GENERATING SEMANTIC DESCRIPTIONS OF WEB AND GRID SERVICES

Marian Babik,[1] Ladislav Hluchy,[1], Jacek Kitowski,[2], Bartosz Kryza,[3]
[1]*Institute of Informatics, Slovak Academy of Sciences, Slovakia*
[2]*Institute of Computer Science, AGH University of Science and Technology, Poland*
[3]*ACK Cyfronet AGH, Poland*

Abstract Web Service Resource Framework (WSRF) is a recent effort of the grid community to facilitate modeling of the stateful services [11]. Design and development of the WSRF service based systems is quite common and there are several emerging WS initiatives, which tries to automate the process of discovery, composition and invocation of such services. The semantic web services are a typical example, showing the potential of how ontological modeling can improve the shortcomings of the service oriented computing. One of the major obstacles in the process is the development of the ontologies, which describe web and grid services. Although, there are numerous standards for modeling semantic services, there are very few frameworks and tools, which can help automate the process of generating the semantic descriptions of services. This article presents a tool, which can semi-automatically generate the OWL-S descriptions for both stateful and stateless services based on the Web Service Description Language (WSDL) and corresponding annotations. Such functionality is inevitable in the grid environment hosting a vast number of services, which have to be semantically described in order to enable automated discovery, composition and invocation.

Keywords: grid services, semantic grids, web services, wsrf, owl-s

1. Introduction

Recently, Web service (WS) technologies are gaining importance in the implementation of distributed systems, especially grids. One such example is the Web Service Resource Framework (WSRF) [11], which extends the current WS technologies by modeling the stateful services. Design and development of the service oriented distributed system is quite common and there are several emerging WS initiatives, which tries to automate the process of discovery, composition and invocation of services. The semantic web services are a typical example, showing the potential of how ontological modeling can improve the shortcomings of service oriented computing.

In this paper we will introduce basic concepts of semantic web services (OWL-S) and web service resource framework (WSRF). Further, we will present the process of adding semantics to the stateful services and highlight the major issues, that we have faced during the development of the WSRF2OWL-S tool. We will also describe the corresponding architecture of the system and provide an illustrating use case of its functionality.

2. Web Ontology of Services (OWL-S)

OWL-S is an ontology-based approach to the semantic web services [5]. The structure of the ontology consists of a service profile for advertising and discovering services, a process model which supports composition of services, and a service grounding, which associates profile and process concepts with underlying service interfaces (see Fig. 1). Service profile (OWL-S profile) has functional and non-functional properties. The functional properties describe the inputs, outputs, preconditions and effects (IOPE) of the service. The non-functional properties describe the semi-structured information intended for human users, e.g. service name, service description, and service parameter. Service parameter incorporates further requirements on the service capabilities, e.g. security, quality-of-service, geographical scope, etc. Service grounding (OWL-S grounding) enables the execution of the concrete Web service by binding the abstract concepts of the OWL-S profile and process to concrete messages. Although different message specifications can be supported by OWL-S, the widely accepted Web Service Description Language (WSDL) is preferred [9].

3. Adding Semantics to the Stateful Services

Service annotation is the process of generating the semantic descriptions (i.e. OWL-S) of both stateless and stateful services from the web service descriptions (i.e. WSDLs). In K-Wf Grid it has become crucial in the process of providing application support and enabling semantics for semantically unaware grid application areas [14]. In the following we will present the issues that we have faced during the design of the ontologies for the stateful services and we will briefly describe the developed annotation tool called WSRF2OWL-S.

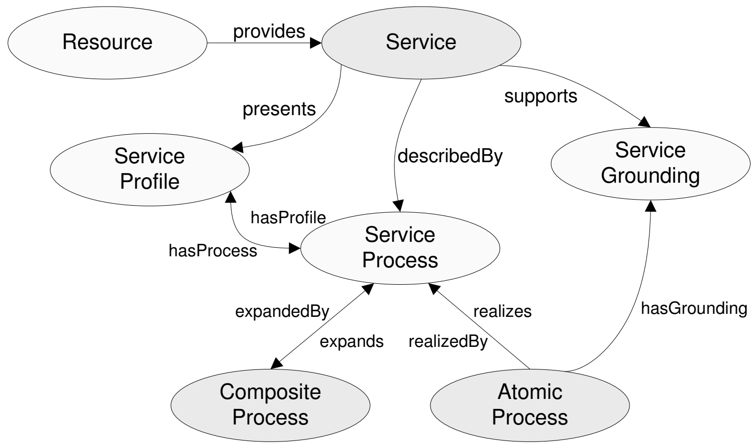

Figure 1. OWL-S concepts.

WS-Resource semantics

In OWL-S, the service capabilities are described by the corresponding IOPE (inputs, outputs, preconditions and effects). Such description can be partly annotated from the WSDL description of the service and is sufficient for the stateless service. Stateful services, i.e. WS-Resources, are, however, composed of a service and a stateful resource [11]. A stateful resource is defined by a single XML Global Element Declaration (GED) in a given namespace. Such GED defines the type of the stateful resource and is motivated by the modeling of complex objects for stateless services. For modeling semantics of the stateful resources this means there are no major differences between stateful and stateless services, i.e. the domain concepts can be derived from the complex types definitions in the service description. There are, however, few important issues, that should be noted.

Resource properties (RP), as defined in the WSRF specification, can be dynamic. This means that it is possible to create and destroy properties on the fly, i.e. if stateful service is providing access to the filesystem and resource properties are listing the file attributes, it is possible to add or remove a file attribute at any given time. In order to model such dynamic behavior by corresponding semantics, it is necessary to consider more advanced techniques for ontological mapping and concept definition. Since such procedures can become quite complicated, we have concentrated our work on the area of static resource properties, i.e. it is possible to change the values of the properties, but the set of properties for given resource remains constant over time.

In the process of designing the stateful services it is possible to use inheritance of the resource properties. Although explicit hierarchy of the resource properties can help in the generation of the semantics, there is no standard, which would describe in details how RP inheritance should be implemented. This can cause major difficulties in parsing of the services and it is necessary to introduce special parsers to extract the RP hierarchies. Furthermore, resource properties can often be used to model the actual inputs and outputs of the service, i.e. a service submitting specific jobs to the cluster can represent the inputs and outputs as properties of the job, thus hiding the inputs/outputs in the resource properties of the service. This has to be considered in the service composition.

Apart from the issues in the process of service discovery, there are also slight differences in the process of service invocation. The WS-Resource is composed of a service and a stateful resource, i.e. it is identified by the so called EPR (end point reference), which describes not only service address but also the identification of a resource. The service identification in the OWL-S Grounding has to be then extended to a more complex structure. For the grid services such extension can also consider the possibility of having multiple instances of the same service hosted by different servers. This can, however, be solved simply by introducing multiple OWL-S Groundings.

Generating OWL-S from web service descriptions

We have designed and developed a tool for generating the OWL-S description for stateful and stateless services from the corresponding web service descriptions (WSDLs) [9]. Such tool is inevitable in the grid environment hosting a vast number of services, which have to be semantically described in order to enable automated discovery, composition and invocation. In the initial stage of the K-Wf Grid project we have successfully used the tool to create semantic descriptions of the services for the flood forecasting domain.

The architecture of the so called WSRF2OWL-S tool is shown in Fig. 2. The main components of the architecture are WSRF2OWL-S engine, translator and GOMOWL-S API. GOMOWL-S API is an extension of the Mindswap's OWL-S API [5]; it defines additional vocabulary, converters and extensions needed by the WSRF (e.g. SimpleEffect, DataObjectInput, etc.). The translation procedure is quite complex and covers

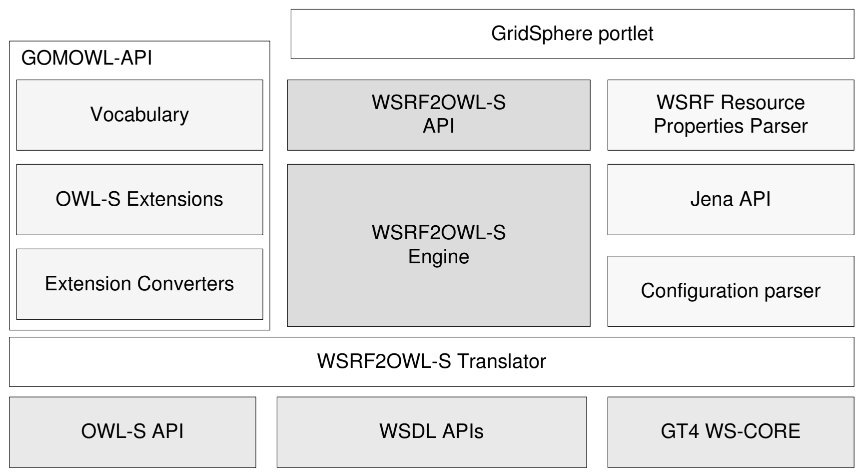

Figure 2. Architecture of the WSRF2OWL-S tool.

the areas already described in previous sections. The translation starts with a configuration and an URL of the WSDL document. The translator parses the WSDL document extracting the operations, port-types, inputs, outputs as well as resource properties. A combination of the WSDL4J [22], Axis WSDL [23] and Globus Toolkit WSDL utilities [12] are used in the process. The translator then generates for each WSDL operation a skeleton of the OWL-S document. Then it creates the inputs, outputs, preconditions and effects and maps the elements to the ontological concepts defined in the configuration. If needed, it will create an ontology, which models the resource properties of the given services. The GOMOWL-S API can be used to extend the OWL-S by the domain dependent constructs, e.g. FloodForecastingWSRFProfile, DataObjectInput, SimpleEffect, etc. The outcome of the process is OWL-S document describing the web service operations, which are then be composed into the workflow [19]. Additionally a GridSphere portlet was developed to provide a graphical user interface for the tool [17]. It supports browsing of the concepts for any given ontology, associating the concepts with the WSDL elements and generation of the OWL-S documents. An automated annotation procedure based on the case-based reasoning is also integrated.

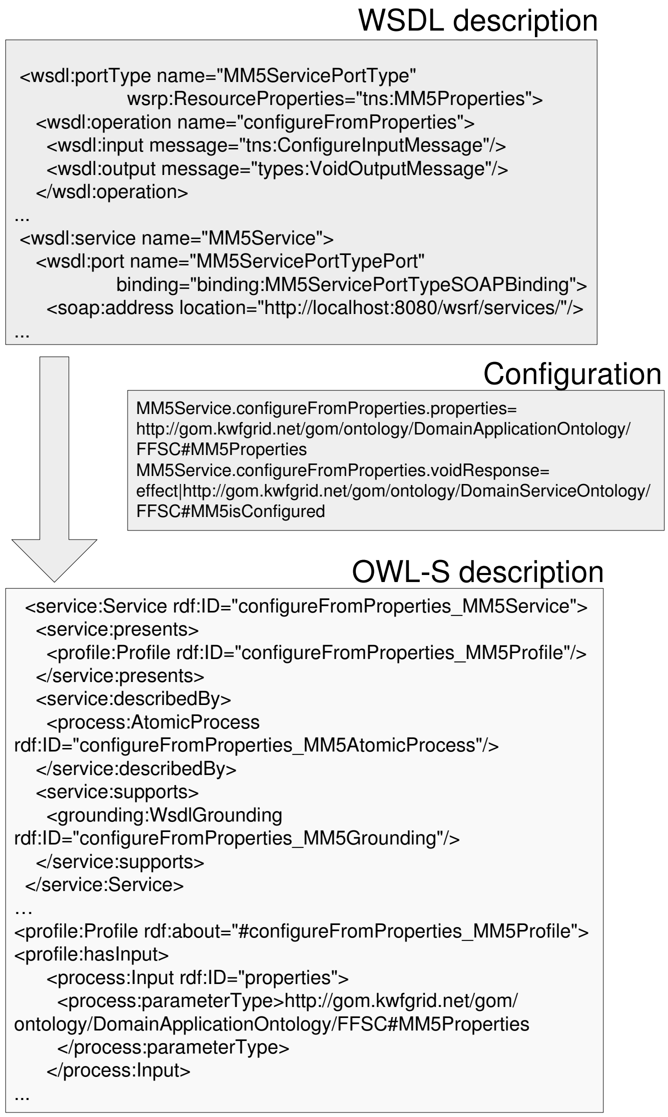

Figure 3. Sample translation of the MM5 configureFromProperties method

4. Application Scenario

The flood forecasting application (FFSC) is based on a network of loosely coupled, cooperating but independent services. It has been used as a pilot application in the project K-WfGrid and CrossGrid [21, 13]. The application consists of three major components, namely, meteorology, hydrology and hydraulics. Each component has several possible models of computation represented by the corresponding web or grid services, e.g., meteorological methods Aladin, MM5; hydrological methods HSPF, NLC, etc.

The translation of the WSDL description of a sample meteorological service MM5 is shown in Fig. 3. Since each service has multiple operations, the semantic descriptions are generated for each operation, thus enabling the possibility to create workflows of service operations [19]. In the example a sample operation configureFromProperties is shown. Apart from WSDL description the translation process also needs configuration, which describes the mapping of the WSDL inputs/outputs to the domain ontological concepts. These concepts can describe information about service (e.g. service name, provider), but also complex inputs and outputs of the service, such as geographical location, geographical information data, watershed, etc. Further they are used to identify the HTML forms, which are presented to the user if additional input is necessary.

Apart from flood forecasting simulations, WSRF2OWL-S was successfully used in generating the semantic description of services for the enterprise resource planning and coordinated traffic management applications [14]. In the future we would like to concentrate on improving the configuration capabilities and broader WSRF support.

5. Related work

One of the challenges of the loosely coupled distributed systems is the ability to dynamically discover and integrate the services needed by the applications. Interoperability among services is especially important in the distributed environments hosting a large number of services, i.e. grids. Semantic descriptions facilitates the process by expressing the characteristics of the service, which is one of the goals of the Semantic Grid initiative [1]. There are many projects, which are trying to develop an architecture for the Semantic Grid such as [2, 4, 3]. S-OGSA is trying to extend OGSA based architecture and provide a reference architecture with explicit handling of semantics as well as defining the associated knowledge services. Guided by the set of design principles it defines a model, the capabilities and mechanisms for the Semantic Grid [2]. InteliGrid aims at developing a grid architecture based on three layers, i.e. conceptual, software and basic resource [3]. Unlike our approach the mentioned projects are trying to address the Grid semantics by a top-down approach, creating reference architectures, which should cover a broad range of applications and requirements. In contrary, WSRF2OWL-S can be seen as a bottom-up approach, which is trying to leverage as much as possible from the existing Semantic Web Service technologies. A similar approach can be seen in the ^{my}Grid, which is a pioneering Semantic Grid project, providing a set of tools and services to enable workflow composition in biological domain [4]. It is however more focused on the support for the OGSA and OGSA-DAI, while we aim at supporting WSRF and OWL-S, which have shown to be more suited for the domain of the K-Wf Grid applications.

The actual transformation process for the stateless services is provided by two WSDL2OWL-S tools [5, 7]. Both tools are based on the corresponding OWL-S API libraries and provide either web-based or graphical user interface. WSRF2OWL-S extends these tools with support for the WSRF service descriptions as well as possibility to use command line tool for batch processing multiple descriptions at once. In terms of service annotation, ASSAM is one of the existing automated semantic web service annotators with machine learning capabilities [8].

In the domain of Semantic Web Services, the Web Service Modeling Framework (WSMF)[15] is an industry scale framework for semantic web service discovery, execution and composition. It is a joint effort of the European research projects on the Semantic Web and Semantic Web Services. It has three development areas concerning conceptual model (WSMO), the representation language (WSML) and the execution framework (WSMX). Although WSMF approach is much more profound and shows many significant contributions to modeling semantic web services, the level of implementation and the development support at the time of evaluation was unacceptable for our purposes. However, since WSMF has gained an enormous momentum in the last year, it will be considered in our future work.

The Internet Reasoning Service (IRS-III) [18] is a Java framework for publishing, locating, composing and executing semantic web services. IRS-II is modeling service based on the tasks, that need to be fulfilled and the problem-solving-methods (PSM) that can be used to solve specific tasks. It utilizes a formal language called Operational Conceptual Modeling Language (OCML). It supports the specification and operationalization of functions, relations, classes, instances and rules. This appears to be more suitable for procedural knowledge representation than OWL. One of the disadvantages of the system is the process of assigning services the corresponding tasks, which has to be done manually, which introduces many possibilities for failure.

METEOR-S [16] attempts to add semantics to the basic stateless web service descriptions by adding semantics to current industry standards. It is an effort of the LSDIS Lab of the University of Georgia. The framework presents an interesting bottom-up approach to the semantic descriptions, the service annotation and WSDL-S, semantic extension of the WSDL.

Acknowledgments

Acknowledgments: The research reported in this paper has been partially financed by the EU within the project IST-2004-511385 K-WfGrid and Slovak national projects, APVT-51-024604; Tools for acquisition, organization and maintenance of knowledge in an environment of heterogeneous information resources, SPVV 1025/04 and VEGA No. 2/6103/6.

References

[1] Goble, C., De Roure, D., "The Semantic Grid: Myth Busting and Bridge Building", in Proceedings of the 16th European Conference on Artificial Intelligence (ECAI-2004), Valencia, Spain, 2004

[2] Alper, P., Corcho, O., Kotsiopoulos, I., Missier, P., Bechhofer, S., Goble, C., S-OGSA as a Reference Architecture for OntoGrid and for the Semantic Grid, GGF16 Semantic Grid Workshop. Athens, Greece. February 2006

[3] Turk, Z., Stankovski, V., Gehre, A., Katranuschkov, P., Kurowski, K., Balaton, E., Hyvarinen, J., Dolenc, M., Klinc, R., Kostanjsek, J. and Velkavrh J., "Semantic Grid Architecture," 2004

[4] C. Wroe, C. A. Goble, M. Greenwood, P. Lord, S. Miles, J. Papay, T. Payne, and L. Moreau, "Automating Experiments Using Semantic Data on a Bioinformatics Grid," IEEE Intelligent Systems, vol. 19, pp. 48-55, 2004.

[5] A. Ankolekar et.al, OWL-S: Semantic Markup for Web Service, 2003, http://www.daml.org/services/owl-s/1.1

[6] Mindswap OWL-S API, http://www.mindswap.org/2004/owl-s/api/

[7] CMU OWL-S API, http://www.daml.ri.cmu.edu/wsdl2owls

[8] A. He and E. Johnston and N. Kushmerick, ASSAM: A tool for semi-automatically annotating semantic web services, In proceedings of the 3rd International Semantic Web Conference, 2004, Springer

[9] E. Christensen, F. Cubera, G. Meredith, S. Weerawarana, Web Services Description Language (WSDL) 1.1, Technical report, WWW Consortium, 2001

[10] Resource Description Framework, http://www.w3.org/RDF/

[11] Web Service Resource Framework, http://www.globus.org/wsrf/

[12] Globus Toolkit, http://www-unix.globus.org/toolkit/

[13] CrossGrid consortium, CrossGrid Technical Annex, 2004, http://www.crossgrid.org

[14] The Knowledge-based Workflow System for Grid Applications FP6 IST project, http://www.kwfgrid.net

[15] Fensel D. and Bussler C., The Web Service Modeling Framework WSMF, Eletronic Commerce: Research and Applications, 1, 2002

[16] P. Rajasekaran and J. Miller and K. Verma and A. Sheth, Enhancing Web Services Description and Discovery to Facilitate Composition, International Workshop on Semantic Web Services and Web Process Composition, 2004

[17] GridSphere portal framework, http://www.gridsphere.org/gridsphere/gridsphere

[18] Motta E. and Domingue J. and Cabral L. and Gaspari M., IRS-II: A Framework and Infrastructure for Semantic Web Services, 2nd International Semantic Web Conference (ISWC2003), Sundial Resort, Sanibel Island, Florida, USA, 2003

[19] Gubala, T., Bubak, M., Malawski, M., Rycerz, K., Semantic-based Grid Workflow Composition, In: Proc. of 6-th Intl. Conf. on Parallel Processing and Applied Mathematics PPAM'2005, R.Wyrzykowski et.al. eds., 2005, Springer-Verlag, Poznan, Poland

[20] Hoheisel, A., User Tools and Languages for Graph-based Grid Workflows. In: Special Issue of Concurrency and Computation: Practice and Experience, Wiley, 2005

[21] Hluchy, L., Tran, V.D., Habala, O., Simo, B., Gatial, E., Astalos, J., Dobrucky, M., Flood Forecasting in CrossGrid project. In: Grid Computing, 2nd European Across Grids Conference, Nicosia, Cyprus, January 28-30, 2004, LNCS 3165, Springer-Verlag, 2004, pp. 51-60, ISSN 0302-9743, ISBN 3-540-22888-8.

[22] IBM WSDL4J Project, http://oss.software.ibm.com/developerworks/projects/wsdl4j

[23] Apache WebServices - Axis Project, http://ws.apache.org/axis/