Published in: Alt, R., Fähnrich, K.-P., Franczyk, B. (Eds.), Proceedings First International Symposium on Services Science ISSS'09, Logos, Berlin, 2009 (www.logos-verlag.de)

A Service Description Language for the Internet of Services

Jorge Cardoso, Matthias Winkler, Konrad Voigt

SAP Research, Chemnitzer Strasse 48, D-01187 Dresden, Germany

{jorge.cardoso|matthias.winkler|konrad.voigt}@sap.com

Abstract: Services currently govern economies and will unquestionably become even more significant in the near future. This trend is supported by the launch of a proposal for a Directive on Services in the Internal Market[2] from the European Commission. Information and communication technology are presently being explored to provide infrastructures to support the notion of the Internet of Services which will enable providers to sell and consumers to purchase services. Such infrastructures will require research and development of new theories, concepts, models and technologies. As such, in this paper, we present a Universal Service Description Language (USDL), an approach to model service descriptions to enable the publication, discovery, selection, contracting and monitoring of service.

1 Introduction

Services currently govern economies and will unquestionably become even more significant in the near future. Outsourcing is one of the main reasons behind the growing number of service available since it allows companies to concentrate on their core competencies, reduce costs and take advantage of highly specialized external expertise. IBM, for example, which is a main producer of goods, has generated in 1998 more than half its revenues from services [IBM 1998]. The European directive for services[1] promises to increase the trade of services in the future.

In business, a service is the non-material equivalent of a good. It is considered to be an activity which is intangible by nature which is provided by a service provider to a service consumer to create a value possibly for both parties. Services normally provide a human value in the form of work, information, advice, skills and expertise. In traditional economies, services are typically discovered and invoked manually, but their realization maybe performed by automated or manual means (or a combination of both). Services can also be defined as a diverse group of economic activities not directly associated with the manufacture of goods, mining or agriculture [OECD 2000]. Examples of services include *hair cutting*, *house painting* or *letter typing*.

[2] http://ec.europa.eu/internal_market/services/services-dir/proposal_en.htm

Services can be categorized according to their economic area. For example, engineering services, healthcare services, telecommunications services, distribution services and retailing services. We use the term universal services to refer to any type of services independently of their economic area. The term Internet of Services (IoS) [Schroth 2007] refers to the infrastructure that enables the provision of universal services to consumers. The IoS describes an infrastructure that uses the Internet as a medium for offering and selling services. As a result, services become tradable goods. Service marketplaces, where service consumers and providers are brought together to trade services and engage in business interactions, are a fundamental building block for the IoS vision [Cardoso et al. 2008].

Services are considerably different from products primarily due to their intangible nature. Most products can be described physically based on observable properties, such as size, color, and weight. On the other hand, services lack of concrete characteristics. Thus, services must often be defined indirectly in terms of the effects they have on consumers. Therefore, the IoS can only achieve its full potential when services can be described in a suitable form to enable their publication, discovery, selection, contracting and monitoring. Compared to Web services, the challenge goes beyond a technical description and adds the requirement to also describe business and operational aspects.

The remaining of this paper is structured in four main sections. In section 2 we describe the nature of universal services. This study and characterization provide a better understanding on the challenges that are needed to address in order to model universal services. Section 3 presents the description language devised to represent universal services. In section 4, we discuss the approach that was taken to formally model services. Section 5 shows the importance of establishing a close relationship between service description languages with service level agreements. Finally, section 6 presents our conclusions.

2 The Nature of Universal Services

Compared to Web services [Curbera et al. 2001], developing solutions for the IoS is more elaborate since services are generally intangible, often inseparable, immersive, bipolar, variable, ostensible with respect to ownership, have long-running interactions and are decoupled.

(1) Intangible. Services are intangible since they do not have a material existence. As a result, it is difficult to create suitable standards to model them and to define attributes to objectively measure them. One of the main questions that this paper will answer is: *what are the fundamental aspects and characteristics of universal services?*

(2) Inseparable. The execution and consumption of services occurs frequently in parallel. This implies that a rigorous match between supply and demand must be

achieved. This leads to a challenging question: *how can the IoS provide description mechanisms to match between supply and demand efficiently?*

(3) Immersive. Services are often executed in collaboration with consumers. This implies that in many cases it is difficult to determine the parties responsible for the degree of success or failure of a service. Therefore, *when distributed services are invoked and executed using process models and involve providers and consumers, how can SLA be specified and monitored?*

(4) Bipolar. Services are often executed by a blend of human and technological resources. Solutions to monitor human involvement in services' execution and the complex relationship between the human and technological dimensions have not been studied in the context of Internet services. As a result, the following question arises: *how to create universal monitoring mechanisms that account for the monitoring of technological resources with the individual monitoring of human resources?*

(5) Variable. Products have a high degree of standardization, while services are very often tailor-made. The variations between similar products of different producers are less prominent than the variations between services. The following question arises, *how to describe the high variability of services?*

(6) Ostensible ownership. The ownership between products and services is distinct. Typically, when a product transaction is completed, the ownership is transferred to the consumer. On the other hand, it is not possible to own a service. Its possession is termed as an ostensible ownership. The following question arises, *how to represent at a given time the ostensible ownership of a service?*

(7) Long-running interaction. Services are often executed by a back-end business process which involves human interaction over time until the service is completed. For example, a service contracted to translate a book from German to English may run for several weeks and require a significant interaction between the translator and the writer. Therefore, services may require more personal contact between the provider and consumers. *How can long-running interactions involving relationships between people, processes and activities be associated with services?*

(8) Decoupled. The lifecycle of any service includes four main phases: *discovery, selection, invocation* and *execution* [Cardoso/Sheth 2005]. In order to capture the full potential of services, consumers must have access to dynamic discovery mechanisms. Once a set of services is discovered, a selection is made and the selected service is invoked. Finally, the service is executed. These four phases can be carried out only with human involvement, with a conjunction of humans and automated devices, or resorting purely on automated machines. *How can the phases of the lifecycle of services be described and represented?*

The first step to enable the development of technological infrastructures to support the concept of the IoS is to study how the most relevant characteristics and particularities of universal services can be abstracted and formally modeled. Such an abstraction will enable the formalization and normalization of the intangible, often

inseparable, immersive, bipolar, variable, ostensible with respect to ownership, long-running interactions and decoupling of universal services. Therefore, this paper presents a conceptual structure to model universal services. The language proposed to describe services is called Universal Service Description Language (USDL).

3 Universal Service Description Language

Products have usually a well defined set of possible variants for customization. For example, if a consumer requires a faster laptop, a more powerful CPU can be designed, built and attached to the motherboard. If a consumer (e.g. Yellow Cab Co.) desires yellow cars, a manufacturer only needs to notify the production chain to select a new color. The same cannot be easily achieved for services. This makes the description of services one of the most important undertakings for the IoS. While Web services (e.g. SOAP/WSDL or REST Web services) are usually seen mainly as technological entities, the IoS will also embrace what we call universal services and requires combining and correlating business, operational and IT aspects into service descriptions.

3.1 Limitation of WSDL for the IoS

The Web Service Description Language (WSDL) was developed to describe the technical details of how a Web service can be accessed and invoked remotely over the Web. It details technical requirements such as Internet addresses, ports, method names, arguments, and data types used by a Web service. The emphasis of WSDL is on technical and implementation aspects of services. WSDL was made to be used by computers. USDL has a different goal since it is also to be used in the IoS by people and organizations.

The IoS has different requirements from the ones fulfilled with WSDL. While the technical description of services is important for SOA, the business and operational perspectives on services have a significant importance for the IoS. Therefore, the USDL aims at bridging the business, operational and the technical perspectives. The business description includes the formal specification of legal, marketing and bundling aspects. The operational description includes functional and behavioral characteristics, and resource requirements. Finally, the technical description specifies how a service can be invoked and relies on references to WS-* protocols.

3.2 Enabling the IoS with USDL

With the proliferation of services in marketplaces as a business solution for enterprises and consumers in general, the features of services offered will become of the highest importance. A better description of the business and operational perspectives will bring to a marketplace an advantage over competitive platforms by being an added value for service providers and consumers. USDL enables to describe business characteristics exposed by an organization for the purpose of providing a way for consumers to invoke and use services. The USDL schema defines three core clusters of information that provide descriptions that a consumer can use to discover, select, invoke services and have a view on services' behavior at execution time. These three groups are the business, operational and technical clusters. Figure 7 shows an overview of the USDL meta-model. As it can be seen from the Figure, USDL has a strong emphasis on business and operations. The technical perspective is reduced.

3.3 Business, Operations and Technical Perspectives

USDL brings together the business, operational and technical perspectives. The *business perspective* describes properties that are fundamental for the characterization of a service. We rely on a set of non-functional properties such as availability, payment, pricing, obligations, rights, penalties, bundling, security and quality [O'Sullivan et al. 2005]. In order to provide a suitable language that can be understood by business stakeholders and consumers, the properties have been clustered into seven groups (each group is called a sub-perspective and sub-perspectives contain properties): roles (providers and consumers), service level, marketing, legal, interaction, bundling and an extension mechanism.

The *operational perspective* describes the operations executed by services. It provides an understanding of what the service is providing from an operational perspective and, thus, what a consumer can expect from a service. Important aspects modeled include operations, functionality, classifications, milestones and phases. USDL approach to the functional description of services is multifaceted since it allows using natural language, keywords (i.e. tagging) and ontologies as fundamental structures to express the functionality of a service. This perspective includes concepts borrowed from the area of project management. For example, phases allow creating groupings to provide a high level description of the business process associated with a service. This implicit process description using phases can serve as a basis for service discovery and indirect functional description and indicate the achievement of an important stage. Milestones provide a way to express the major states that a service will reach during its execution.

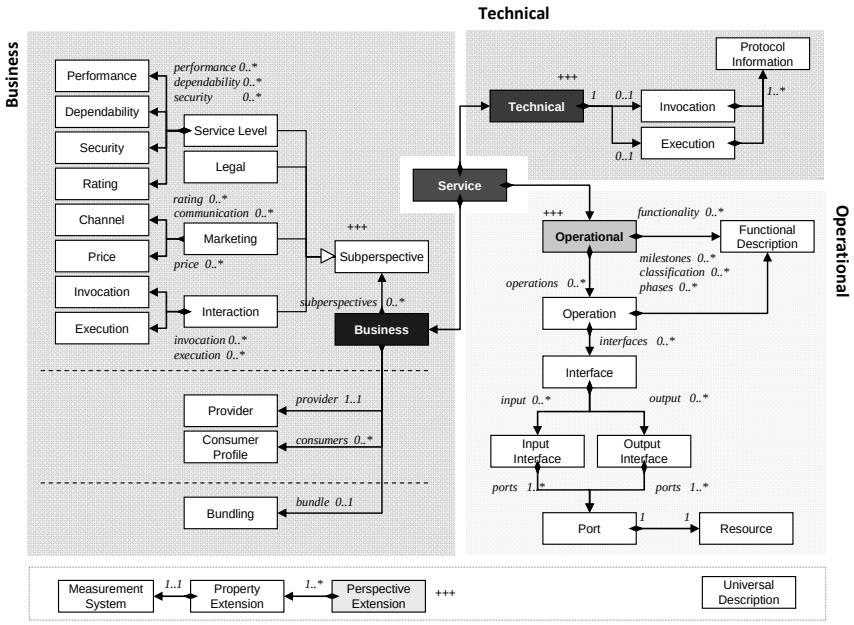

Figure 7: Formal Model Behind the USDL

The *technical perspective* allows specifying technical information of services exposed by an organization. Since the lifecycle of services include several phases, this perspective is divided into two main sections: invocation and execution. The first section describes how the invocation with the service is done. The second section describes how the interaction with the executing service is done. This perspective acts as a central point that allows to reference existing Web standards in order to describe technical aspects of services such as interfaces (e.g. WSDL), user interfaces, and communication, messaging and transaction protocols.

4 USDL Meta-model

In order to establish a proper base for USDL we provide a formal specification. This serves for purposes such as communication and implementation for integration with other specifications. Additionally, it refines the definition of the concepts presented. Therefore, we define and formalizing a meta-model for USDL.

Formalizing a language can be done in several ways. Some examples are: a meta-model, a grammar, an ontology, a XML schema, an implementation in a specific programming language, an algebraic specification, etc. We decided to define a meta-model and will advocate in the following paragraphs why a meta-model-based USDL specification is an appropriate solution. While addressing the term meta-model we refer to the OMG specification of the Meta Object Facility

(MOF). In MOF, a meta-model consists of concepts of the object-oriented world in order to formalize the static structure of a language. These concepts are packages, classes with attributes and operations, data types, inheritance as well as references and associations in order to express relationships between classes. For the definition of constraints, MOF is accompanied by the Object Constraint Language for expressing invariants for a given context (set of elements of a meta-model).

[Fischer et al. 2004a], [Fischer et al. 2004b] stated that a grammar, in contrast to a meta-model, lacks modularization and specialization as well as of a notation of inter language relation for an integration of existing languages. They argue that a modern language specification must offer more than a pure syntax definition, but also show the internal structure and support relations to other language specifications. This is supported by a meta-model approach on specifying a language and it is one of the reasons for our option to use a meta-model.

Furthermore, we have chosen a meta-model approach because it grants a formal and matured solution for expressing a language's syntax (static structure). Since USDL is a descriptive language, a meta-model is a good option for representing it. That is because a descriptive language does not cover behavioral aspects of a system to model and does not impose the combination of concepts in order to construct information to be modeled. In other words, it can be seen as a definition of a form to be filled. Furthermore, the supporting technology should provide facilities for the development of tools for USDL as well as the definition of an exchange format for persisted service descriptions. This is covered by MOF implementations. Additionally, the following points drove our decision:

(1) MOF-based meta-models are a matured, well-understood and established technology.

(2) MOF is an OMG standard which allows an easy integration of other standards such as UML, BPMN, etc.

(3) The integration of other languages (existing data and models) such as process languages and schema languages is supported.

Especially the last point is addressed by SAP using the *Modeling Infrastructure* (MOIN) [Altenhofen et al. 2006]. MOIN is a MOF-based repository implementation, which is designated to provide a common infrastructure for SAP models. The infrastructure also covers a design tool development environment. MOIN is similar to the *Eclipse Modeling Framework* (EMF, http://www.eclipse.org/emf) which also provides a variety of supporting tools and frameworks.

We base our work on MOIN to allow the integration with existing data residing in SAP systems such as process models or services definitions of the *Enterprise Service Repository* (ESR) (a repository based on the UDDI specification, http://uddi.xml.org). Besides the integration aspect, MOF also provides a mature base for tool generation and implementation. This allows a rapid prototyping, i.e. by rapidly developing editors for the creation of USDL instances for testing purposes. Test instances can be used to validate USDL and discover conceptual problems.

We have formalized USDL with a MOF-based meta-model (Figure 7) to represent its formal base. We pursued the goal to keep the model as simple as possible. We made use of composition and structuring of the elements, but avoided grouping mechanism and generalization, since this reduces the comprehensibility of a model. Building upon MOF, as an established OMG-standard with implementations available such as MOIN and EMF, we believe this will provide a base for an implementation of USDL in terms of integration and tools.

5 From USDL to Service Level Agreements

After discussing the structure and formalization of USDL, we will now describe how service level agreements (SLA) can be created from a USDL-based description instance. Service level agreements are formal contracts between service consumers and providers negotiated prior to service provisioning. They serve as a base for monitoring the provisioning and consumption of the service, which is necessary to assure a trustful business interaction between the involved parties [Winkler et al. 2008].

5.1 Specifying Service Level Agreements

A number of different approaches for specifying SLA exist (e.g. Web Service Level Agreements (WSLA) [IBM 2003], SLAng [Lamanna et al. 2003], WS-Agreement [Global Grid Forum 2005]). WSLA and SLAng are not being developed any further. WS-Agreement is a specification from the Open Grid Forum [Global Grid Forum 2005]. It defines a language and protocol for the offering of capabilities by service providers, the negotiation of agreements between service consumers and providers, and for monitoring the compliance to these agreements. While the WS-Agreement language provides a structure for SLA documents it does not specify which aspects of a service are described and how. This needs to be handled by a specific language for service description. WS-Agreement facilitates the negotiation procedure and enables us to use our own service description language, namely USDL, for specifying the aspects and characteristics of the service.

WS-Agreement also specifies a process for creating SLA. Service consumer and provider take the roles of agreement initiator and responder. The agreement initiator requests an agreement template from the responder. Based on this template an offer is created and sent back to the responder who then validates and accepts or rejects it. In this section we describe how to generate an initial agreement template while the negotiation process is out of scope.

An agreement template consists of three main sections: the agreement context, the terms, and the creation constraints section. The agreement context specifies information about the involved parties and their roles. The terms section is used to

describe what the service will provide and a number of guaranties for its execution. The creation constraints section specifies rules for the creation of a valid offer from the template.

5.2 USDL to WS-Agreement Mapping

The WS-Agreement template (Figure 8) can be generated from the USDL service description via a transformation. While some information available in the service description can be mapped to WS-Agreement elements, it is necessary to extend WS-Agreement templates by using USDL statements. General information on a service and the functionality it provides is presented in the *ServiceDescriptionTerms* section. This section includes the name of the service, its version number, and a functional classification. Further information such as a service ID, other classifications or bundling information can be added but was omitted due to space limitations. It is important to emphasize that the *ServiceDescriptionTerms* section contains USDL statements from the service description since WS-Agreement, as mentioned above, does not provide the means for describing services but requires a suitable service description language.

The *ServiceProperties* section is used to define further measurable service attributes. All measurable attributes contained in the USDL description along with their metric are mapped to the WS-Agreement *Variable* element. The service's execution time is specified in our example. In contrast to the *ServiceDescriptionTerms* section where USDL code fragments are integrated into the template code we have a mapping from USDL to WS-Agreement.

Finally, the *GuaranteeTerm* section is used to set up specific *ServiceLevelObjectives*, e.g. min, max, average, or concrete values which are guaranteed for service provisioning. They can be specified for each service attribute listed as *Variable* in the *ServiceProperties* section. The information of concrete values for the service attributes in USDL is mapped to the *CustomServiceLevel* element. In our example, the *executionTime* variable, which was defined in the *ServiceProperties* section, is referenced. A *ServiceLevelObjective* is specified which guarantees that *executionTime* is two hours maximum. As in the *ServiceProperties* section we defined a mapping to WS-Agreement instead of using USDL statements.

To extend the template with a *CreationConstraints* section, additional modeling beyond the current scope of the USDL is necessary. The section can be used to specify a value range for a specific parameter as well as certain relationships between different parameters. We will capture *CreationConstraints* at a later point of time within the project.

```
<wsag:Template>…
  <wsag:ServiceDescriptionTerm wsag:Name=„EcoInf" wsag:ServiceName=„Eco Calc">
     <usdl:ServiceDescription>
      <usdl:ServiceName>Eco Calculator</usdl:ServiceName>
      <usdl:Version> v.1.5.3</usdl:Version>
      <usdl:Classification>
        <usdl:ClassificationName>UN/SPSC</usdl:ClassificationName>
        <usdl:ClassificationValue>12322122</usdl:ClassificationValue>
      </usdl:Classification>
    </usdl:ServiceDescription>
  </wsag:ServiceDescriptionTerm>
  <wsag:ServiceProperties wsag:ServiceName=„Eco Calculator">
    <wsag:VariableSet>
      <wsag:Variable  wsag:Name=„executionTime“  wsag:Metric=„xsd:duration“>…
      </wsag:Variable >
    <wsag:VariableSet>
  </wsag:ServiceProperties>
  <wsag:GuaranteeTerm wsag:Name=„ExecutionTime_GUARANTEE„ monitored=„true“>
    <wsag:ServiceScope  wsag:ServiceName=„Eco Calculator"/>
    <wsag:ServiceLevelObjective>
      <wsag:KPITarget>
        <wsag:KPIName>Execution_Time</wsag:KPIName>
        <wsag:CustomServiceLevel>P2H</wsag:CustomServiceLevel>
      </wsag:KPITarget>
    </wsag:ServiceLevelObjective>
  </wsag:GuaranteeTerm>…
</wsag:Template>
```

Figure 8: WS-Agreement Template with USDL Elements

6 Conclusions

In this paper we have presented the Universal Service Description Language (USDL) as a language for describing business, operational, and technical aspects of universal services. Such description languages will be fundamental for the success of the Internet of Services. The USDL accounts for the specific characteristics of universal services while at the same time preserving means for describing aspects of accepted Web service standards, e.g. WSDL and BPEL. USDL has a formal specification created using a MOF-based meta-model. Besides the advan-

tages of a formal specification, such as simplified communication and precise semantics, we pursue the goal of integrating existing Web-based models and take advantage of existing technologies. Using a MOF-implementation, such as MOIN infrastructure from SAP, existing services and workflows can be easily integrated into USDL using reference mechanisms. Furthermore, USDL models can be published within SAP systems, so existing models and tools can access and use the information and vice versa. We have also presented a mapping between USDL and WS-Agreement showing that USDL service descriptions can serve as a base for specifying service level agreements. WS-Agreement SLA templates can be created via a transformation from USDL descriptions.

Acknowledgments

The TEXO project was funded by means of the German Federal Ministry of Economy and Technology under the promotional reference 01MQ07012. The authors take the responsibility for the contents.

References

Altenhofen, M., Hettel, T., Kusterer, S., OCL Support in an Industrial Environment, in: Proceedings 6th OCL Workshop at the UML/MoDELS, 2006.

Cardoso, J., Sheth, A., Introduction to Semantic Web Services and Web Process Composition, in: Cardoso, J., Sheth, A. (Eds.), Semantic Web Process: Powering Next Generation of Processes with Semantics and Web Services, Springer, Heidelberg 2005, pp.1-13.

Cardoso, J., Voigt, K., Winkler, M., Service Engineering for The Internet of Services, in: Filipe, J., Cordeiro, J., Cardoso, J. (Eds.), Enterprise Information Systems, Springer, Berlin 2008.

Curbera, F., Nagy, W., Weerawarana, S., Web Services: Why and How, in: Proceedings Workshop on Object-Oriented Web Services - OOPSLA 2001, Tampa (FL) 2001.

Fischer, J., Holz, E., Prinz, A., Scheidgen, M., Tool-based Language Development, in: Proceedings Workshop on Integrated-reliability with Telecommunications and UML Languages, Rennes 2004.

Fischer,J, Piefel, M, Scheidgen, M., Using Metamodels for the Definition of Languages, in: Proceedings 4th SDL and MSC Workshop (SAM04), Ottawa 2004.

Global Grid Forum, Web Service Agreement Specification (WS-Agreement), Version 2005/09.

IBM Corporation, Web Service Level Agreement (WSLA) Language Specification, Version 1.0, January 2003.

IBM, Annual Report 1998, http://www.ibm.com.

Lamanna, D.D., Skene, J., Emmerich, W. Specification Language for Service Level Agreements, EU IST 34069 Deliverable D, March 2003.

OECD, The Service Economy, Science Technology Industry Business and Industry Policy Forum Series, Paris 2000.

O'Sullivan, J., Edmond, D., Hofstede, A.H.T., Formal Description of Non-functional Service Properties, Technical FIT-TR-2005-01, Queensland University of Technology, Brisbane 2005.

Schroth, C., Janner, T., Web 2.0 and SOA: Converging Concepts Enabling the Internet of Services, in: IT Professional, (2007) 3, pp. 36-41.

Winkler, M., Cardoso, J., Scheithauer, G., Challenges of Business Service Monitoring in the Internet of Services, in: Proceedings 10th International Conference on Information Integration and Web-based Applications & Services (iiWAS2008), Linz 2008.