Enabling Semantic Web Services for Telematics

Josef Noll[1,2], Erik Lillevold[1], Marianne Rustad[1]
[1]UniK, N-2027 Kjeller, Norway, [2]Telenor R&D, N-1331 Fornebu, Norway
josef@unik.no, erlille@unik.no, marianne@marekatt.nett

Abstract—
Several standards for Semantic Web Services (SWS), supported by tools and methodologies are under development. The Adaptive Grid Services (ASG) project develops a semantic Web service platform in order to fulfil the needs of Telematics, Telecommunications, eGovernment and Enterprise IT. Platform development is well under way, supporting semantic specification, registration, discovery, composition and enactment of composed and atomic services.
Business in SWS delivery will be based on extensions of existing infrastructure, thus a roadmap for evolution is of crucial importance for business development. The roadmap covers the major items: standardisation, platform development, service scenario, and SWS business. Based on an example in the Telematics domain, this paper shows that prototype development is well under way, and tools are expected to reach the developer market within mid 2006. More crucial are domain ontologies, standardisation, and the service inclusion through existing platforms, expected to happen until 2008. Substantial market penetration will then follow.

Introduction

To develop and maintain services for the future that are both attractive, easy to use and cheap enough developers have realized that new methodologies, techniques and tools are necessary. Based on these facts concepts and technologies like Service Oriented Architectures (SOA), Web Services (WS), Semantic Web (SW) and Semantic Web Services (SWS) have gradually grown up to show their viability, especially if they are used in combination.

Ongoing work in standardization bodies (e.g. W3C and OASIS) and research forums (e.g. EU IST FP 6) supports these developments. The ASG (Adaptive Service Grid) project is a typical example of an EU project that has as its goal to utilizing concepts and technologies from SOA, WS, SW and SWS and demonstrates their maturity by developing services within selected business domains. All major software vendors have already adopted Web Service (WS) technology, i.e. WSDL, SOAP and UDDI to be a corner-stone in their future tools for service creation. They move to SWS to meet the requirements of future enterprise systems; to create and execute in near real-time and cope with context awareness, personalization and mobility [1]. SWS adds extra semantics to the service descriptions enabling WS to work together more flexible, intelligent and automatic.

This paper starts with the usage requirements for advanced service platforms, established through real-life business scenarios. It then provides the various approaches when bringing Semantics to Web Services. Focus is here on the difference between the selected approaches WSDL-S, OWL-S and WSMO, as seen from the service provider point of view. The following section will then explain how external services can be brought into the Adaptive Services Grid (ASG) platform. The advantages of the platform are outlined as compared to the OASIS reference model (OASIS RM). Based on a Telematics example, we will then outline how real-world services can be enhanced by ontologies and mapped to the platform. The paper will conclude with the roadmap for the delivery of semantic web services, indicating estimates for time-to-market of standardization, tool development and end-customer services.

Scenarios and Requirements

In order to establish requirements for future services platforms four scenarios were selected, and each of them used to validate the suitability of an ASG based service execution [2]. The scenarios cover:

- Telematics, commuter and tourist use cases
- Dynamic ASP supply chain (as part of a Telecom scenario)
- Service Monitoring (as part of an Enterprise scenario), and
- Document discovery (as part of an e-Government scenario).

The analysis of all scenarios led to seven business requirements for service delivery:

Reuse of existing functionality, e.g. services and infrastructure, makes use cases cost-efficient to realise.

Standards and Reliability are essential for industry to adopt solutions.

Openness will allow integration of additional services with as little changes as possible.

Adaptivity to current environmental constraints, e.g. user preferences and user connectivity is key for user acceptance of new services.

Dynamic and transparent service composition is required to adapt to the specific service requests.

QoS awareness handles specific user requests, e.g. budget or time constraints.

Semantic Awareness is crucial for understanding the user request, service discovery and service composition.

These requirements are taken into account when defining the semantic web services platform, used later on to establish real-world services. Dynamic supply chain and Telematics scenario are excellent candidates for service demonstration, because they deal with a dynamic service world, offering a wide range of external services to an end-customer. This paper will use the Telematics scenario, as it is familiar to the public reader.

Telematics Scenario

Originally, the term Telematics derived from an integration of the terms Telecommunication and Informatics, today more often known as ICT (Information and Communications Technology). The term Telematics is today more commonly narrowed down to mean Intelligent Transportation System (ITS), i.e. an ICT system dealing intelligently with everything related to transportation such as vehicles, roads, public transportation, travel information, traffic management, and traffic emergency.

Our Telematics scenario presents a location based mobile service within transportation. The goal of Telematics services is to provide customers with any information that might be of interest while commuting to work or visiting an unfamiliar area, for example as a tourist. In the commuting situation, the goal is to improve travel efficiency by the support of route planning and rescheduling based on information on the traffic situation, road conditions, public communication, and parking area locations. These services are also useful in a tourist situation, but with the additional goal of improving the experience of the visited area. This can be information about the cause of a traffic incident, estimated delay, alternative routs, etc. Further information can be about points of interest (POIs), such as opening hours and route description.

The customer coordinates combined with information about the user’s route, the traffic conditions on alternative routs, traffic forecasts based on traffic history, user preferences, public transportation, etc. can then be utilized to sort out the relevant incidents in the traffic and propose best-effort action for the user. The Traffic Incident Service alerts the traveler about major incidents that is likely to influence the planned journey, using the customer’s coordinates, direction and route information, in combination with the coordinates and expected delay caused by the traffic incident.

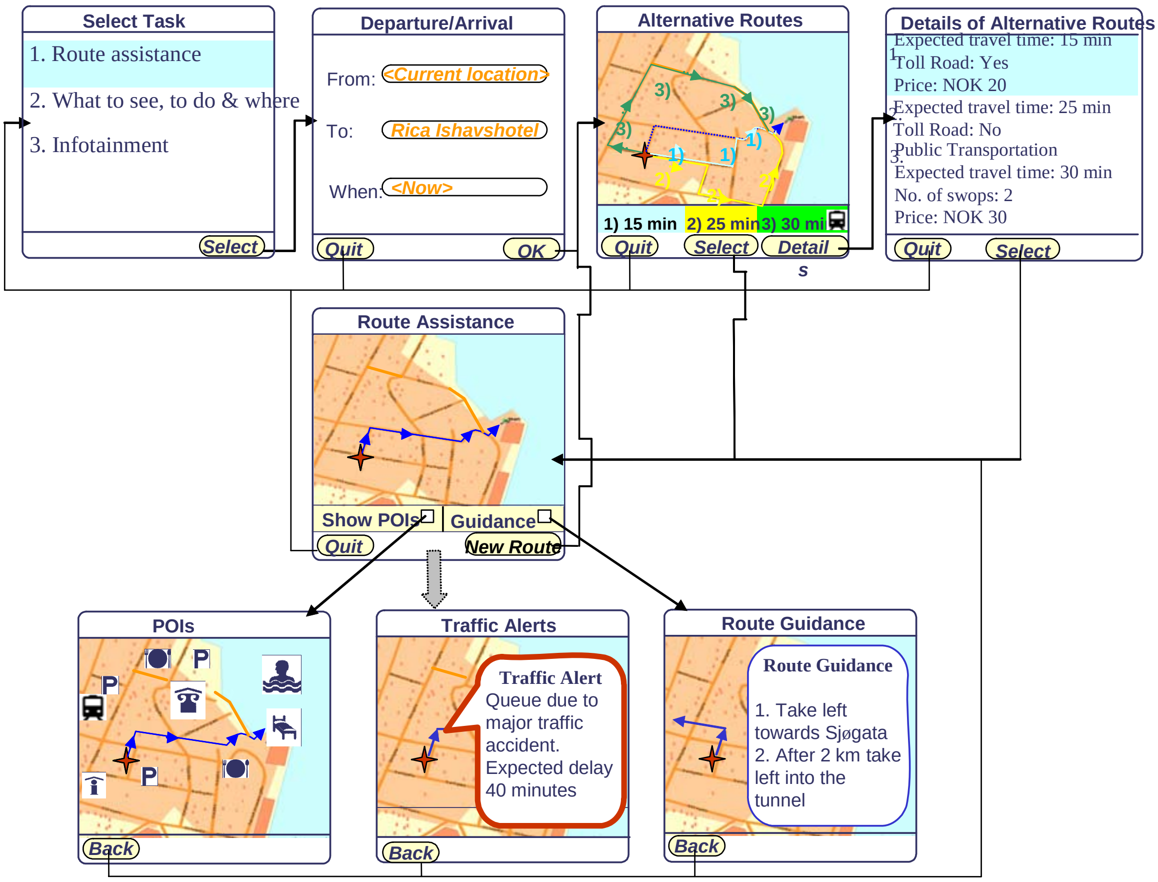

Figure 1 - Telematics application flow, demonstrating complexity of end-user service delivery

The application flow of Figure 1 shows the complexity of the delivery system, where the service broker has to connect various sources of information into a dynamic delivery system. Examples of Telematics services have been demonstrated on various occasions. However, they have not reached the end-customer as a regular service, as the service broker can't maintain and update the services on an economically acceptable way. This is the reason to move to a semantic web services platform, described in the following sections.

Service Oriented Architecture (SOA)

The history of evolution of SOA has its roots back to the 1980's when the Object Oriented paradigm emerged and early 1990's when the first commercial usage of the Web started. Distributed component models and commercial component based architectures (Corba) paved the way for the Web Service in the late 1990's [3].

A SOA Framework

SOA as a vision evolved well, but different implementations hampered the applicability. To avoid those problems the Organization for the Advancement of Structured Information Standards (OASIS) established a framework and drafted a Reference Model (RM) that a system has to adopt in order to claim compliance with the OASIS SOA specifications [4]. According to the OASIS framework SOA is an architectural paradigm (model) that does not necessarily mean usage of Web Services although Web Service is a popular implementation. The SOA RM should capture core principles and axioms of SOA and be used as a template for the SOA architecture.

Web Service (WS) vs. Semantic Web Service (SWS)

With WS we move from the traditional human-centric to the application-centric Web, i.e. that web services do not only support human beings, but also directly application programs that use them to make new end-user services. The Web Service Description Language (WSDL) specification began with WSDL 1.0 in 2000. Enhancements of this standards, WSDL 1.2, has been renamed WSDL 2.0, as it includes some major changes. Draft releases of the WSDL 2.0 standard have been available since March of 2004. A central concept here is the Web Service Description Language (WSDL), standardised by W3C and used to describe the service based on XML. It defines and locates a web service by using the following elements:
< portType > to define the operations performed.
< message > for defining the messages used.
< types > to define the data types used
< binding > element define the communication protocols used by the web service.
This service description is registered in a Service Registry by the Service Provider and read from there by a potential Service Requester. The Service Registry and the transactions, Publish and Find, are specified by UDDI (Universal Description, Discovery, and Integration).

The ultimate vision of Web Service was that it is possible to automate co-operation between and integration of web applications in the long term. However, even with a lot of efforts in industry and standardisation bodies we are still quite away from having automated the interactions among Internet applications. Currently, none of the Web Service Standards have the capabilities to reach the goals. Thus the concept of SWS has evolved and enhanced the formal description of Web Services through semantic added to the description of the service [5].

The major research contributions to the SWS standards are Web Ontology Language for Services (OWL-S), Web Service Modeling Ontology (WSMO), Web Services Semantics (WSDL-S), and the Semantic Web Services Framework (SWSF). They are complementing each other. The paper will provide a short overview over the standards, concentrating on the advantages of each of them.

Semantic Web Service (SWS)

There are two major approaches in how to extend WS to SWS. Most approaches like OWL-S, WSMO and SWSF create a SWS description language, and then map it to the Web service. WSDL-S on the other hand annotates directly the WSDL file and defines semantics as part of the file.

The terminology used in SWS is recalled here: **XML** provides a syntax for structured documents, but imposes no semantic constraints on the meaning of these documents. **XML Schema** is a language for restricting the structure of XML documents and also extends XML with datatypes, which might be seen as a first step to add Semantics. **RDF** is a datamodel for objects ("resources") and relations between them, provides a simple semantics for this datamodel, and these datamodels can be represented in XML syntax. **RDF Schema** is a vocabulary for describing properties and classes of RDF resources, with a semantics for generalization-hierarchies of such properties and classes [OWL2004].

Web Ontology Language for Services (OWL-S)

OWL-S[1], former DAML-S is an OWL-based Web Service Ontology [6], with first specifications names *DAML-S 0.5* back in 2001. It provides a core set of markup language constructs for describing properties and capabilities of their Web services in a computer-interpretable form. It is the first well-researched Web Service Ontology, and as such the reference for all further developments.

OWL-S provides a semantic view of the service through a profile. The link to WS or other type of service is called *grounding*. Figure 2 provides the formal mapping between the OWL-S described ontology and the WSDL described Web Service. It subdivides between the operation/process of

[1] DAML Services, http://www.daml.org/services/owl-s/

the service, and the input/output from the services. The process is described through a process model, while the input/output is described through types [7].

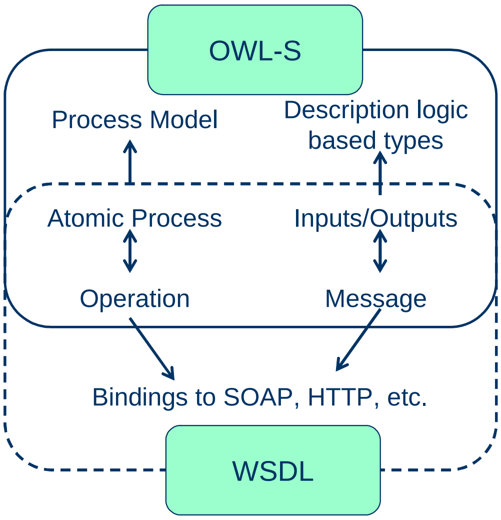

Figure 2 - Formal Mapping of OWL-S to Web Services (WS)

OWL adds more vocabulary for describing properties and classes: among others, relations between classes (e.g. disjointness), cardinality (e.g. "exactly one"), equality, richer typing of properties, characteristics of properties (e.g. symmetry), and enumerated classes.

OWL-S is quite complex for a non-expert to understand. It needs tools like OWL-S API and WSDL2OWL, which currently have some severe limitations. A more principle criticism is that OWL-S only focuses only on one ontology, it should allow multiple parameters mapping to one syntactic parameter.

Web Services Semantics (WSDL-S)

WSDL-S, submitted to the W3C by IBM on 1 October 2005, is using the Extensibility Elements of WSDL 2.0, i.e. semantic annotations are added to the WSDL document elements that have constructs to represent service descriptions like interface, operation, message, binding, service and endpoint. The first three, namely interface, operation and message constructs deal with the abstract definition of a service while the remaining three (binding, service and endpoint constructs) deals with service implementation. The WSDL-S proposal focuses on semantically annotating the abstract definition of a service to enable dynamic discovery, composition and invocation of services.

WSDL-S defines a Semantic Model to capture the terms and concepts used to describe and represent the Web service. WSDL-S uses the extensibility of WSDL. Four particular parts of the semantic model are distinguished:

1. Input Semantics – the meaning of input parameters
2. Output Semantics – the meaning of output parameters
3. Precondition – a set of semantic statements that are required to be true before an operation can be successfully invoked
4. Effect – a set of semantic statements that must be true after an operation completes execution after being invoked. Different effects can be true depending on whether the operation completed successfully or unsuccessfully.

Semantic annotation in WS descriptions is an obvious first step in bridging Web services with Semantic Web technologies. It relies on both the WSDL and XML schema extension mechanisms to reference external semantic models. It is a lightweight approach extending WSDL files with semantic annotations. WSDL-S is independent of any particular semantic annotation language, examples of which include WSMO and OWL-S.

The lightweight approach makes WSDL-S easy to handle. But it does not support mapping of many semantic parameters to one web service parameter, as it is restricted it to the principle mechanisms of WEB services, mainly to achieve a result specific to one service. Main criticism is that it is too flexible with respect to model, script and rule language. Thus values of semantic parameters are not always identical to values of syntactic parameters [8].

Semantic Web Services Framework (SWSF)

SWSF was submitted to W3C by National Institute of Standards and Technology (NIST), National Research Council of Canada, SRI International, Stanford University, Toshiba Corporation, and University of Southampton on 9 September 2005 [9].

SWSF includes the Semantic Web Services Language (SWSL) and the Semantic Web Services Ontology (SWSO). It builds on the WSDL (v1.1), and extends through supporting the specification of workflows composed of basic services. A promising candidate is the Business Process Execution Language for Web Services (BPEL4WS). The second focus is on the choreography, the information exchange and agreed ordering rules to perform a Web service-based transaction. With respect to registering Web services for purposes of advertising and discovery, will SWSF build on the UDDI.

FLOWS, the ontology of service concepts used in SWSF, is a more comprehensive ontology as compared to OWL-S. In terms of coverage it is distinguished by its axiomatization of messages, something that was not addressed in OWL-S. Both attempt to provide an ontology for Web services, but FLOWS had the additional objective of acting as a focal point for interoperability, enabling other business process modeling languages to be expressed or related to FLOWS.

Web Service Modeling Ontology (WSMO)

WSMO was submitted to W3C by DERI Innsbruck at the Leopold-Franzens-Universität Innsbruck, Austria, DERI Galway at the National University of Ireland, Galway, Ireland, BT, The Open University, and SAP AG on 04 April 2005.

WSMO (Web Service Model Ontology) uses WSMF's (Web Service Modelling Framework) four elements for describing semantic Web Services [10]:

- Ontologies for terminology used by other elements
- Goals is the definition of the problem the Web services is to solve
- Web Services descriptions that define different aspects of a Web Service
- Mediators handle interoperability of other ontologies.

These four are syntactical modelled by WSML (Web Service Modelling Language).

WSDL is a form of syntactical contract by specifying the format of the messages sent between the web service and the client. In contrast, WSMO describes functionality and behaviour of the Web Service. The descriptions are for discovering and automatic composition of Web services. The UDDI (Universal Description, Discovery and Integration) registry for WSDL specifications are based on keywords. WSMO discovery uses the semantics in the WSMO description of a web service. WSMO can be located in UDDI registers, but it is more effective to have a specific WSMO repository as in the ASG platform.

WSMO versus SWSF
WSMO is a parallel effort to SWSF and base its work on almost the same fundamental technologies, e.g on F-logic. Nevertheless, the two groups have pursued complimentary goals. WSMO has focused heavily on the language effort. In particular to develop a "conceptual syntax" for top-level descriptions of services; this might make the specifications easier to read for the end user. WSMO has also paid special attention to the issue of OWL compatibility [11].

The major distinction between the WSMO effort and the SWSF is with respect to the ontology domain. WSMO is focusing on describing Web service choreography through guarded transition rules, while SWSF focuses on extending the functionality of the rule language (SWSL-Rules) that supports meta-reasoning and reification extensions.

WSDL-S versus WSMO
In WSDL-S, the semantic model of a Web service is expected to contain the semantics of input and output parameters and the specifications of preconditions and effects of service operations, plus the categorization of a WSDL interface. WSMO specifies a more detailed model where a Web service can have a capability with preconditions, assumptions, postconditions and effects; and an interface with choreography and orchestration; and where data are described using ontologies.

WSDL-S is based on the WSDL model of Web services interfaces consisting of separate operations, and preconditions and effects in WSDL-S are attached to operations. In contrast, WSMO distinguishes between preconditions, assumptions, postconditions and effects, but these aspects are modeled on the whole Web service. WSMO talks about a Web service as a whole, without splitting it to operations. Additionally, WSMO models the choreography and orchestration interface(s) of a Web service, an aspect not covered by WSDL-S at all.

We have selected WSMO for the interaction with external services, as it is more focused towards end-user requests (as compared to SWSF). WSMO's interaction model includes choreography and orchestration, and is as such superior to OWL-S, which has orchestration and WSDL-S, which has no interaction model.

The ASG Model and Architecture

Our Semantic Service Platform is based on the ASG Reference Model [12], shown in Figure 3. ASG is a SOA compliant system, as it fulfils the OASIS RM requirements. ASG adds important features like negotiation of non-functional parameters, monitoring of service delivery and dynamic invocation of external services [13].

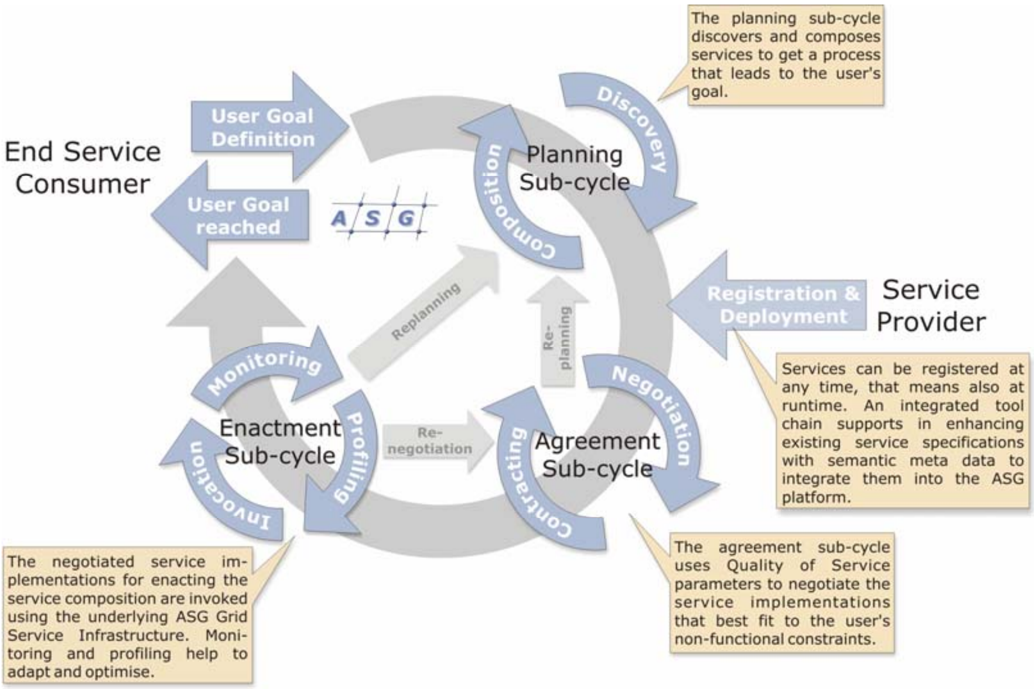

Figure 3 - The ASG Reference Model

In the area of service planning, ASG supports discovery and composition:

- *Composition*: This is an automated service composition according to a specified goal.
- *Discovery:* It provides functionality to store, retrieve and delete semantic service specifications, service grounding specifications, composed services and ontologies.

In the agreement of a service, ASG supports negotiation and contracting:

- *Negotiation:* It provides functionality responsible to select appropriate service implementations (groundings) with respect to Quality of Service (QoS) parameters, which are defined in the user request that should be used inside a service composition.

- *Contracting:* It provides functionality used to mange the contracts between the parties that are involved in a service usage scenario.

In the enactment of services, it supports profiling, invocation and monitoring:

- *Invocation:* It provides standardized interfaces to instantiate and to invoke atomic services. It abstracts and encapsulates possible heterogeneous service hosting and runtime environments of the underlying ASG Infrastructure, e.g. a Grid infrastructure. A major responsibility is the late and adaptive binding of resources in the ASG infrastructure to execute atomic service implementations enabling efficient load balancing and reliable service provisioning.

- *Profiling*: It provides functionality responsible for collecting history data and building service profile that can be used during negotiation process. It offers functionality to dynamically log and retrieve service enactment behaviour characteristics like response time, reliability, functional correctness, etc.

- *Monitoring:* It provides functionality to monitor service instance invocation characteristics. Examples of monitoring attributes are performance (response time) or costs. The offered monitoring information is used to dynamically identify eventually broken Service Level Agreements (SLAs) or unreachable services, which would trigger an adaptive re-planning of the composed service.

Entry point for all external usable functions of an ASG reference architecture implementation is the Façade.

Applying ASG to Telematics

In this chapter we will give an overview of how to make a new ASG (a SOA compliant system) services in general and ASG based Telematics services specially. When starting to realise a SOA system and services it is recommended to choose a well defined method supporting the promised capabilities of SOA. The method used by ASG is the Use Case Definition that defines the use case in a scenario within your service domain, e.g. a commuter in the Telematics domain.

Our scenario describes the commuter (or tourist), who expects dynamic information about traveling (see Figure 1). The customer coordinates combined with information about the user's route, the traffic conditions on alternative routs, traffic forecasts, user preferences, public transportation, etc. can then be utilised to sort out the relevant incidents in the traffic and propose best-effort action for the user. The Traffic Incident Service alerts the traveller about major incidents that is likely to influence the planned journey, using the customer's coordinates, direction and route information, in combination with the coordinates and expected delay caused by the traffic incident.

The list of stakeholders contains companies interested in supplying services to the end-user, as such: Mobile operators and location providers, Payment operators, Points-of-Interest providers as e.g. tourist offices, lodging providers, guiding and sightseeing providers, Encyclopedia and travel guide providers. Car/traffic related providers are gas and service stations, road authorities, radio broadcasters, traffic and weather forecasts.

The Application flow describes the user interaction flow in sketches and is detailed in Use Cases descriptions. The Telematics environment with potential stakeholders listed previously covers a wide Service Landscape. This service landscape, a list of atomic services, is too complex to be composed to end-user services in a conventional platform. Service Composition is required to identify and describe the possible types of services based on semantics of services from the service landscape.

From the previously analyzed SWS initiatives we see that not a single ontology will exist, but that different ontologies have to co-operate to fulfill the service request of the traveler (see Figure 4). We suggest a many to one mapping as in [14], allowing ontologies to interact with each other.

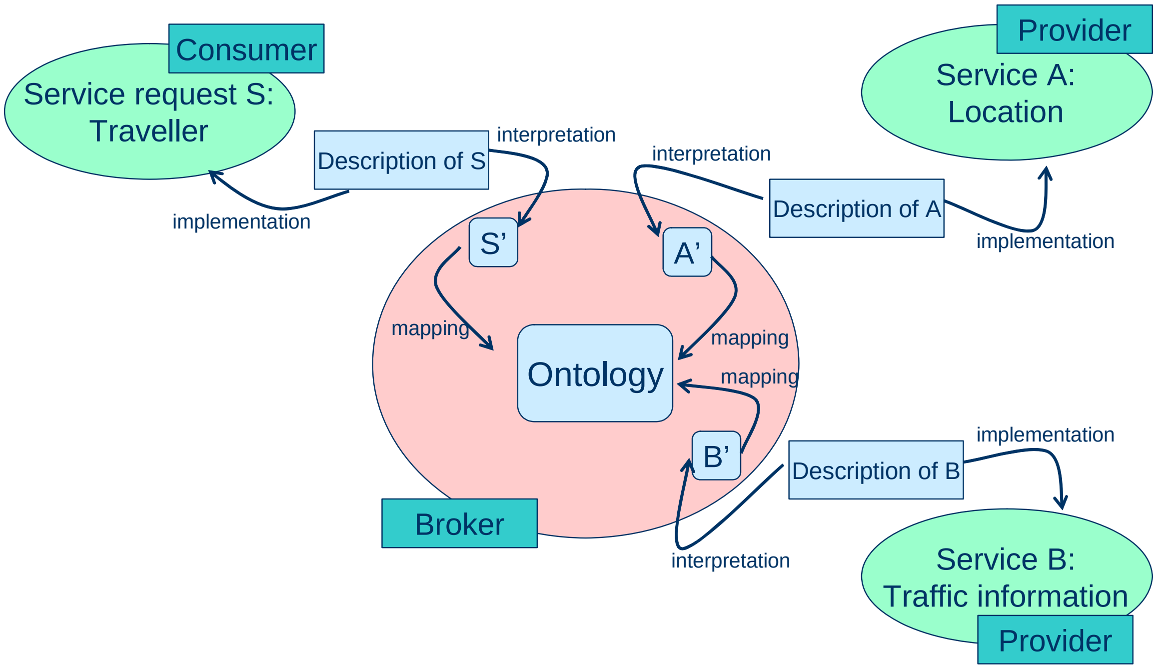

Figure 4 - Telematics, many to one mapping of services in the ASG platform.

Adding Semantics to Telematics

ASG combines an advanced Semantic Web Services platform with real-world services. This section provides initial steps on how to add ontology to the external services, and ground them to the platform.

Since ASG uses WSMO the attraction booking service is equipped with ontology to add semantics to the description of the web service. The goals are defined, the web services itself are described and if there are external ontologies the mediators handle this. Ontologies are under development, but far from being usable right now. Our service example of “attraction booking” uses the Protégé software[2] to build the WSMO description of the booking service. It includes three files: a project file binding the files together, an ontology file and a file for the instances of the ontology. Figure 5 demonstrates the usage of Protégé for defining Instances of the Concert booking.

[2] Protégé software, http://protege.standford.edu

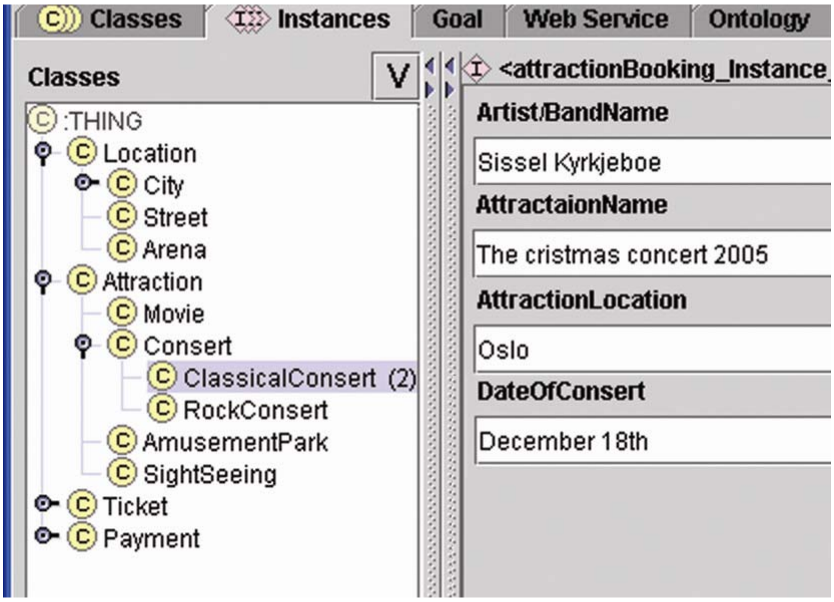

Figure 5 - Instances of the use case "Attraction Booking" in the Telematics Scenario

Using WSMO for semantic web services includes the ontology element, here WSML. The following lines are taken from the WSML code in order to explain the relation between concepts, non functional parameters and instances:

```
concept ClassicalConcert subConceptOf Concert
  nonFunctionalProperties
    dc#description hasValue "Concept of a classical concert "
  endNonFunctionalProperties
  hasArtist ofType human
  hasDateOfConcert ofType _date
  hasAttractionName ofType _string
  hasLocation ofType Location
```

This implementation work supports the envisaged goal of supplying semantically supported Web Services, following the roadmap outlined in the next section.

Roadmap for Telematics Services

The availability of open Telematics services in the future will depend highly on two parallel developments: The industry uptakes of SWS and the standardization of the telecommunication Web Service (Parley-X/3GPP OSA), including semantics for Telematics data.

In ASG ongoing work on the service platform and prototype as well as service developments show that SWS maturity is under way, and supporting tools are expected to reach the developer market within mid 2006 (Figure 6). A more general approach to the roadmap includes not only the WSMO standardisation used here, but also the developments in WSDL-S and OWL-S. Most important for service brokers are Semantic Toolkit developments, domain ontology developments and the interoperability between different ontologies [11].

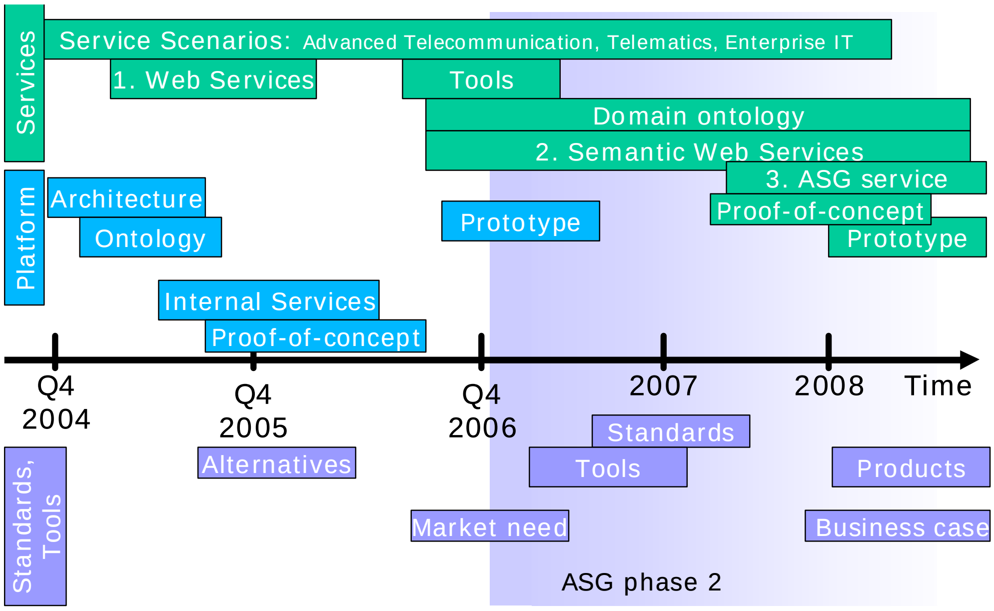

Figure 6 - Roadmap to ASG based services

Conclusion

This paper discusses the ongoing developments in the Semantic Web Services world, based on a specific usage scenario. Four scenarios in Telecommunications, Telematics, IT-industry and eGovernment have been used to establish the 7 business requirements for a future semantic service platform. The Adaptive Services Grid (ASG) platform fulfills the requirements, and extends the specifications of the OASIS SOA Reference Model (RM). In spite of the big conformity to the RM it is a great chance that interoperability problems will occur between an ASG system and an OASIS compliant SOA system. The reason is the choice of different semantic standards for describing their respective data models. In comparing the SWS approaches OWL-S, WSDL-S, SWSF and WSMO we suggest to support heterogeneous ontologies through a many to one mapping.

Based on an example in the Telematics domain, the design specifications for advanced services are outlined. The business potential is addressed through an analysis of end-user services. Roles of stakeholders are identified, and their atomic services put together to value-added end-customer services. We selected WSMO as the ontology for service Implementation, as it offers the most flexible service invocation and supports mediators for different ontologies.

Implementation work is outlined through attraction booking scenarios. Ongoing platform and prototype service developments show that prototype development is well under way, and tools are expected to reach the developer market within mid 2006. More crucial are domain ontologies, standardization, and the service inclusion through existing platforms, expected to happen until 2008. Substantial market penetration will then follow in the years afterwards.

References

[1] J. Hendler, T. Berners-Lee, and E. Miller, "Integrating Applications on the Semantic Web," Journal of the Institute of Electrical Engineers of Japan, Vol 122(10), October, 2002, p. 676-680.

[2] F. Kileng, J. Noll, E. Lillevold, B. Peissl, A. Wahler, M. Pilarski, D. Dmowski, M. Momotko, V. Törmälä, V. Lyytikäinen, "Design of ASG-based services", Adaptive services Grid, – EU Deliverable D7.II-1, August 4, 2005, http://asg-platform.org

[3] S. Dalal, S. Temel, M. Little, M. Potts, and J. Webber, "Coordinating business transactions on the web", *IEEE Internet Computing*, 7(1):30--39, 2003

[4] OASIS, "Reference Model for Service Oriented Architectures", Working Draft 10, 15 November 2005, http://www.oasis-open.org/committees/ download.php/15389/wd-soa-rm-10.pdf

[5] S. Arroyo, R. Lara, J. Gómez, D. Berka, Y. Ding, and D. Fensel: Semantic Aspects of Web Services. In M. P. Singh (ed.), Practical Handbook of Internet Computing. Chapman & Hall and CRC Press, Baton Rouge, 2004.
[6] D.L. McGuinness, F. van Harmelen, "OWL Web Ontology Language Overview", W3C Recommendation 10, Feb 2004, http://www.w3.org/TR/owl-features/
[7] R. Lara, A. Polleres, J. Scicluna, H. Lausen, "Formal Mapping and Tool to OWL-S", WSMO working draft 17 Dec 2004, http://www.wsmo.org/TR/d4/d4.2/v0.1/
[8] J. Kopecký, "Aligning WSMO and WSDL-S", WSMO working draft D30v0.1, 5.Aug 2005, http://www.wsmo.org/TR/d30/v0.1/20050805/
[9] S. Battle, A. Bernstein, H. Boley, B. Grosof, M. Gruninger, R. Hull, M. Kifer, D. Martin, S. McIlraith, D. McGuinness, J. Su, S. Tabet, "Semantic Web Services Framework (SWSF)", W3C Member Submission 9 September 2005, http://www.w3.org/Submission/SWSF/
[10] D. Roman, U. Keller, H. Lausen, J. de Bruijn, R. Lara, M. Stollberg, A. Polleres, C. Feier, C. Bussler, and D. Fensel: Web Service Modeling Ontology, Applied Ontology, 1(1): 77 - 106, 2005.
[11] R. Lara, D. Roman, A. Polleres, and D. Fensel, "A Conceptual Comparison of WSMO and OWL-S", Lecture Notes in Computer Science, Vol. 3250, Jan 2004, pp 254 – 269
[12] G. Laures, K. Jank., "ASG Reference Architecture: Requirements, Current Efforts and Design", Adaptive Services Grid – EU Deliverable D6.V-1, 31 July 2005, http://asg-platform.org
[13] J. Noll, E. Lillevold, "Applying Semantic Web Technologies for mobile communications", WWRF #15, Paris, 8.-9.12.2005
[14] G. Vetere and M. Lenzerini, "Models for semantic interoperability in service-oriented architectures", IBM Systems Journal, Vol 44, No 4, 2005, pp 887-903

Josef Noll is Prof. stip. at the University Graduate Centre at Kjeller, Norway (UniK) in the area of Mobile Services. He is also part time Senior Researcher at Telenor R&D in the Products and Markets group. He received his Ph. D. from the University of Bochum (D), worked for the European Space Agency at ESTEC from 1991-1997, and from 1997-2005 at Telenor R&D. His working areas include Mobile Authentication, Wireless Broadband Access, Personalised Services, Mobile-Fixed Integration and the Evolution to 4G systems. In ASG he is activity leader for the business and use case.

Erik Lillevold is former Senior Researcher at Telenor R&D with more and 30 years of experience in different fields within Computer and Communication Science. From 1971 – 1986 he was a research scientist at NDRE (Norwegian Defense Research Establishment) working with different kind of computer communication projects. NDRE has close relation to US DoD. Therefore he become early involved in the DARPANET (later Internet) project (1974) ending with a stay as International Fellow at SRI, Menlo Park, California (1981). Lillevold was later on involved in ISO and CCITT standardization of the MHS (Message Handling System) developed by CCITT and specified in the X.400 Recommendations.
In ASG he focuses on Web Services and Agent Technology, in addition to telecom systems.

Marianne Rustad has taken her bachelor degree at Bodø Graduate School of Business. The master degree was started at Brunel University West London and is now under completion at the University Graduate Center UniK. Her master thesis includes real-world service access in the ASG environment and comparisons of Semantic Web services approaches.