Semantics to energize the full Services Spectrum
Ontological approach to better exploit services at technical and business levels

Amit Sheth, Kunal Verma, Karthik Gomadam
LSDIS Lab, Dept of Computer Science, University of Georgia
{amit, verma, karthik}@cs.uga.edu

Introduction

Services are pervasive in today's economic landscape, and services-based architectures are being rapidly adopted as IT infrastructure. The need to take a broader perspective of services to include people and organizational descriptions as opposed to technical interface descriptions has already been recognized as part of an overall vision of Services Science [3][5]. The focus of this article is to present the Semantic Services Science (3S) model, which seeks to demonstrate the essential benefits of semantics to the boarder vision of Services Science, with service descriptions that capture technical, human, organizational and business value aspects. We assert that ontology based semantic modeling and descriptions can be used to energize services across the broad service spectrum. In this article, we demonstrate how 3S approach could be used along three points in this spectrum: 1) semantic descriptions of standard Web services with the help of WSDL-S, semantic policies and agreements, 2) semantic descriptions of light weight Web services using Web 2.0 technologies (e.g. REST, AJAX) and 3) ontology based profiling of people and organizational aspects of the assets associated with the knowledge services.

Why use semantics as the basis of the service model? Interoperability has been a key challenge in IT for well over a decade. While Electronic Data Exchange standards and XML has provided the basis for data exchange and syntax level interoperability, IT infrastructure needs semantic interoperability to fully exploit and interoperate with respect to all its data and service resources. The fast emerging Semantic Web shows how the use of formal knowledge representation, typically in the form of ontologies, leads to machine processable descriptions, and how the adoption of ontologies that provide common vocabulary and shared knowledge leads to improved semantic interoperability. There are three primary advantages of creating models that employ semantics -1) they promote reuse and interoperability among independently created and managed services, 2) ontology supported representations based on formal and explicit representation lead to more automation, and 3) explicit modeling of the entities and their relationships between them allows performing deep and insightful analysis. For example, if a particular business component is declared as critical, and the relationships between the business and IT components supporting them are explicitly modeled, simple semantic queries may allow business managers to verify if adequate IT resources have been allocated to that component [6]. This reasoning would have otherwise required a systems analyst with great deal of knowledge of both the business values and IT of the organization. 3S utilizes semantic descriptions to capture relationships between services, the people, the organizational aspects, the business values and allows business managers to reason on them to easily create new services and to efficiently allocate resources to the services.

On the technical front, much of the modeling effort on services has so far focused on standard Web services in the context of SOA enabled by Web service Description Language (WSDL), SOAP (Simple Object Access Protocol, an XML-based message exchange format) and UDDI (Universal Description, Discovery, and Integration, a technical specification for implementing registries that allow publication and discovery of Web service). It is however possible to take XML-based descriptions used by these standard Web services (and in principle other syntactic descriptions of services) and annotate them with semantics specified in ontologies or conceptual models to gain the above discussed benefits of a semantic approach. The emerging field of Semantic Web Services (SWS), semantics is exploited to discover services using semantic (rather than syntactic) description, to more effectively integrate, compose or orchestrate services to support workflows or processes. The same approach to semantic interoperability can be accorded to fast growing web-based services using Web 2.0 technologies (e.g. REST, AJAX),

often termed as lightweight services, and to more expansive knowledge services that go beyond the scope of Web services to encompass human skills and organizational aspects.

Why use the 3S Model? As businesses re-factor their organizational structure to become service providers, they must possess the ability to provide new services by optimizing their technical and organizational resources across the globe. The 3S model provides a comprehensive model of the all resources of a business that can be queried using Semantic Web technologies to quickly assemble the resources for providing new services. Let us consider the example of a global telecommunications provider that decides to provide a new video streaming service in the USA. From an organizational perspective, the manager would try assemble a team that has a mixture of experience in delivering this technology in other countries as well as in depth knowledge of the program availability, demographics and market preferences in the USA. Based on the knowledge profiling aspect of the 3S stored using ontologies, the project manager can quickly identify relevant organizational roles for this project. Similarly the technical aspects of the 3S model can be used for service discovery (using UDDI registries) to discover technical resources whose capabilities are semantically described using WSDL-S. In addition, with the help of semantics descriptions of Web 2.0 based light based services; Web interfaces to aggregate data from multiple sources can be quickly created for customer and internal use. This example provides one instance of the use of the 3S model with respect to resource discovery. Other benefits include using the 3S model for composition of services, runtime adaptation of business processes, Web based collaboration and generating context driven content.

Broader Definition of a Service

"A service is a provider-client interaction that creates and captures value" [4]. The challenge for service providers is to create services by optimally leveraging all their resources, often in a global environment. It requires that service providers are cognizant of not just their software capabilities but also of their global workforce, the current business trends and demands of their consumers. As businesses reorganize their overall structure to focus on their core-competency and become primarily service providers, a broader perspective of services beyond the traditional services known to computer science professionals is needed. This pervasive view must account for technical, people, social, organizational, and business aspects of offering a service. The 3S model makes an initial attempt to act upon Paul Horn's suggestion of identifying human and software assets as a starting point of modeling a service. It builds upon previous modeling efforts in Web services, knowledge profiling and asset management. We define a service as: - "A service is value proposition provided to an organization and its definition includes the following – 1) technical description of the implementation technology (SOAP/WSDL, REST, AJAX); 2) organizational description of the people and their roles involved in developing, rendering, managing or supporting the service; and 3) the business or organizational value it renders."

Consequently the relationships between the components of a service related to parts 1 and 2 of the definition are shown in Figure 1. The central entity is the organization that consumes or offers services. A service is realized using the assets that the organization possesses (a more complete view may be extended to include external assets including those employed, co-development and outsourcing). The assets include software, hardware, IP, and human assets. Software assets include any software, applications or infrastructural components that can be converted to services or used to create services. From an implementation point of view, the software assets may convert to services by either using Web service technologies (WSDL, SOAP, UDDI) or light weight approaches like REST and AJAX. Human assets include project managers, software developers, customer relationship representatives and other relevant persons involved with creating, developing, marketing, maintaining or otherwise managing the service.

Our definition of service is specifically targeted towards business managers involved in creating a new service/project, and requires targeting the problem from both functional and organizational perspectives. The functional perspective involves locating services for reuse and employees that can help create the new service/project. The organization perspective involves creating a team and defining their roles and responsibilities. Here semantics are used to enhance the descriptions

so that business managers can realize new projects/services efficiently. Our framework for semantically describing assets is described next.
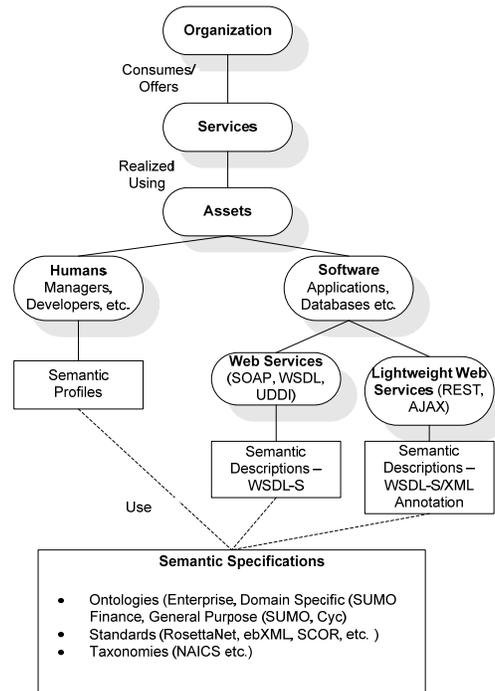

Figure 1: A broader view of a service that includes people, technology and the organizational perspective

Semantics for the complete spectrum of services

The recently defined field of Semantic Web tries to add more meaning to resources (both information and services) on the Internet by annotating it with concepts from agreed upon domain models. The annotations act as a common underpinning between different users and promote greater interoperability. Ontologies provide the most accepted way of creating conceptual models for domains. W3C recommended ontology specification languages like OWL and RDF/S provide a way to specify conceptual models using formal languages, while UML maybe an alternative language for modeling assets such as software. In the context of the Semantic Web, the ontologies are formal specifications of knowledge that capture or represent agreements. Ontologies can be domain specific with the scope of an entire industry (e.g., manufacturing), sub-area of a science (e.g., Glyco ontology for Glycomics) or only a type of application (e.g., for Sarbanes-Oxley compliance). They can also be domain-independent covering eCommerce (e.g., ontology based on RosettaNet or ebXML) or policy and agreement (e.g., ontology based WS-agreement). Note that in many cases, agreements documented as textual specifications can be captured as ontologies by using formal representations. While we do not particularly focus on other forms of capturing semantic models in this article, less expressive knowledge representations like taxonomies, thesauri or folksonomies can be useful in capturing domain knowledge with corresponding loss in degree of interoperability and automation.

Our approach for semantically representing the complete spectrum of services is based on annotating Web service descriptions with domain-specific and domain-independent ontologies. This is exemplified in WSDL-S [12] that provides an incremental and evolutionary approach for adding semantics to existing Web service standards by using extensibility elements of (WSDL). In addition to WSDL-S, three other specifications have been recognized by W3C as member submissions to be considered as inputs for possible recommendations in this area. This is an indicator of the growing recognition by standard bodies like OASIS and W3C about the need for

semantic descriptions of the services to handle real world issues of efficient reuse and interoperability.

Semantic Web service Proposals and Community efforts

Table 1 shows the four major specifications submitted as input to W3C for Semantic Web services recommendations. In addition to these, have been work in semantically capturing the non-functional aspects of Web services (e.g. semantic policy and semantic agreement).

Table 1: A comparison of four Semantic Web service specifications submitted to W3C as input for further activity in this area.

Specification	Description	Formalism
OWL-S [7]	An OWL based upper ontology semantically representing Web services.	Description Logics
SWSL (Semantic Web service Language) [10]	A combination First Order Logic and rules to represent Web services.	First order logic, Different variants of rule languages (Horn, HiLog etc.)
WSDL-S [12]	Use of extensibility elements in WSDL to annotate elements with terms in ontologies.	Agnostic (examples typically use description logics, but use of UML for conceptual modeling is also recognized)
WSMO [13]	A F-Logic based conceptual model for representing Web services.	F-Logic

A relevant effort that expresses views on architectural issues is [1], the W3's proposed task forces in this area are at [11], and a sample of books on this topic include [2] and [9].

Semantics for technical services

Next we give an overview of the technical aspects of the 3S model with respect to Web services and lightweight Web services.

Semantics for Web services

Web services provide standards based solution for creating such services by facilitating reuse, interoperability and composition of existing services and applications. While Web services standards resolve platform heterogeneities, and XML provides the basis of syntactic interoperability, there exist many other types of heterogeneities in the business environment as shown in

Table 2. The table brings into the focus how Web service standards only provide support for platform/system level heterogeneities since they operate at the syntactical level. Semantics is a critical requirement for handling all the other heterogeneities.

Table 2: Types of heterogeneity and how SWS may enable semantic annotations that support interoperability at that level (* indicates support in WSDL-S, and may also be supported by other SWS specifications)

Type of Heterogeneities	Approaches to resolve heterogeneities
Interaction Protocol /Execution Level Difference in interaction protocol of services/exception handling capabilities	Using preconditions and effects to specify ordering constraints*. Semantic annotation of exceptions.
Non Functional Heterogeneous specification of provider requestor non-functional requirements	Semantic annotation of WS-Policy and WS-Agreement

Functional Difference in functional descriptions of Web services	Annotating operations and adding preconditions/effects of the operations *
Syntax/Structure/Semantic Difference In data schemas at syntactic and semantics levels	Data Mapping support by annotating inputs and outputs with ontological concepts* (XML supports syntactic interoperability
Platform/System Different operating systems, Platforms	Web service Standards (SOAP and WSDL)

One relatively comprehensive approach for semantically modeling Web services is based on four types of semantics [8] – functional (what a Web service does), data (how to interact with the service), non-functional (including quality of service attributes of the service) and execution semantics (modeling runtime behavior/exceptions etc.) of the Web service. The different types of semantics are critical enablers of the different value propositions of Web services – 1) *search* or discovery (the ability to match service partners, often consumers and providers), 2) *reuse* (easily use the services found), 3) *interoperability* (ability to freely exchange data between the services and substitute services with the same functionality) and 4) composition (being able to combine multiple services to carry out complex business processes).

For *search* and *reuse*, services and requests are annotated with functional semantics so that they can be efficiently discovered by services or components that require the desired functionality.

Table 2 provides an insight into how semantics can be used to resolve different types of heterogeneities for *interoperability*. Finally composition uses functional semantics for finding the appropriate services, data semantics for passing data across the services, non-functional semantics to ensure that services meet the non-functional criteria and execution semantics for ensuring that the ordering constraints of invoking operations of services are not violated.

Adding Semantics to light-weight Web-based services and mashups

An important contribution of the 3S model is proposing the use of semantics in Web 2.0 based lightweight services which are now increasingly being realized using the REST architectural approach. REST based services use the classical HTTP GET-POST approach to invoke services, Uniform Resource Identifiers (URIs) to represent resources and messages encoded in XML for communication. Although SOAP has advantages in terms of greater tooling support and the ability to support quality of service guarantees (e.g. security and transactions), the ease of use and lightweight style of interaction over the web using XML have made REST services very popular. One of the most popular applications of lightweight Web services is called a mashup, which is basically a web site that aggregates content from different providers. A mashup uses lightweight services to query the providers to get content in XML format. Due to difference in data definitions (XML schemas) of different providers, a semantic approach is needed for seamless integration of the data. We exemplify the need for semantics with the help of the example below.

Figure 2 illustrates an example of a fictional online bookseller application myBook.com that is created using lightweight Web services. Whenever a user wants to buy a book, the application queries different book vendors (in our example, the vendors are ubn.com and yaos.com) using REST Web services. It is often the case that the XML schemas of the data returned by different vendors are different from the schema of the myBook.com. We propose adding semantics to light weight services by annotating the XML schema of the service inputs and outputs. In the example illustrated below, the conceptual model (ontology) captures the domain of books and all the sellers annotate their XML schema. In the example, annotating the <shippingDetails> element in the yaos.com and <shipping> element in the myBook.com schema, with the same concept in the ontology will help the application infer that they are semantically similar.

Using this approach will allow myBook.com to address the heterogeneities in the schemas of different providers and deliver the content seamlessly to the user.
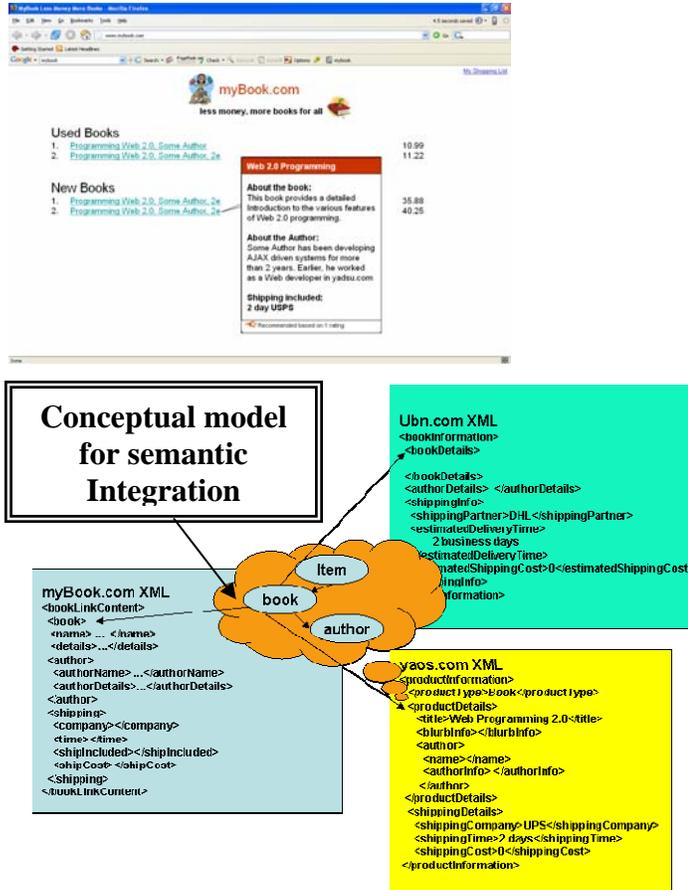
Figure 2: Scenarios demonstrating semantics for light-weight Web-based services

Using Semantics for Organizational and People Perspective

In this section, we provide an overview of the organizational and people aspects of the 3S model. Knowledge profiling entails formally managing knowledge resources in order to facilitate access and reuse of knowledge. Knowledge that is profiled depends on the asset that is being profiled. For example, to profile a Web service asset, one would need to profile features like the capability of the Web service, data semantics of the inputs and the outputs and policies associated with the service that capture enterprise features offered (like security, reliable messaging) and also quality of service guarantees like round trip network time. This can be done with ontology-driven semantic annotation of service descriptions. On the other hand, when profiling a human asset such as a software engineer, we need to capture the knowledge with respect to his areas of expertise, skill sets, relationships experience, the projects he has worked on and the roles he has played in the projects. This is along the lines of creating a social network for the human capital of an organization. The semantic descriptions would allow these human service engineers to be found, create relationships with, streamlined, and applied to appropriate part of the service creation and enactment.

One of the main purposes of enterprise knowledge profiling is to capture knowledge so that it is accessible and reusable to the enterprise. At a physical level, infrastructures like knowledge warehouses can be created to store this knowledge. However, an important factor in determining reusability and accessibility is the logical structure or the conceptual models that dictate the way the knowledge is stored. In order to be able to share the knowledge that is profiled, one would need to go beyond schemas while capturing knowledge. It is here that semantics help.
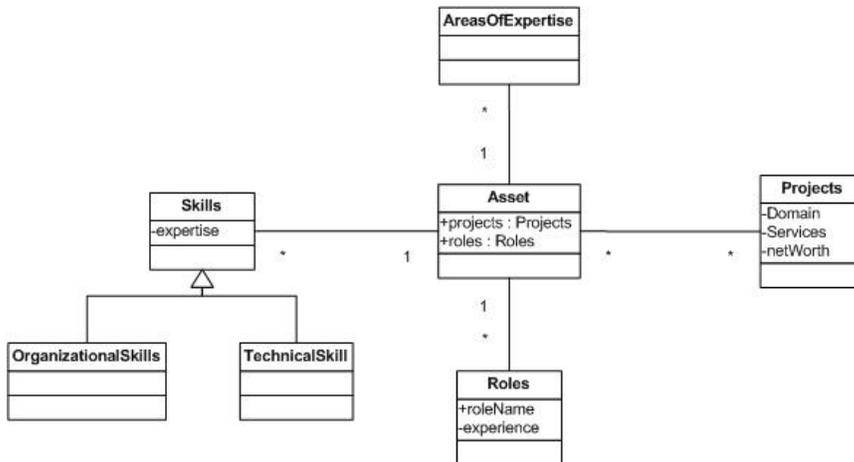
6

Figure 3: Conceptual model (outline) to profile a human asset in software development

Figure 4 give an outline of a conceptual model to profile a human asset in software development. The model captures the projects the person has worked on, the roles he has played (developer, project manager), his areas of expertise (networking, SOA, project management), his technical skills (like programming, network management), and his managerial and organizational skills. Further it also captures his experience in various roles. Based on the semantic profiles of human assets, the managers can quickly assemble resources for new projects.

Service Sciences in Universities

Table 3: Different types of service courses taught in top 40 computer science schools

System Level Courses in Services	Application Level Courses in Services	Business Level Courses in Services
12	18	1

We reviewed curriculum of 40 prominent computer science departments to study the number of services or Web services based courses at graduate level. We divided the courses into three groups – 1) system level courses where the emphasis was on operating system level support for Web services, 2) application level courses where the emphasis was on learning Web service technologies to build applications and 3) courses that discuss the business aspects of services. From our brief survey, we concluded that while 75% of the universities have at least one course on technical aspects of Web services, there is no coverage of the business and organization aspects of services as advocated in Service Sciences. In the context of globalization induced transformations occurring in industry, there is increased need to supply graduates from technical areas (such as Computer Science) with better understanding and knowledge of business and organizational aspects. Our view is that Computer Science departments on their own are not well equipped and well motivated do a better job in meeting this expectations of industry. Two possible agents of change are (a) better involvement by industry in conveying requirement to Computer Science Departments (e.g., by participating on curriculum committees), supported by more collaboration and funding (Industry funding of higher education research is a small fraction of funding from federal and state sources) and (b) significant collaboration between business and Computer Science departments. The first of this is unlikely to happen if the industry continues to put all its effort in global sourcing of technical talent, ignoring its involvement in higher education at home.

Conclusions and Future Work

In this paper, we have presented the Semantic Services Sciences (3S) model. The key to this model is the use of ontologies in representing the technical, people and organizational aspects of services. In addition to further refining this model, we intend to pursue using semantics to link

organizational layers in a services based enterprise. [6] proposes using ontologies and Semantic Web techniques to represent relationships between the models of different organizational layers of an enterprise – the strategy model, organization model, the execution model and the implementation model. This would lead to more direct links between the business layers and the IT infrastructure leading to more efficient alignment of IT services with the business goals of the enterprise. We believe that using the information captured in the 3S model can be used as a starting point to implement the architecture proposed in [6]. Our future work includes extending the 3S model to capture additional semantic descriptions; examples include semantic descriptions of business value of services, and novel forms of assets such as those involved in rendering knowledge services or in staging experience.

References

[1] M. Burstein, C. Bussler, T. Finin, M. Huhns, M. Paolucci, A. Sheth, and S. Williams, A Semantic Web services Architecture, IEEE Internet Computing, September-October 2005, pp. 52-61.

[2] J. Cardoso and A. Sheth, Eds., Semantic Web services, Processes and Applications, Springer Book Series on Semantic Web & Beyond: Computing for Human Experience, 2006.

[3] Paul Horn. The New Discipline of Services Science, Business Week, January 21, 2006. http://www.businessweek.com/technology/content/jan2005/tc20050121_8020.htm

[4] Services Sciences, Management and Engineering, http://www.research.ibm.com/ssme/

[5] Jim Spohrer & Paul Maglio. Emergence of Service Science: Services Sciences, Management, Engineering (SSME) as the Next Frontier in Innovation, (presentation), IBM Almaden Research Center, October 2005.

[6] Juhnyoung Lee, 2005. Model-driven business transformation and the Semantic Web. Communications of the ACM, volume 48, (12), Dec. 2005, 75-77.

[7] OWL-S: Semantic Markup for Web services, W3C Member Submission, http://www.w3.org/Submission/2004/SUBM-OWL-S-20041122/

[8] A.P. Sheth, "Semantic Web Process Lifecycle: Role of Semantics in Annotation, Discovery, Composition and Orchestration," Invited Talk, Workshop on E-Services and the Semantic Web, WWW, 2003. http://lsdis.cs.uga.edu/lib/presentations/WWW2003-ESSW-invitedTalk-Sheth.pdf

[9] M. P. Singh and M. N. Huhns, Service-Oriented Computing Semantics, Processes, Agents, John Wiley & Sons, Ltd., 2005

[10] SWSL, Semantic Web service Language, W3C Member Submission, http://www.w3.org/Submission/SWSF-SWSL/

[11] W3C Semantics for Web services Characterization Group Charter, http://www.w3.org/2005/10/sws-charac-charter.html

[12] R. Akkiraju, et al., WSDL-S Web services Semantics – WSDL-S, W3C Member Submission, http://www.w3.org/Submission/WSDL-S/

[13] WSMO Web service Modeling Ontology (WSMO), W3C Member Submission, http://www.w3.org/Submission/WSMO/