] is aimed at covering all the functionality of WSMO, WSML and WSMX. With the WSMT the user is able to create and manage WSMO ontologies, web services, goals and mediators through the WSML human readable syntax, create mappings between two WSML ontologies for the purposes of instance transformation and thirdly manage and interact with the WSMX environment. The primary focus of DOME [11] is the use of WSMO as an ontology language and thus focuses only on the ontology and mediator parts of the WSMO specification. Users of DOME can create and manage their WSMO ontologies and mediators through the WSML human readable, XML and RDF syntaxes. They can also create mappings between two ontologies that can be used later by an execution environment.

3.2 The Broker Layer

The broker layer consists of Discovery, Adaptation (including selection and negotiation), Composition (web service composition techniques such as planning), Choreography, Mediation ((a) Ontology mediation: techniques for combining Ontologies and for

[3] http://www.sws-challenge.org

overcoming differences between Ontologies; (b) Process mediation: overcoming differences in message ordering, etc.), Grounding, Fault Handling (Transactionality, Compensation, etc.), and Monitoring. There are no concrete plans yet for Fault Handling and Monitoring.

Discovery (chair: Holger Lausen) The Discovery working group develops different discovery implementations that are compatible with WSMO, WSML and specifically WSMX. The scope of the working group is to develop solutions based on existing descriptions (WSDL, UDDI, and text) as well as on more advanced descriptions based on semantic annotations using WSML [16, 8]. Some tasks in this component are the development of a discovery engine based on keywords and existing annotations (WSDL), extended beyond for example WSDL description to related documentation, interpretation of, the semantic descriptions.

The working group has a parallel approach with respect to implementation of the component. On the one hand we implement some basic infrastructure including key word search on existing service documentation and on the other hand we will evaluate the different existing implementation to assess how they can be extended and integrated in a more common framework.

The working group develops a discovery engine based on keywords and existing annotations (WSDL, HTML docs, etc). The data set we will operate on will come from publicly available Web service descriptions. Initially this has been limited to the information that can be obtained from the WSDL files. A search request can be expressed using keywords or advanced template search that allows to query for specific operation names or similar. WSDL documents can also be retrieved by URL. This phase will also include basic monitoring functionality for determining if the service specified in the given WSDL document is available.

Besides the pure WSDL description the working group extends the search to related documentation. This related information can be extracted from UDDI (taxonomies, specific tmodels, etc) or relevant web pages. Based on the initial work of the WSRD group in DERI Galway some information retrieval techniques will be applied to the keywords found in the WSDL file corpus.

The working group has also started interpreting the semantic descriptions, where the search expressivity increases with the expressivity of the underlying ontology language. The working group will be open to various approaches, at the beginning those approaches might not be compatible (i.e. annotations using language/model A will not necessarily work with annotations using language/model B). However the long-term goal is to make them interoperable or merge them.

Adaptation (chair: Ioan Toma) After discovering a set of potentially useful services, a SESA needs to check whether the services can actually fulfill the users concrete goal and under what conditions. Those that cannot fulfill the goal are removed from the list of discovered services. This step is required as it is not feasible for a service to provide an exhaustive semantic description. Giving the Amazon bookstore service as an example, it is not feasible for Amazon to update the semantic description of their Web service every time a new book is available or an existing book is changed, therefore

we must whether that Amazon actually currently has a copy of the book requested by the user, and at an acceptable price. The process of checking whether and under what conditions a service can fulfill a concrete Goal is called negotiation in SESA, and it also encompasses by so-called filtering. By filtering we understand the process of narrowing the set of discovered services which provide the functionality requested, by considering only the services that have the appropriate non-functional properties requested by the user. Furthermore building a ranking/order relation based on non-functional properties criteria like prince, availability, etc. is also part of the filtering process [29].

Once a list of Web services than can fulfil the user's concrete goal is prepared, a SESA must then choose one of the services to invoke. It is important that this selection is tailored to the user's needs, as for example while one user may require high quality another may prefer low price. This process is called selection. Negotiation, filtering and selection are tasks of the Adaptation working group.

Composition (chair: Jörg Hoffmann) The Composition working group develops methods to do Web Service composition (WSC), starting from web service descriptions at various levels of abstraction, specifically, the functional level and process level components of WSMO. Such methods are implemented as tools in the relevant contexts, in particular WSMX. Potential applications of WSC technology are researched, and modelled using WSMO/WSML; case studies are run with the developed tools, ultimately resulting in technology export.

In more detail, the working group revolves around the following topics:

- Language subset/capability extensions. The working group deals with as large as possible subsets of WSMO/WSML. Naturally, the developed technology has started with restricted language subsets, and incrementally moves on to richer subsets. If new features/scenarios become relevant on the side of WSMO/WSML, these will become new targets for WSC.
- Applications, case studies, benchmarking, technology export [15]. A vital ingredient to WSC research is to stay as close as possible to the envisioned fields of commercial/industrial application. The working group uses and strengthens DERI's contacts in this respect. Possible areas of application shall be identified, and increasingly realistic scenarios shall be modelled. These models play a crucial role in evaluating the developed WSC techniques, and thus guiding the research into which kinds of methods will work and which will not. The case studies may eventually lead to fostered collaborations and, ultimately, to technology export.
- Addressing efficiency problems [14]. Since WSC is a notoriously hard problem it is PSPACE-complete even in extremely simple formalisms it is essential to develop heuristic techniques that have the potential to scale satisfyingly in practical instances of the WSC problem.
- The working group expects that, eventually, notions of optimality will become relevant for WSC: What is the best service satisfying the composition task, and how can we compose that service? We intend to contribute to both the development of such notions and to their algorithmic treatment.

Choreography (chair: James Scicluna) The Choreography part of SESA is meant to provide a process language which should allow for formal specifications of interactions and processes between the service providers and clients, define reasoning tasks that should be performed using this language, and implement an engine to support the execution of interactions, as well as to support reasoning in this language.

The model for WSMO Choreographies is currently stable [27]. It is inspired by the Abstract State Machines (ASM) methodology and inherits the core principles such as the state, transition rules and flexibility to model any kind of behavior. The syntax of the choreography language has been defined as a result of the model. It is similar to the ASM language with some obvious constructs that have been introduced in order for it to fit with the WSML language. The semantics are defined using a set-based approach and describe the operational behavior of choreographies on the same lines as for ASMs.

The work of the Choreography API has been divided in different parts, namely, the API (i.e. the interfaces), the implementation, the parser and the serializer. The API defines the interfaces and methods (with no implementation) for the objects within the language constructs. The implementation part implements the interfaces so that a user can easily create and manage the language constructs. The parser loads up an object model representation in the memory from a choreography description in a WSML file. The serializer, performs the reverse operation, that is, it saves the memory representation of the language to the equivalent syntax representation in a WSML file. All of these modules have been completed.

The main steps involved in the implementation of the choreography engine are the design with particular emphasis on the interaction with other WSMX components and the actual programming. Both of these aspects are in a stable condition but eventually they evolve as WSMX gets better and as requirements change. Particularly, we will work further towards a more expressive and intuitive language to deal with problems related to choreography reasoning with special emphasis on web service compatibility [23]. This language will be as a layer on top of the ASM methodology and hence compliant with such a formalism. Finally, this language will be able to express the existing interaction and workflow patterns which capture the major possible use cases in business processes and service interactions.

Mediation (chair: Adrian Mocan) Mediation in SESA aims at providing flexible mediation service at both data and process level. Data Mediation provides automatic data transformation from the format used by the source party to the format required by the target party involved in conversation, while Process Mediation is concerned with the heterogeneity of the public processes of the participants in a conversation.

- Data Mediation provides automatic data transformation from the ontology used by the source party to the ontology required by the target party involved in conversation [24]. As WSMX is a semantic enabled service execution environment, we assume that the data to be mediated is semantically described, i.e. it consists of ontology instances. As a consequence the WSMX Data Mediation Service has to support instance transformation from terms of one ontology to the terms of another ontology, based on the set of already created mappings between the two given ontologies.

– The Process Mediator service has the task of solving the communication (behavioral) mismatches that may occur during the communication between a requestor and a provider of a service [4]. As in WSMO, the requestor is a WSMO Goal, while the provider is a Semantic Web Service, the Process Mediators task is be to accommodate the mismatches between the goals requestedChoreography and the SWSs choreography.

Grounding (chair: Jacek Kopecky) Apart from discovering web services and composing them, a SESA also needs to communicate with the Web services send the necessary request messages and receive the responses. Because internal communication within the SESA uses semantic data and practically all currently deployed Web services use their specific XML formats, the External Communication component needs to translate between the involved data forms. This translation is also known as data grounding [20]. Above that, this SESA also needs to support concrete network protocols (HTTP, SOAP, other bindings) to be able to exchange messages with the Web service.

As grounding has to be based on the Web Services Description Language, the work on this component also contains W3C efforts towards Semantic Web Services. In particular, this means the WSDL RDF mapping [19] from Web Service Description WG, and the Semantic Annotations for WSDL [22] in the SA-WSDL WG at W3C.

3.3 The Base Layer

The base layer provides the exchange formalism used by the architecture, i.e., Formal languages, Reasoning (techniques for reasoning over formal descriptions; LP, DL, FOL, behavioral languages, etc.) and Storage and Communication.

Formal Languages (chair: Jos de Bruijn) Descriptions in a Semantically-Enabled Service Oriented Architecture (SESA) need different formal languages for the specification of different aspects of knowledge and services [16, 26, 6]. The descriptions in a SESA can be decomposed into four dimensions:

– Static knowledge (ontologies)
– Functional description (capabilities)
– Behavioural description (choreography and orchestration)
– Non-functional Properties

Tasks for this working group include the integration of FOL-based and nonmonotonic LP-based languages, the explicitization of context for use with scoped negation, and the development of rules for the, Semantic Web (through the W3C RIF working group). Furthermore, requirements on the functional descriptions of services and as well as a semantics for web service functionality need to be devised. Requirements need to be gathered on the description of a choreography and an orchestration and a semantics needs to be devised. Finally, purpose and usage of non-functional requirements will be investigated. Future work of this working group will focus on:

- Static knowledge integrating knowledge based on classical first-order logic and nonmonotonic logic programming; important issues are the representational adequacy of the integration, as well as decidable subsets and a proof theory, so that reasoning becomes possible; scoped default negation; rules for the Semantic Web RIF working group; connection between Semantic Web languages RDF, OWL.
- Functional description requirements need to be gathered on the functional specification of services and a semantics needs to be devised which can be combined with the language for the description of ontologies, in order to enable the use of ontologies for the description of web service functionality. An important use case for the functional description of services is discovery. Therefore, it is expected that many requirements on the functional description of services will come from the discovery research goal.
- Behavioural description there exist several formal languages which are suitable for behavioural description. Examples are transaction logic, situation calculus, and action languages. Requirements need to be gathered on the description of choreography and an orchestration and semantics needs to be devised. A key challenge is the combination of this language with ontology languages in order to enable the reuse of ontology vocabulary in the choreography and orchestration descriptions. Finally, this language needs to be connected to the language for capability description in order to prove certain correspondences between the functional and behavioural description of services.
- Non-functional Properties Non-functional properties can at least be divided into two categories: (1) meta-data, e.g., author, description, etc., of the WSML statements in a description and (2) actual non-functional properties, i.e., actual properties of services (e.g. pricing, QoS, transactions). NFPs require a deeper investigation into their purpose and their usage.

The work on Formal Languages establishes the theoretical foundations for the WSML family of languages, used for the description of Web Services.

Reasoning (chairs: Darko Anicic and Stijn Heymans) The Reasoning working group develops an efficient and extensible reasoning engine for expressive rule-based languages (WSML Core/Flight/Rule), as well as description logic based languages (WSML-DL) [1, 2]. The reasoner is based on state-of-the-art reasoning algorithms (for query answering, logical entailment, etc.). The SESA needs the reasoning component for service discovery as well as both process and data mediation. Mission critical features of the Reasoning component are: hybrid reasoning based on DLs and logic programming, reasoning with very large instance bases, reasoning with heterogeneous and conflicting information, and reasoning in distributed environments. Also one of the major objectives of this working group is the implementation of a Rule Interchange Format (RIF)[4]. RIF aims to specify a common format for rules in order to allow rule interchange between diverse rule systems. This format (or language) will function as an interlingua into which rule languages can be mapped, allowing rules written in different languages to be executed in the same reasoner engine. The RIF layer of our reasoner engine will

[4] http://www.w3.org/2005/rules/

be capable of handling rules from diverse rule systems and will make WSML rule sets interchangeable with rule sets written in other languages that are also supported by RIF.

Storage and Communication (chair: Reto Krummenacher) The storage components, plural on purpose, shall provide repositories to store objects needed to ensure successful processing of user request to SESA. There might be a need for different storages tailored to the particular needs: web service descriptions, goals, mediation rules, workflows, and execution semantics. It is already known that the Execution Management component requires repositories for ontologies and data instances (service descriptions in particular). The idea is to use a Triple Space infrastructure to do so. The mission of the Storage Component team is thus to find out which means of storage are required and in what way these requirements can be fulfilled in the easiest and simplest way to provide optimal service to the application layer components and the vertical services [28, 21, 9].

3.4 Vertical Services

Vertical services consist of Execution management and Security (authentication and authorization, encryption, trust/certification). There are no concrete plans yet for Security.

Execution Management (chair: Thomas Haselwanter) The Execution Management working group is responsible for the management of WSMX as a platform and for the coordination of the individual components [30, 10]. As the kernel of the system it enables and realizes the overall operational semantics of WSMX that let the system achieve the promised functional semantics of its client-side interface. It takes the functionality offered by the individual components of the framework and orchestrates these atomic pieces into a coherent whole in an orderly, and consistent fashion. These properties are guaranteed by the execution semantics, which are executed over the set of services that are available to the execution management component

4 Technical Task Force

A comprehensive framework as described in the previous section can only be effective if the different components are aware of the global vision as well as communicate among each other to accomplish this joint vision. Chaired by Mick Kerrigan, the mission of the Technical Task Force is to oversee the implementation efforts within each of the DERI objectives to ensure that the different prototypes are interoperable. The Technical Task Force aims to improve communication between the different objectives to ensure transparency and understanding of current development efforts. This will be achieved by identifying dependancies between working groups, sharing the requirements one working group has on another and aiding a working group in prioritizing certain implementation efforts based upon the needs of other groups. The Technical Task Force meets on a monthly basis to discuss the current status of the implementation efforts across the objectives and devise plans to bring these prototypes into a coherent architecture.

5 Summary

This paper outlined a comprehensive framework that integrates two complimentary and revolutionary technical advances, Service-Oriented Architectures (SOA) and Semantic Web, into a single computing architecture, that we call Semantically Enabled Service oriented Architecture (SESA) and details of how these technologies are developed within DERI Innsbruck have been provided. While SOA is widely acknowledged for its potential to revolutionize the world of computing, this success is dependent on resolving two fundamental challenges that SOA does not address, namely integration, and search or mediation. In a service-oriented world, millions of services must be discovered and selected based on requirements, then orchestrated and adapted or integrated. SOA depends on but does not address either search or integration. The contribution of DERI Innsbruck is to provide the semantics-based solutions to search and integration that will enable the SOA revolution. The paper provides a vision of the future, enabled by SESA, that places computing and programming at the services layer and places the real goal of computing, problem solving, in the hands of end users.

Acknowledgements

The work is funded by the European Commission under the projects ASG, DIP, enIRaF, InfraWebs, Knowledge Web, Musing, Salero, SEKT, SEEMP, SemanticGOV, Super, SWING and TripCom; by Science Foundation Ireland under the DERI-Lion Grant No.SFI/02/CE1/I13 ; by the FFG (Österreichische Forschungsfrderungsgeselleschaft mbH) under the projects Grisino, RW^2, SemNetMan, SEnSE, TSC and OnTourism.

References

1. S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases*. Addison-Wesley, 1995.
2. C. Beeri and R. Ramakrishnan. On the power of magic. In *PODS '87: Proceedings of the sixth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, pages 269–284, New York, NY, USA, 1987. ACM Press.
3. M. Brodie, C. Bussler, J. de Bruijn, T. Fahringer, D. Fensel, M. Hepp, H. Lausen, D. Roman, T. Strang, H. Werthner, and M. Zaremba. Semantically enabled service oriented architectures: A manifesto and a paradigm shift in computer science. Technical report, DERI Innsbruck, 2006.
4. E. Cimpian, A. Mocan, and M. Stollberg. Mediation enabled semantic web services usage. In *1st Asian Semantic Web Conference (ASWC2006)*, 2006.
5. E. Cimpian, M. Moran, E. Oren, T. Vitvar, and M. Zaremba. D13.0 overview and scope of wsmx.
6. Jos de Bruijn, Holger Lausen, Axel Polleres, and Dieter Fensel. The web service modeling language: An overview. In *Proceedings of the 3rd European Semantic Web Conference (ESWC2006)*, number 4011 in Lecture Notes in Computer Science, pages 590–604. Springer-Verlag, 2006.
7. D. Fensel. *Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce*. Springer-Verlag, 2004.

8. Dieter Fensel, Uwe Keller, Holger Lausen, Axel Polleres, and Ioan Toma. What is wrong with web services discovery. In *W3C Workshop on Frameworks for Semantics in Web Services*, 2005.
9. Andreas Harth and Stefan Decker. Optimized index structures for querying rdf from the web. In *In Proc. 3rd Latin American Web Congress (LA-WEB 2005)*, 2005.
10. Thomas Haselwanter, Maciej Zaremba, and Michal Zaremba. Enabling components management and dynamic execution semantic in wsmx. In *In Proceedings of the WIW 2005 Workshop on WSMO Implementations*, 2005.
11. Jan Henke. The table metaphor: A representation of a class and its instances. In *Proceedings to Workshop on User Aspects of the Semantic Web (ESWC)*, 2005.
12. M. Hepp. Representing the hierarchy of industrial taxonomies in owl: The gen/tax approach. In *ISWC Workshop Semantic Web Case Studies and Best Practices for eBusiness (SWCASE05)*, 2005.
13. M. Hepp. Products and services ontologies: A methodology for deriving owl ontologies from industrial categorization standards. *Int'l Journal on Semantic Web and Information Systems (IJSWIS)*, pages 72–99, 2006.
14. Jörg Hoffmann and Ronen Brafman. Conformant planning via heuristic forward search: A new approach. 170(6–7):507–541, 2006.
15. Jörg Hoffmann, Stefan Edelkamp, Sylvie Thíebaux, Roman Englert, Frederico Liporace, and Sebastian Trüg. Engineering benchmarks for planning: the domains used in the deterministic part of ipc-4. 26:453–541, 2006.
16. Uwe Keller, Holger Lausen, and Michael Stollberg. On the semantics of functional descriptions of web services. In *Proceedings of the 3rd European Semantic Web Conference (ESWC2006)*, Budva, Montenegro, June 2006. Springer-Verlag.
17. Mick Kerrigan. The wsml editor plug-in to the web services modeling toolkit. In *Proceedings of the 2nd WSMO Implementation Workshop (WIW)*, 2005.
18. Mick Kerrigan. WSMOViz: An Ontology Visualization Approach for WSMO. In *Proceedings of the 10th International Conference on Information Visualization*, Jul 2006.
19. Jacek Kopecky and Bijan Parsia (editors). Web services description language (wsdl) version 2.0: Rdf mapping. Technical report.
20. Jacek Kopecky, Dumitru Roman, Matthew Moran, and Dieter Fensel. Semantic web services grounding. In *In Proc. of the Int'l Conference on Internet and Web Applications and Services (ICIW'06)*, 2006.
21. Reto Krummenacher, Martin Hepp, Axel Polleres, Christoph Bussler, and Dieter Fensel. Www or what is wrong with web services. In *In Proc. of the 2005 IEEE European Conf on Web Services (ECOWS 2005)*, 2005.
22. Holger Lausen and Joel Farrell (editors). Semantic annotations for wsdl. Technical report.
23. A. Martens. On Compatibility of Web Services. *Petri Net Newsletter*, 65:12–20, 2003.
24. Adrian Mocan, Emilia Cimpian, and Mick Kerrigan. Formal Model for Ontology Mapping Creation. In *Proceedings of the 5th International Semantic Web Conference (ISWC 2006)*, November 2005.
25. Charles Petrie. It's the programming, stupid. *IEEE Internet Computing, "Peer to Peer"*, 2006.
26. Axel Polleres, Cristina Feier, and Andreas Harth. Rules with contextually scoped negation. In *Proceedings of the 3rd European Semantic Web Conference (ESWC2006)*, number 4011 in Lecture Notes in Computer Science. Springer-Verlag, 2006.
27. D. Roman and J. Scicluna. D14 ontology based choreography of wsmo services. Technical report, 2006.
28. Omair Shafiq, Reto Krummenacher, Francisco Martin-Recuerda, Ying Ding, and Dieter Fensel. Triple space computing middleware for semantic web services. In *In Proc. 2006*

Middleware for Web Services (MWS 2006) Workshop at the 10th Int'l IEEE Enterprise Computing Conference (EDOC 2006), 2006.
29. I. Toma and D. Foxvog. D28.4 non-functional properties in web services. Technical report, 2006.
30. Maciej Zaremba, Matthew Moran, and Thomas Haselwanter. Applying semantic web services to virtual travel agency case study. In *In Proceedings of the 1st Workshop: SemWiki2006 - From Wiki to Semantics, co-located with the 3rd Annual European Semantic Web Conference (ESWC 2006)*, 2006.