Semantic Tools for Workflow Construction

Ondrej Habala, Marian Babik, Ladislav Hluchy,
Michal Laclavik, and Zoltan Balogh

Departement of Parallel and Distributed Computing, Institute of Informatics, Slovak Academy of Sciences
Ondrej.Habala@savba.sk, Marian.Babik@saske.sk, hluchy.ui@savba.sk,
Michal.Laclavik@savba.sk, Zoltan.Balogh@savba.sk

Abstract. In this paper we present design and development of a knowledge framework for grid and web service-based workflow composition and execution. We highlight the corresponding architecture and the process of service annotation, discovery and composition in the project K-WfGrid [5]. We describe in detail the challenges of a 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 are presented as well as the architecture, which exploits semantics in discovery and composition of service workflows. Further, we demonstrate how experience management solutions can aid the process of collaborative service discovery and composition. The whole K-Wf Grid 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 of WS implementation is the Web Service Resource Framework (WSRF) [3], 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 architecture and several interesting implementation details of a knowledge-based framework for workflow management in a service-based grid environment. The framework is based mainly on the knowledge of semantics of the environment, and the process of application workflow execution is driven by this knowledge. We therefore also briefly describe a corresponding architecture for discovery, composition and invocation of both stateful and stateless services and we provide a brief overview of the Web Ontology for Services (OWL-S). We show how a stateful service can be described in terms of

V.N. Alexandrov et al. (Eds.): ICCS 2006, Part III, LNCS 3993, pp. 980–987, 2006.

the OWL-S specification. Furthermore, we present the process of assisting the user in the composition and discovery of the services by using an experience management system based on text notes [13] as well as on instance based learning.

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 Agent (KAA) consumes these events and generates knowledge that is stored in the Grid Organizational Memory (GOM). This knowledge can be later reused by the components of the workflow orchestration and execution environment.

Workflow Composition Tool (WCT) provides the functionality of composing abstract workflows of Grid 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 [7]. 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.

In the following we will describe more closely the WSRF2OWL-S translation, describing the semantics of the WSRF services, the actual implementation and

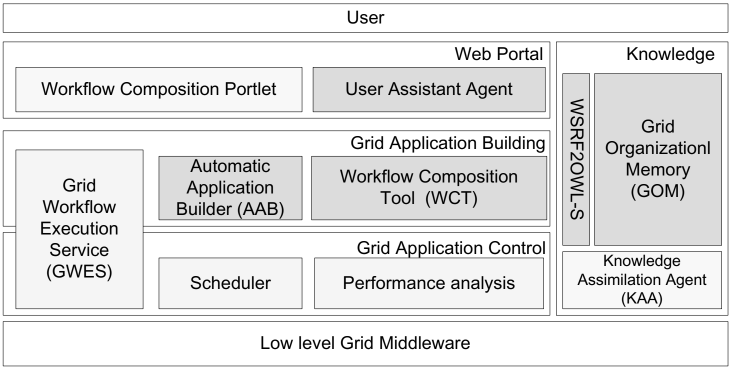

Fig. 1. A simplified scheme of K-Wf Grid architecture

use case of the system operation based on the flood-forecasting application. We will also describe the User Assistant Agent (UAA) and its role in the process of composition and refinement of the workflow. A detailed description of the Workflow Composition Tool and the Grid Workflow Execution Service can be found in [6, 7].

3 Adding Semantics to Stateful Services

3.1 Web Ontology for Services (OWL-S)

OWL-S is an ontology-based approach to the semantic web services [1]. 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. 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 [2].

3.2 Semantic Annotation of Services Based on OWL-S

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. We have developed an annotation tool called WSRF2OWL-S. During its development we have faced several issues, mainly problems caused by the dynamic nature of resource properties of the WS-Resource standard and the complexity of stateful services and their interaction model.

3.3 WSRF2OWL-S Tool

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) [2]. 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.

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 [8], Axis WSDL [9] and Globus Toolkit WSDL utilities [4] 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 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 [10, 11] 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. Based on 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 may not give good results for certain geographical location or in certain season, and others can be more appropriate. Such knowledge can not be put into semantic description for all cases but can be easily described by 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 actual 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. The main difference between existing annotation solutions such as Anotea [12] and EMBET is detection of ontology elements from existing domain ontology (other annotation solutions try to create such ontology).

The Knowledge Assimilation Agent (KAA) is used as another tool which enables knowledge to be exploited in Grid workflows and in Grid collaborative environments. The approach used in KAA deals with behavior prediction of Grid services. The performance measures of Grid services are estimated based on the past cases. Instance based learning (IBL) is used to estimate the performance of Grid services. We elaborate generic IBR methods by case retrieval refinement process through semantic description of discrete features and service input data. Case representation is crucial for capturing information about WS operation invocations. We model the case structure using ontology. In our approach a single case represents a single WS operation invocation (please see figure bellow).

Sample KAA scenario can be presented by the flood forecasting application (see Chapter 5) used in K-Wf Grid as a prototype demonstration. KAA is used to discover dependencies between execution time of a Grid service and the area for which the forecast is being computed. The geographic area is stored in a file (resource), which is semantically described. Ontology which describes the geographical area is used for refinement of case retrieval during performance prediction of the forecasting WS. Past grid service invocations give us information how long will a computation last for a given geographic area. Thus a user willing to submit a flood forecast for a certain area can estimate the length of forecast to be computed.

5 Service Based Flood-Forecasting Simulation Cascade

The flood prediction application (FFSC) is based on a a network of loosely coupled, cooperating but independent services. The K-Wf Grid's flood application became instantiated in several servers across the testbed. It is a set of loosely coupled services, with several possible execution scenarios. Main data provider and also external input to the cascade is the Slovak Hydrometeorological Institute (SHMI). The SHMI provides us with input data for the first stage of our application, the *Meteorology*. In this stage, we employ two models, *ALADIN* and *MM5*, for weather predictions, the latter having three distinct operation modes (simple, one-way nested and two-way nested). The predicted weather conditions are used in the *Watershed integration* stage to compute water runoff into the target river. This result is then further processed in the *Hydrology* stage, where two models - *HSPF* and *NLC* - compute river levels for selected geographical points. These levels are then used to model water flow in the last, *Hydraulic* stage of the application. Concurrently all important results are optionally visualized, packaged and displayed to the user - if required.

Apart from the simulation models, preprocessor and associated tools, the data flow contains also several *job packagers* and a *User Proxy Service*. These services implement our approach towards interactive grid jobs and also toward *in-process user collaboration*. The user proxy service may receive a ZIP file (prepared by a visualization service) with a HTML sub-tree, which is displayed to a certain user. The user is then notified of his/her new task (currently by e-mail, later also ICQ notifications will be implemented) and may view the HTML sub-tree, included images, animations, tables, etc. and may even react and provide input by filling out HTML forms (which are further processed by following services in the application workflow). For each specialised task another user may be asked to provide input, thus collaborating on a bigger job, requiring expertise of several users. Apart from enabling seamless multi-user collaboration on a single workflow, the concept of the User Proxy Service enables (with its asynchronous notifications) users to leave long-running workflows unattended and return to them on request, either to view the computed results or to provide some input.

5.1 Example Application Scenario

When a user logs into the system, he/she starts by using the User Assistant Agent interfaces (see Fig. 2). 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. Free text problem definition is important, when a user starts to work with the system and wants to define the problem in a way understandable for automatic service composition.

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, in which the abstract transition is 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 (see the light gray boxes in Fig. 2).

At this point, the system is ready to start execution of the workflow. When the user clicks the play button in the workflow visualization (lower left part of Fig. 2), the system starts looking for real service instances, which are able to perform the

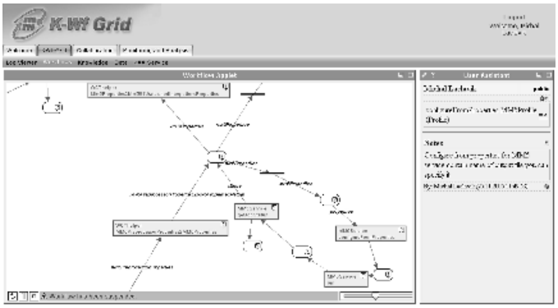

Fig. 2. K-Wf Grid portal with User Assistant pane on the right, workflow visualization on the left

tasks represented by the class transitions. These service instances are evaluated by the KAA based on their previous monitored behaviour, 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.

The User Assistant pane has another important role in the workflow execution process. To aid user in service selection, input data provision and general orientation in the world of web and grid services, it provides description of the workflow elements, if such description (note) has been entered previously by other user. This is yet another form of experience management, this time based on the text notes passed between users. These text notes are entered in a form of a line (or several lines) of text through a button in the User Assistant pane. After the note is entered, the context of the currently selected element of the workflow is analysed, verified by the user and the note is bound to this semantic context. Then the User Assistant is able to find the note later, if a similar context of a workflow element is present in the workflow, and the note may be displayed and may possibly guide the user with previous user's experience. These notes may be used to describe certain special qualities (or deficiencies) of some service classes or instances, such as ability/inability to work under certain conditions or to provide quality results for some tasks.

6 Conclusion

The presented K-Wf Grid system is primarily intended to simplify composition and execution of service-based workflows. It also offers several useful collaboration capabilities, which make it very usable in teams composed of several domain experts and application users. The system is able to learn from previous successes and failures. It can be used for a wide range of applications, from long-running simulations to interactive grid applications, requiring user input at several places of their workflow. The applications can be easily extended and the system is able to use both stateless web services as well as the new stateful WSRF-based grid services.

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, Research and development of a knowledge based system to support workflow management in organizations with administrative processes, APVT-51-024604; Tools for acquisition, organization and maintenance of knowledge in an environment of heterogeneous information resources, SPVV 1025/04.

References

1. A. Ankolekar et.al, OWL-S: Semantic Markup for Web Service, 2003, http://www.daml.org/services/owl-s/1.1
2. 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
3. Web Service Resource Framework, http://www.globus.org/wsrf/
4. Globus Toolkit, http://www-unix.globus.org/toolkit/
5. The Knowledge-based Workflow System for Grid Applications FP6 IST project. http://www.kwfgrid.net
6. 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
7. Hoheisel, A., User Tools and Languages for Graph-based Grid Workflows. In: Special Issue of Concurrency and Computation: Practice and Experience, Wiley, 2005
8. IBM WSDL4J Project, http://oss.software.ibm.com/developerworks/projects/wsdl4j
9. Apache WebServices - Axis Project, http://ws.apache.org/axis/
10. Thomas H. Davenport, Laurence Prusak, Working Knowledge, ISBN 1578513014, May, 2000
11. Ralph Bergmann, Experience Management: Foundations, Development Methodology, and Internet-Based Applications, Lecture Notes in Artificial Intelligence, ISBN 3540441913, 2002
12. Annotea Project, http://www.w3.org/2001/Annotea/, (2001)
13. Laclavik, M., Gatial, E., Balogh, Z., Habala, O., NGuyen, G., Hluchy, L., Experience Management Based on Text Notes (EMBET), eChallenges Conference, 19 - 21 October 2005, Ljubljana, Slovenia ,2005