Knowledge-Based Grid Workflow System

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

Department of Parallel and Distributed Computing, Institute of Informatics,
Slovak Academy of Sciences
Marian.Babik@saske.sk, Ondrej.Habala@savba.sk, Ladislav.Hluchy@savba.sk,
Michal.Laclavik@savba.sk

Abstract. In this paper we present design and development of the grid workflow system based on the semantic grid services. We will highlight the process of the discovery and composition of grid and web services into workflows and semantic technologies supporting the users during the workflow composition and evaluation of the results [13]. We will describe in detail the challenges of the design and development of the semantic grid service model, which is based on the Web Service Resource Framework (WSRF) and the Web Service Ontology (OWL-S). Semantic descriptions of the WSRF services as well as mapping of such descriptions to the workflow model will be presented. Further, we will demonstrate how experience management solutions can help in the process of service discovery and user support. The system provides a unique bottom-up approach in the Semantic Grids by combining the advances of semantic web services and grid architectures [1].

1 Introduction

Recently, knowledge-based technologies are gaining importance in the implementation of the distributed systems and grids. Especially interesting is the design and development of the service-oriented workflow systems, which tries to automate the process of discovery, composition and execution of the services. One such example is the semantic web services effort, showing the potential of how ontological modeling can improve the shortcomings of the service-oriented computing. In this paper we present a user-friendly knowledge-based workflow system, which can support complex user interactions during the composition of the workflow, e.g. automated composition of dynamic workflows based on the user queries, storage and retrieval of the past workflows and results based on the current user context. The system is based on our previous work [18] and tries to reduce the complexity of the grid-based workflow systems by exploiting the semantic descriptions of services, user context and experience.

The structure of the paper is follows: In Sections 2 and 3 we described the underlying semantic and workflow technologies used in the system. Section 4 presents our approach in mapping the workflow model to the semantic web services and the semantic descriptions of user context and experience. We conclude with a flood-forecasting application scenario demonstrating the benefits of the system and description of related work.

Y. Shi et al. (Eds.): ICCS 2007, Part III, LNCS 4489, pp. 265–272, 2007.

2 Web Ontology of Services (OWL-S)

OWL-S is an ontology-based approach to the semantic web services [9]. 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. In the following we will denote the WSDL operations as O_i and input and output messages as M_{in}^j, M_{out}^j respectively.

3 Workflow Model

The services and the middleware for discovery, composition and execution is based on the Web Service Resource Framework (WSRF) [10], a recent standard for the grid services. Since we do not rely on the single service, but on the set of services, it is necessary to describe the application in terms of the *workflow*, which represents a control and data flow of the corresponding services. The representation of the workflow should be easy to use and intuitive, however it should also be able to cope with the complexity of the Grid, i.e. large number of services, dynamic scheduling, automatic handling of resource properties, etc.

We have relied on the workflow model based on the High-level Petri Nets (HLPN), which allows to compute the output tokens of a transition from the input tokens [17]. In this model service operations are represented as transitions. Each service operation is represented by the *transition* T_i, denoted by a thick vertical line. The variables of the formal parameters are represented as input, output *places*, P_{in}^j, P_{out}^j, shown as empty circles. Each place can hold a number of *tokens*, which represent the data items of the parameters. The input and output parameters of the operation are shown as input and output *edge expressions*, respectively. Additionally, each transition can have a set of Boolean *condition* functions. Firing transition and thus calling the service operation is only allowed if all of its conditions evaluate to true. Given the individual service representation, we can describe the complex workflows by connecting the output places of one transition with input places of another transition. All of the mentioned concepts and relations are defined in the form of an XML-Schema as Grid Workflow Description Language (GWorkflowDL). A detailed description of the

GWorkflowDL and High-level Petri Nets is beyond the scope of this document and can be found in [17].

4 Semantics

4.1 Overview

In order to support the complex user interactions, automated discovery, composition and execution of service operations, it is necessary to provide a mechanism for semantically describing the set of services, users, application domains and grid environment [6]. Semantic descriptions together with the related Semantic Web technologies can then allow to automatically discover services based on the user query, compose the services into the workflow, optimize the execution of the service operations and present the results in a clear and understandable way. Further, it can support collaboration among users, possibility to store and re-use the important results and workflows gathered by previous runs.

The knowledge captured by the semantic descriptions in the form of ontologies has two major areas: semantic description of services allowing to discover, compose and execute the workflows and semantic description of user context and experience. In the area of semantic service descriptions we have used the upper ontology of services OWL-S [9]. In the area of the user context and experience we have relied on the ontological descriptions and mechanism defined by the knowledge and experience management. In the next section we will describe the service descriptions and its mappings to the workflow model as well as the issues we faced during the extension of the OWL-S standard to the grid services. In Section 4.3 we will present our approach in describing the system context and experience.

4.2 Semantic Description of Grid Services

Our description of services has three layers: *grid layer*, represented by the WSDL operations and messages, *workflow layer*, represented by the GWorkfloDL elements and *ontological layer*, represented by the OWL-S and OWL ontologies (see Fig.1). The corresponding mapping between grid and workflow layer is quite straightforward. Each workflow transition T_i represents one WSDL operation O_i, whose input/output messages $M^j_{in/out}$ are mapped to input/output tokens $P^j_{in/out}$. Ontological layer provides semantic description of services in terms of service *profile* and *grounding*. Service *process* is atomic thus leaving the workflow description to the *workflow layer*. Service *profile* functional properties, i.e. inputs, outputs, preconditions and effects, are described by the corresponding domain concepts, e.g. flood-forecasting concepts - river, basin, location, watershed. Non-functional properties are described by service name, service providers, quality of service metrics and service classes (hierarchy of services). Service *grounding* maps the functional properties to the actual WSDL messages. Since we already have such information in the *workflow layer*, we provide service *grounding* descriptions only for compatibility purposes. This means, that based on the service

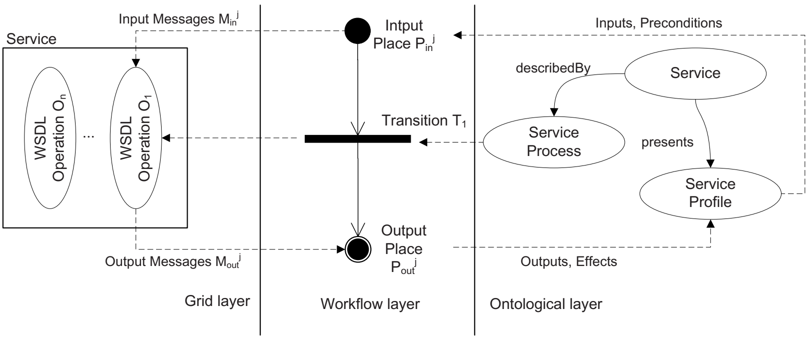


Fig. 1. Layers of the service description with corresponding mapppings

profiles, we can infer whether it is possible to connect inputs and outputs of any two services and thus create workflow [12]. Since OWL-S is based on the OWL-DL we can use any OWL-DL capable reasoner for this task. Another benefit of such approach is that we have a direct mapping between workflow transitions and service *profile.* [1]

4.3 Semantic Description of User Context and Experience

Capability to store and re-use knowledge is presented in several system requirements such as possibility to store and re-use previous workflows and results, guide the users during the workflow composition according to previous experience or creating workflows based on user queries. These requirements reflect the fact that in the system with many services and users, everyone can benefit from the previous cases experienced by others. An example of such functionality is creation of a workflow from a simple query such as, flood-forecasting for tomorrow for the basin of Danube river. The answer to such query is a list of services together with list of notes describing each service an its suitability for the requested area and time. Upon selection of the concrete service the system should be able to either compose the workflow or show all fitting workflows computed previously by other users. It should also suggest the location of available results for the given query. In order to provide such functionality it is necessary to find a suitable technology, which can semantically describe the user context and experience.

Semantic description of user context and experience allows to determine similarity and relations among different contexts experienced by other users. This is the domain of the knowledge and experience management [19], where such functionality is defined as a capability to collect lessons from the past associated to cases. Experience of the user is modeled as a Case-Lesson pair (C, L), where C denotes the *Case Space* and L denotes the *Lesson Space*. The Case Space C

[1] The described mapping is implemented by the tool available under EDG license at: http://http://www.tuke.sk/fei-cit/babik/wsrf2owls/

is usually represented as a set of vectors c, whose elements determine the user context. The Lesson Space L is a set of notes (i.e. free text), workflows and results, which have the form of some global URI. The elements of the user context vector c are simply instances of the ontological concepts, i.e. instances of the domain concepts such as concrete time, city, basin or instances of the service concepts, i.e. service descriptions.

Given the Case-Lesson pair database and the current user context vector c_u, the system can compare its similarity to the previous cases and list the most relevant Lessons L to the user. The similarity of the two vectors c, c_u is determined by using an ontological similarity metric based on similarity graphs [20].

The actual build-up of the current user context vector c_u is based on user queries and actions in the user interface, e.g. user query: "Flood-forecasting for tomorrow in Bratislava" would result in the context having the following instances $(DaveF2DVisualization, Bratislava, Danube, 241206)$; the vector contains instances of the concepts Service, City, River, Date, which were determined from the user query. As previously described, evaluating the current user context vector c_u results in the list of lessons from previous cases. Based on the user actions in this list, the user context can be further extended/updated, e.g. by clicking on the note describing the service *DaveF2DVisualization*, the system can present other similar services and update the context based on next user decisions. If the user is interested in the concrete workflow or result, its URI is then used to enact the workflow engine and either run the workflow or present its results. A special case is the possibility to compose the workflow based on the identified service (as was described in Section 2).[2]

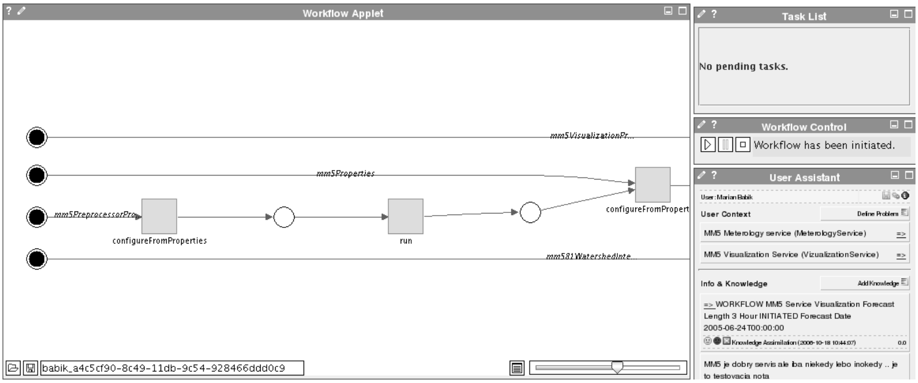
5 Application Scenario

In this section we present a scenario which showcases the functionality described in previous sections[3]. The scenario is hosted by the K-WfGrid user portal shown in Fig. 2 [13]. The portal has only three main components: control panel, user assistant and workflow panel. The *user assistant* guides the user throughout the composition and execution of the workflow, which is shown in the *workflow panel*. The workflow can be started, stopped or suspended at any time by using the *control panel*, which also provides a list of tasks that the user needs to accomplish in order to make progress (e.g. provide input data).

The system allows the user to type a free text query, e.g. flood forecasting in Bratislava. The *user assistant* parses the query and determines the user context, e.g. in this case the list of services appropriate for the region and concepts reflecting the location and basin. This context is presented as a list of clickable concepts, which the user can select. Based on the user selection the *user assistant* displays the list of relevant lessons, e.g. past workflows, results and notes

[2] An implementation of the described approach is available at http://www.ikt.ui.sav.sk/?page=software.php

[3] A captured demo session can be found at http://www.gridworkflow.org/kwfgrid/distributions/movie-09-ffsc-red.avi



No pending tasks.

▷ ‖ □ Workflow has been initiated.

User : Marian Babik

MM5 Meterology service (MeterologyService) =>

MM5 Visualization Service (VizualizationService) =>

=> WORKFLOW MM5 Service Visualization Forecast Length 3 Hour INITIATED Forecast Date 2005-06-24T00:00:00

Knowledge Assimilation (2006-10-18 10:44:07) 0.0

MM5 je dobry servis ale iba niekedy lebo inokedy .. je to testovacia nota

Fig. 2. K-Wf Grid portal with *control panel* (top right), *user assitant* (bottom right) and *workflow panel* left. A sample workflow for wheather prediction is shown; grey boxes denoting the transitions/service operations; circles denoting places/service parameters.

submitted by previous users. The next steps vary based on the user actions. The user can browse through existing notes, workflows and results to see if someone has already computed any similar results or workflows. Based on the user selection the system will display the results and notes describing each service. By clicking on the workflow the system will load it into the *workflow panel* and the user can restart it with his/her data by using the *control panel*. Generally, the process of user guidance is based on the semantic descriptions of user context and experience as described in Section 4.3.

Another alternative path is followed when the user selects a concept, which describes a flood-forecasting method, e.g. DaveF2DVisualization. Such concept is related with the semantic description of the service hosting the method. Based on the semantic description of the service the system can automatically compose an abstract workflow and present it to the user. This process is based on the semantic description of services as described in Sec.4.2. This abstract workflow is then analyzed, relevant services capable of creating the solution are found, and a more concrete workflow is created, where the abstract transition is replaced by several classes of services. The user is presented by a list of tasks that he/she needs to fulfil in order to continue execution the workflow, e.g. specifying time and date for the forecast. The user can then start the workflow through *control panel*. Consequently, the system looks for real service instances and executes them. The process of instance lookup and execution is visualized in real-time in the *workflow panel*. After the workflow finishes the user can get the results by clicking on the corresponding output place in the *workflow panel*, e.g. in this case it would display an animation of the possible flooding in the region. The workflows and results can be annotated by the user during the whole process. The overall process of composition and execution is captured by the semantic description of context and experience is stored by the system for later retrieval. The benefits of the approach are mainly user-friendly interface with possibility to compose workflow based on user query, as well as user support and guidance throughout the process of composition and execution.

6 Related Work

There are many existing workflow-based graphical problem solving environments such as [3,5,4]. Our approach differs mainly in the exploitation of the user context and experience, which enables storing and re-using past workflows and results, creation of workflows based on queries and complex support for user interactions during the process.

In the domain of Semantic Grid there are many projects, which are trying to develop an architecture for the Semantic Grid applications such as [2,7,8]. 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. On the contrary, our approach 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 [7]. 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 our applications.

In the context of the *Semantic Web Services*, WSMO, WSMX and IRS-III provide an industry scale framework for discovery, composition and execution of web services [14]. A similar approach to ours is taken by the WSDL-S and METEOR-S, which attempts to add semantics to the basic web service descriptions [15]. Generally, none of the existing semantic web service frameworks provide support for composition and execution of the grid services or any possibility for user assistance based on experience or knowledge re-use.

7 Conclusions

We have described a working flood-forecasting system based on the semantic grid services. We have shown how it can compose and execute workflows while supporting complex user interactions and assisting technologies. The system is currently tested and evaluated by the Slovak Hydrometeorological Institute on the datasets of the major Slovak river basins.

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, NAZOU SPVV 1025/2004, RAPORT APVT-51-024604, VEGA 2/6103/6, VEGA 2/7098/27.

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. Taylor, I., Shields, M., Wang, I., Harrison, A., Visual Grid Workflow in Triana, Journal of Grid Computing, pp. 153-169, vol. 3, September 2005
4. Deelman, E., et.al., Pegasus: Mapping Scientific Workflows onto the Grid, Grid Computing, 2004, Springer, LNCS
5. Hategan, M., von Laszewski, G., Amin, K., Karajan: A Grid Orchestration framework, Supercomputing 2004, Pittsburgh, 6-12 November 2004.
6. Kryza, B., Slota, R., Majewska, M., Pieczykolan, J., Kitowski, J.: Grid organizational memory-provision of a high-level Grid abstraction layer supported by ontology alignment, Future Generation Computer Systems, Volume 23, Issue 3, March 2007, Pages 348-358
7. 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.
8. Li Zha, Wei Li, Haiyan Yu, Xianghui Xie, Nong Xiao, Zhiwei Xu: System Software for China National Grid. NPC 2005: 14-21
9. A. Ankolekar et.al, OWL-S: Semantic Markup for Web Service, 2003, http://www.daml.org/services/owl-s/1.1
10. Web Service Resource Framework, http://www.globus.org/wsrf/
11. Globus Toolkit, http://www-unix.globus.org/toolkit/
12. 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
13. The Knowledge-based Workflow System for Grid Applications FP6 IST project. http://www.kwfgrid.net
14. Fensel D. and Bussler C., The Web Service Modeling Framework WSMF, Eletronic Commerce: Research and Applications, 1, 2002
15. 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
16. 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
17. F. Neubauer, A. Hoheisel, J. Feiler: Workflow-based Grid application, Future Generation Computer Systems 22 (2006) 6-15
18. Hluchy, L., et.al., Flood Forecasting in CrossGrid project. In: Grid Computing, 2nd European Across Grids Conference, Nicosia, Cyprus, January 28-30, 2004, LNCS 3165, 2004
19. Ralph Bergmann, Experience Management: Foundations, Development Methodology, and Internet-Based Applications, Lecture Notes in Artificial Intelligence, ISBN 3540441913, 2002
20. Laclavik, M., Babik, M., Balogh, Z., Hluchy, L., AgentOWL: Semantic Knowledge Model and Agent Architecture. In: CAI journal,. Vol. 25, no. 5 (2006), p. 419-437.