Semantic Grid Services in K-Wf Grid

Marian Babik, Emil Gatial, Ondrej Habala, Ladislav Hluchy, Michal Laclavik
Martin Maliska
Department of Parallel and Distributed Computing, Institute of Informatics,
Slovak Academy of Sciences
Bratislava, Slovakia

Abstract

In this paper we present design and development of the semantic grid services for the flood forecasting simulations. We will highlight the corresponding architecture and the process of service annotation, discovery and composition in the project K-WfGrid [12]. We will describe in detail the challenges of the flood-forecasting application and corresponding design and development of the service oriented model, which is based on the well known Web Service Resource Framework (WSRF). Semantic descriptions of the WSRF services will be presented as well as the architecture, which exploits semantics in discovery and composition of services. Further, we will demonstrate how experience management solutions can help in the process of discovery and composition. The system provides a unique approach in Semantic Grids by combining the advances of semantic web services and grid architectures.

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) [10], 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 present the process of design and development of the flood-forecasting simulation services and the process of their translation to the semantic descriptions. We also briefly describe a corresponding architecture for discovery, composition and invocation of both stateful and stateless services. We provide a brief overview of the Web Service Resource Framework specification (WSRF) and Web Ontology for Services (OWL-S) [8] and show how can the stateful service be described in terms of the OWL-S specification. Furthermore, we present the process of assisting the user in the composition and discovery of the services by using the experience management system.

2. K-Wf Grid Architecture

Fig. 1 presents an architecture of the system components of the workflow orchestration and execution environment in the project K-Wf Grid. The main user interface for developing semantic-based Grid applications is the User Assistant Agent (UAA), which contacts the Grid Workflow Execution Service (GWES) that manages the process of composing and executing the services. The automated semantic service composition is partly delegated to the Automatic Application Builder (AAB), the Workflow Composition Tool (WCT), and the user (by means of the UAA). The Automatic Application Builder and the Workflow Composition Tool are knowledge-based semi-automatic modeling services, which in cooperation with the User Assistant Agent can propose known solutions to problems solved in the past. The semi-automatic composition of the services is enabled by the semantic description of the grid services, which is the main responsibility of the WSRF2OWL-S part of the Grid Organizational Memory (GOM). When parts of the workflow are ready to be executed on the Computing Grid, the Grid Workflow Execution Service asks the Scheduler for the optimal resource, due to some user-defined metrics. Then, the corresponding Web Service operation is invoked remotely on the Grid middleware using WSRF protocols. The events triggered by the workflow orchestration and execution will be published by means of an event system. The Knowledge Assimilation Agents (KAA) consume these events and generate knowledge that is stored in the Grid Organizational Memory (GOM).

Workflow Composition Tool (WCT) [15, 16] provides the functionality of composing abstract workflows of Grid

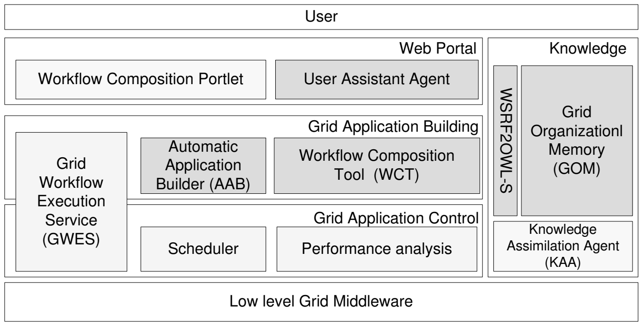

Figure 1. A simplified schema of the K-Wf Grid architecture.

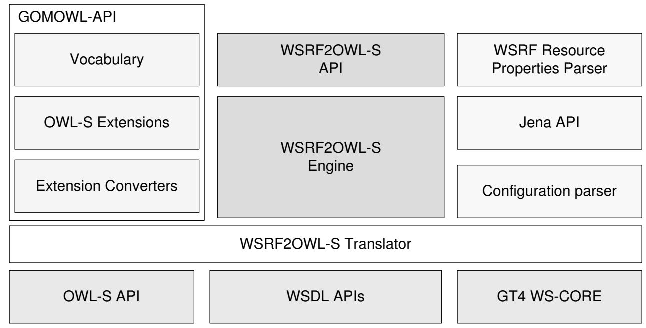

Figure 2. Architecture of the WSRF2OWL-S tool.

applications from simple user requirements. It employs semantic reasoning techniques over OWL-S descriptions (i.e. subsumption, classification) and it tries to propose a solution to the user's problem by using provided descriptions of available resources. Such a solution is delivered in form of an abstract workflow instance composed of service operations. This workflow of operations is based on the Petri nets model, which has several advantages over the directed acyclic graphs, for a detailed description see [16]. The main input to the WCT is a description of data (results) which is to be produced by the target application and, optionally, a set of user-provided data (input) to be used by the composed application. It is also possible to upload an incomplete workflow as an input in order to complete it automatically. The main output of the composition process is a refined description of the abstract workflow. During the composition and refinement of the workflow the User Assistant Agent is used to guide the user according to the experience it gained in the past compositions.

3. Adding Semantics to the Stateful Services

3.1. 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, enterprise resource planning and coordinated traffic management domains.

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-API, which provides the interface to the Grid Organizational Memory (GOM). 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 Jena API [7] and Globus Toolkit WSDL utilities [11] 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 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 are OWL-S documents describing the web service operations, which are then be composed into the workflow as described in the next section.

4. Experience management in the discovery and composition of services

Knowledge and experience management [17] is known more from area of information systems and organizational process management. However, we believe that such approach can be used also in the area of web service composition. When services are composed automatically, several composed workflows can be presented to the user. According to available semantic description such workflows can be viewed as identical for the user problem. For example, if we compose services to predict weather forecast in Bratislava this can be fulfilled with MM5 or Aladin meteorology service due to semantic description. However, one of the models does not have to give good results for certain geographical location or in certain season, and others can be more appropriate. Such knowledge can not be put to semantic description for all cases but can be easily captured from expert users in form of text notes while the system is used. These notes (human experience) are processed using semantic annotation and the system detects semantic context (ontology concepts and individuals) of the note which is reviewed and confirmed by user. Such note with assigned context can thus be displayed in future to the user in similar context.

Use of semantic annotation is important for appropriate notes context detection and thus helping service composition when displaying relevant experience in concrete user context. Annotation is also used for appropriate service discovery and to help user specify problem using free text which is translated to semantic description of the problem. Discovered semantic elements (user requirements) are then used by the system to compose services to fulfill the user problem. The main idea of the used annotation algorithm is to detect relevant structured knowledge described by a domain specific ontology model in the unstructured text.

4.1. Example Application Scenario

In this section, we present a scenario, in which a user states a problem to be solved, a workflow solution is created and executed. When user logs into the system, he/she starts by using the User Assistant Agent interfaces. He/she enters a text description of the problem to be computed. This description is then analyzed for known keywords and detected elements are presented to the user to confirm detected context of the problem. When some elements of the problem context are confirmed by the user, they become the semantic representation of user's problem and composition of the services can start. Detected ontology elements are from the service ontology generated by the WSRF2OWL-S as well as from other domain ontologies related to the input and output data, such as geographical ontology of the target area (in case of geographically bound simulations, for example).

After the problem context and semantic description is stated, the system creates a so-called abstract workflow, consisting only of unknown input, one high-level, abstract task (transition) and the defined output (the solution). This abstract workflow is then analyzed, relevant services capable of creating the solution are found, and a more concrete workflow is created by backward-chaining the respective inputs and outputs. Afterwards the abstract transitions are replaced by several classes of services. These are not yet actual service instances and calls, but rather representations of certain capabilities, which are known to be found in one or more real service instances. At this point, the system is ready to start execution of the workflow, which means that the system starts looking for real service instances, which are able to perform the tasks represented by the class transitions. These service instances are evaluated by the Knowledge Assimilation Agent based on their previous monitored behavior, and the instance believed to perform best (according to a selected metric, for example speed) is then executed. If the system is unable to find service instance for the class transition, user's attention is required. Also, the system is able to recover from a fault of the selected service instance, and to use another instance, possibly working one.

After the constructed workflow is successfully computed, the user may view the output data. There is also the possibility of incorporating one or more users into the workflow, by using special User Proxy Service. This service is provided with a visualization of the data the user should see (created by a specialized service, preceding the User Proxy Service in the workflow). It sends a notification to the supposed user (via e-mail or ICQ, currently) with a URL that he/she should visit and review in order to continue the process of executing the workflow.

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, 5]. There are many project, which are trying to develop an architecture for the Semantic Grid applications such as [2, 4, 3, 6]. 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.

In the domain of Semantic Web Services, the Web Service Modeling Framework (WSMF)[13] is an industry scale framework for semantic web service discovery, execution and composition. 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. METEOR-S [14] attempts to add semantics to the basic stateless web service descriptions by adding semantics to current industry standards. The framework presents an interesting bottom-up approach to the semantic descriptions, the service annotation and WSDL-S, semantic extension of the WSDL.

6. Conclusions

The project K-Wf Grid provides a system, that is able to aid users in using the service-based grid, it is using workflow management of application services, semantic description of the infrastructure as well as the target application domain, monitoring of all interaction with the grid and reasoning on the monitored events in order to be able to predict behavior of the grid in future workflow execution. The project has successfully implemented a prototype of the system, which is available online. It moves now toward the final system, which will consist of a set of independent modules (most of them implemented as web or grid services themselves) that will together be able to implement the idea of constant performance optimization through knowledge acquisition and reuse.

7. 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, RAPORT APVT-51-024604 and NAZOU SPVV 1025/04.

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] H. Zhuge, Semantic Grid: Scientific Issues, Infrastructure, and Methodology, Communications of the ACM. 48 (4)(2005)117-119.

[6] H.Zhuge, China's E-Science Knowledge Grid Environment, IEEE Intelligent Systems, 19 (1) (2004) 13-17.

[7] Jena A Semantic Web Framework for Java, http://jena.sourceforge.net/

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

[9] E. Christensen, F. Cubera, G. Meredith, S. Weerawarana, Web Services Description Language (WSDL) 1.1, Technical report, WWW Consortium, 2001, http://www.w3c.org/TR/wsdl

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

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

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

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

[14] 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

[15] 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

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

[17] Ralph Bergmann, Experience Management: Foundations, Development Methodology, and Internet-Based Applications, Lecture Notes in Artificial Intelligence, ISBN 3540441913, 2002