Service-Oriented Architecture for Knowledge-enriched Workflows Modelling and Execution

Juan C. Vidal, Manuel Lama, and Alberto Bugarín

Department of Electronics and Computer Science
University of Santiago de Compostela
E15782 Santiago de Compostela, Spain
{jvidal,lama,alberto}@dec.usc.es

Abstract: This work presents a Service-Oriented Architecture that supports the management, execution and monitoring of knowledge-enriched workflows. This architecture is based *(i)* on a framework that allows the specification of workflows from knowledge components; and *(ii)* on the interfaces that are defined in the workflow reference model proposed by the Workflow Management Coalition.

1 Introduction

Organizations are facing the challenge of globalization and unprecedented levels of competitiveness. Market demands force organizations to find solutions in order to increment their productivity and to reduce their costs. A way to achieve these objectives is the improvement of their business processes, and *workflows* have proven to be a good solution for such purpose [vdAvH04]. In this context, workflows management system (WMS) [GHS95] facilitate the modelling of the processes as well as the communication, coordination and collaboration between their participants.

However, current WMSs have two major disadvantages: firstly, they do not model the knowledge that solves the problem. Instead of explicitly defining this knowledge, such as the knowledge used in a specific domain (the experience), it becomes part of the process definition. This makes the reuse of a part or the whole workflow complicated and even more when the domain changes; secondly, their integration and interoperability with the enterprise information systems requires a lot of programming efforts.

In this paper, we present a Service Oriented Architecture (SOA) [Erl05] and an infrastructure that provides the set of functionalities required by any WMS according to the workflow reference model defined by the Workflow Management Coalition (WfMC) [Wor99]. Within the proposed architecture, those functionalities are published as Web services that can be directly invoked by the clients that need to integrate workflows in their organizations [CNW01].

On the other hand, the proposed architecture has been defined on the top of a conceptual framework that allows the specification of workflows from a set of knowledge components [VLB06b]. This framework extends the traditional workflow modelling framework [vdA98] such that the control structure (process dimension) and the organization structure (resource dimension) that define a workflow are complemented with knowledge components: tasks, domain models, problem-solving methods [BF98] and ontologies (knowledge dimension). A better reuse and a richer definition of workflows is achieved through the use of this new modelling framework.

This paper is structured as follows: section 2 describes the SOA that supports the WMS and integrates the knowledge-based framework. In this section, we pay special attention to the web services that have been defined to cover the workflow life cycle. Section 3 describes the infrastructure that allows the execution of the knowledge-enriched workflows. Finally, section 4 presents work contributions and its future developments.

2 Service-Oriented Architecture

This section describes the SOA that gives the agents with the capability to manage, execute and monitor their knowledge-enriched workflows. We must mention that the proposed architecture has been influenced by the conceptual framework for modelling this kind of workflows [VLB06b], although this framework is out of the scope of this paper.

The proposed SOA is depicted in Figure 1. In this architecture a client, e.g. a software agent, accesses through SOAP to the UDDI registry in which the provider has published a set of Web services oriented to the management and use of knowledge-enriched workflows. The functionalities provided by these Web services are captured from the reference model specified by the WfMC [Wor99] which lists the interfaces that must be created *(i)* in order to define and manage the components that compose a workflow and *(ii)* to monitor and perform its execution. The following subsections will describe each one of these services and the elements of the architecture.

2.1 Services for the Management of Workflow Components

These services provide the functionalities included in the *process definition interface* of the reference model of the WfMC, and allow the management of each one of the three dimensions that define a knowledge-enriched workflow, i.e. the control structure, the organization structure and the knowledge:

- *Management of the control structure*. The definition of a workflow establishes a set of functional and data dependencies between the tasks that must be performed (process dimension). Since our workflow framework models these dependencies by means of high-level Petri nets [ISO02, Jen03], this layer of Web services provides the functionalities for the definition and management of this kind of Petri nets. In order to integrate the process and knowledge dimensions, the static and dynamic model of the high-level Petri nets has been described semantically through an ontol-

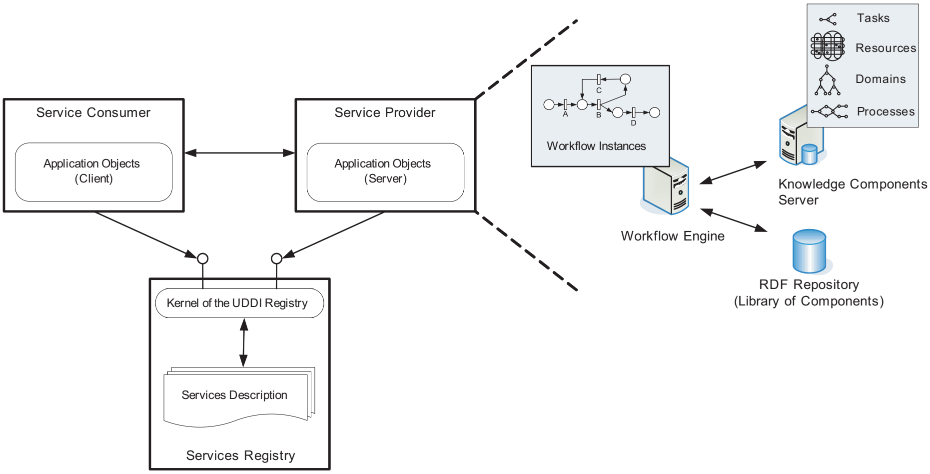

Figure 1: Service-oriented architecture that supports the workflow framework

ogy [VLB06a] expressed in the OWL language [DCvH+04]. Nevertheless, the Web services defined in this layer will handle BPEL4WS [CGK+02] and PNML [WK03] languages from which the control structures of the workflows will be introduced in the system.

- *Management of the organization structure*. A key aspect of a workflow is the definition of its organizational structure: it captures the roles, organizational units and resources of the organization. This structure is used to define *(i)* the participants of the workflow and *(ii)* the permissions needed to execute each task. For this purpose, the resources are classified within the organization structure. We must remark that the resources that will perform a task can be both human or software resources (such as Web services).

- *Management of the knowledge framework*. Tasks, problem-solving methods and domain models are the components of the knowledge dimension of our workflow framework. We have extended an ontological framework based on the Unified Problem-Solving Method description Language (UPML) [FMB+03] in order to store these components. Through UPML, these components are semantically described through ontologies (we used the OWL Language [DCvH+04] for such representation) and stored as a library of knowledge components.

2.2 Services for the Execution of Workflow Tasks

These services provide the functionalities included in the *client application interface* of the workflow reference model of the WfMC. Through these services, clients can obtain, perform and submit their work. For example, if a client is registered as a workflow participant (resource), these services will help it to check the work that the system will assign to it during the execution of the workflow. In the proposed architecture, all the tasks that a resource can perform will be executed through the invocation of a Web service, independently if the resource is human or software.

2.3 Services for the Administration and Monitoring of Workflows

These Web services provide the functionalities included in the *system monitoring and administration interface* of the workflow reference model of the WfMC, and allow the clients to manage the workflow instances that are being executed. For example, we have created Web services for stopping the execution of a workflow instance or for reassigning the tasks between the resources. These services also provide the access to the monitoring information of workflows execution, e.g. execution mean time of a task, resources load, etc.

2.4 Workflow Management System

The WMS depicted in Figure 1 implements our workflow framework [VLB06b] and provides the functionalities that facilitate the execution of the previously described Web services. This system is composed of three elements:

- *Workflow engine*. The core of the system. It is in charge of the instantiation, control and monitoring of the workflows execution.
- *Knowledge components server*. The workflow engine is connected to this element which provides the mechanisms to reason and verify each one of the components that define a workflow. As we previously mentioned, these components are represented by means of ontologies conform with the UPML specification. For this reason, this element uses the KAON2 inference engine [MS06] which allows us to infer new knowledge in OWL ontologies.
- *Library of components*. The workflow engine is also connected to a RDF repository *(i)* that stores and manages the data of each one of the knowledge components that define a workflow, i.e. the data that allow us to define a task, a problem-solving method, a high-level Petri net or a domain model; and *(ii)* that store the data created during the execution of a workflow. In this case, both the monitoring information and the intermediate data (such as inputs and outputs) of the execution of the workflow. This RDF repository has been developed with the Sesame 2.0 framework [Ope07].

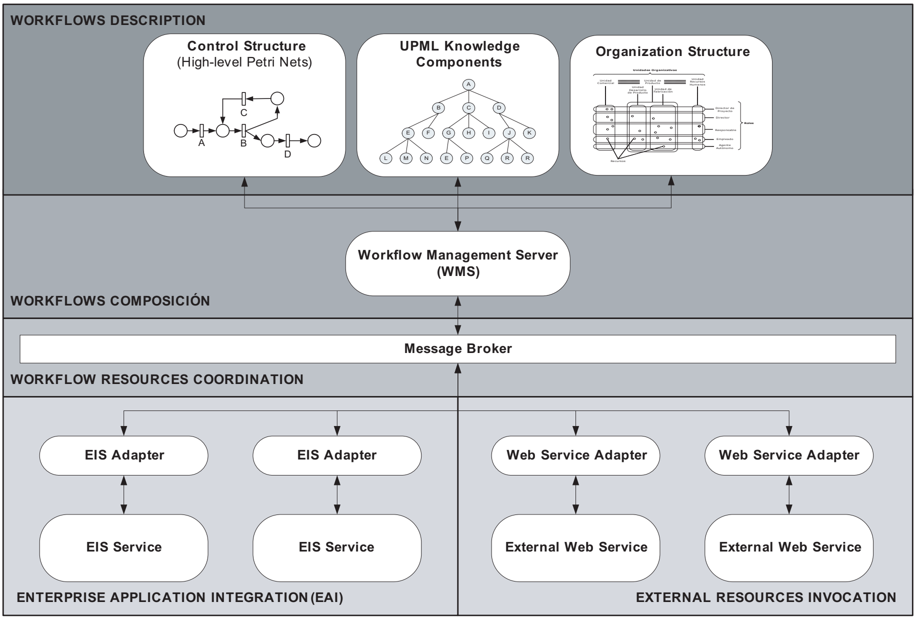

Figure 2: Infrastructure for the execution of knowledge-enriched workflows

3 Workflows Execution Infrastructure

The main aim of the previously described architecture is to provide the infrastructure for the execution of knowledge-enriched workflows by means of a service-oriented perspective. The execution infrastructure defined for our workflow framework and supported by the architecture is depicted in Figure 2. It defines a four layer model which captures the components that are involved in the execution.

The first layer of the infrastructure provides the access to the *definition of the workflows* that must be executed. This layer composes the description of the workflow from its knowledge components. It should be remarked that the knowledge components used to define a workflow, such as the control structures, organization structures, problem-solving methods, tasks, and domain models, are all defined independently. Following the philosophy of UPML, our conceptual framework specifies the workflows through a set of knowledge components that are related by means of adapters. These adapters define the way these knowledge components can be related and the conditions under which they can be combined. This feature makes the reuse of workflows easier since it only implies the redefinition of the adapters.

The second layer of the proposed infrastructure is responsible for handling the execution and composition of the workflows. This layer uses the first layer in order to obtain the workflow description and thus each one of the tasks that must be executed. This layer

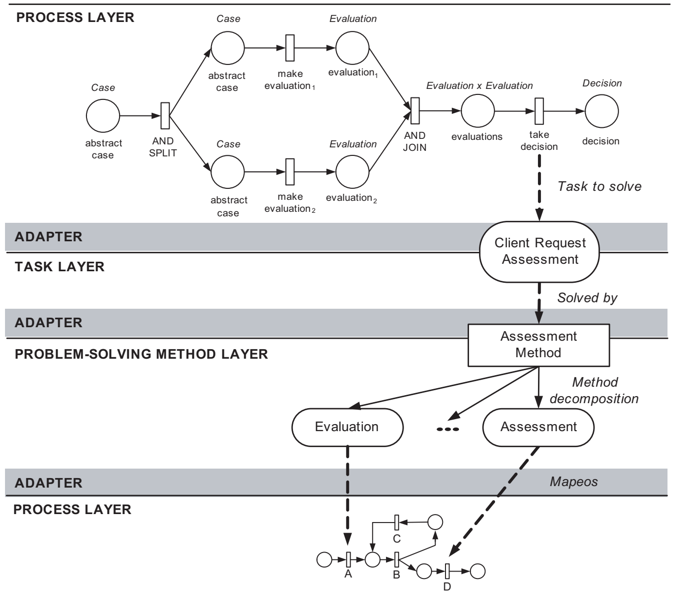

Figure 3: Example of execution of a knowledge-enriched workflows

performs the following operations:

- *Execution*. The execution of a task implies that the most suitable resource must be selected in order to drive its execution. The information to select it is obtained from the adapters that define the relations between the knowledge components. Specifically, these adapters relate the tasks that must be performed with the permissions needed to perform them and with the knowledge required by these tasks. From this information, the scheduler of the workflow engine assigns the work to the most suitable resources. When the work is assigned to a software resource (in our case a Web service) then the workflow engine *(i)* searches the service description in the RDF repository (which acts as a service registry) and *(ii)* call this service through the third layer of the infrastructure.

  Figure 3 depicts the knowledge components that are involved in a typical execution of a workflow. The first layer of this figure represents the control structure of the workflow described by means of a Petri net. Each one of the transitions (rectangles) depicted in this net represents a task to perform. These tasks are described by a knowledge component in the UPML framework and stored in the RDF repository. A problem-solving method must be selected in order to solve a task (third layer). If the problem-solving method that solves the task is non-composite then a Web service will perform this method.

- *Composition*. When the method that solves a task is composite then it will be composed by a set of tasks controlled by another Petri net structure. The composition between the different Petri nets implies the definition of a hierarchical net [GB05]. The transition that represents the task solved by the composite problem-solving method is substituted by the Petri net that defines its control structure.

Figure 3 depicts an example of such composition. In this case, the transition *take decision* is associated with the task that performs the assessment of the client requests. In this scenario, the task is solved by means of a method composed of two tasks (*Evaluation* and *Assessment*). These two tasks are mapped with two transitions of the Petri net that defines the control structure of the method.

The third layer of the infrastructure facilitates the coordination of the resources that participate in the workflow. This layer is defined by a *message broker* which establishes the logic of the message exchange. Through this solution the heterogeneity of the resources that participate in the workflows is hidden by the message broker. This architecture also uses the publication/subscription interaction model.

Finally, the fourth layer depicted in Figure 2 integrates the systems that will execute the workflow tasks, such as the Enterprise Information Systems (EIS) or Web services published by external providers. This integration is performed through adapters which will participate in the execution of the workflows like any other resource. This layer enables the access to all the systems with the same programming model and data formats.

4 Conclusions and Future Work

In this work we have presented a SOA that supports a conceptual framework for the definition of knowledge-enriched workflows. This architecture provides as Web services the interfaces defined in the workflow reference model of the WfMC and the infrastructure needed to define, compose, execute and monitor this kind of workflows.

With the proposed solution it is possible to specify workflows that include the knowledge about the way they are solved and about the domain in which they are defined. Furthermore, the adapters between the knowledge components will facilitate the reuse of workflows, e.g. the same control structure (Petri net) can be used in several domains changing only the adapters. Anyway, the use of Web services and the proposed infrastructure simplifies the integration of the WMS with other systems of organizations.

At present, as it depicted in Figure 4, we are developing two graphical user interfaces that will facilitate the specification of knowledge-enriched workflows. The first interface will facilitate the definition of high-level Petri nets annotated by means of ontologies. These nets will be used to define the control structure of the workflows. The second graphical user interface will facilitate the definition of the knowledge components of the workflows, i.e. the tasks, problem-solving methods and domain models. This interface will facilitate the access to the library of components defined in the RDF repository. We must mention that both interfaces will access the WMS through the Web services specified in section 2.1

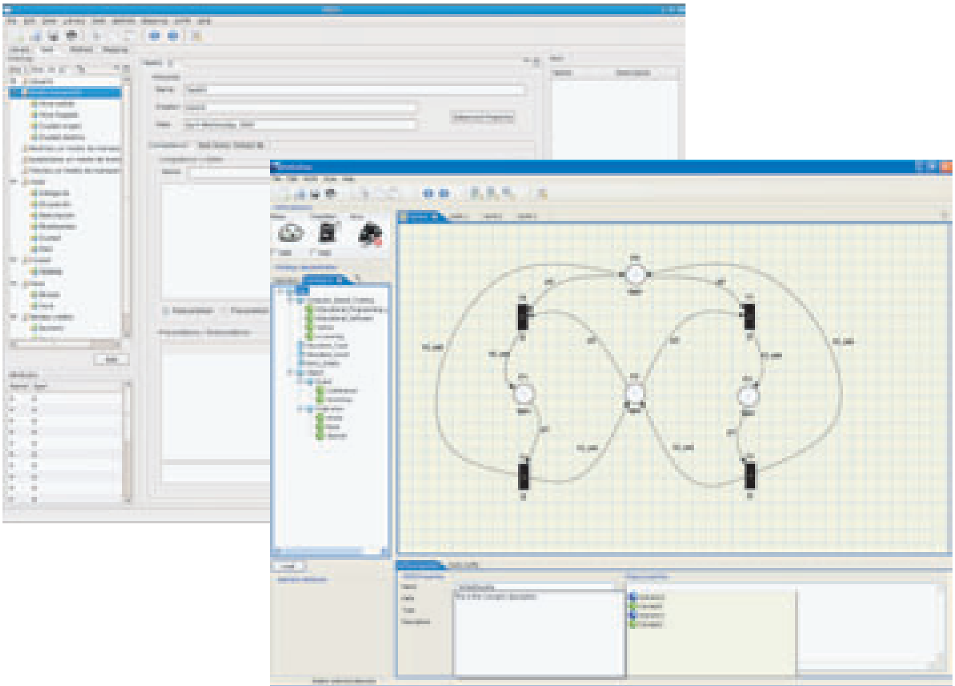

Figure 4: Knowledge components and high-level Petri nets editors

Acknowledgment

Authors wish to thank the Xunta de Galicia and Martínez Otero Contract, S.A. for their financial support under the projects PGIDIT06SIN20601PR and PGIDIT04DPI096E.

References

[BF98] V. Richard Benjamins and Dieter Fensel. Editorial: Problem-Solving Methods. *International Journal of Human-Computer Studies*, 49(4):305–313, 1998.

[CGK+02] F. Curbera, Y. Goland, J. Klein, F. Leymann, D. Roller, S. Thatte, and S. Weerawarana. *Business Process Execution Language for Web Services, Version 1.0*, November 2002. Standards proposal by BEA Systems, International Business Machines Corporation, and Microsoft Corporation.

[CNW01] Francisco Curbera, William A. Nagy, and Sanjiva Weerawana. Web Service: Why and How. In *Proceedings of the OOPSLA-2001 Workshop on Object-Oriented Services*, Tampa, Florida, USA, 2001.

[DCvH+04] Mike Dean, Dan Connolly, Frank van Harmelen, James Hendler, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider, and Lynn Andrea Stein. *OWL Web Ontology Language Reference*, October 2004. W3C recommendation. Available at http://www.w3.org/TR/owl-ref.

[Erl05] Thomas Erl. *Service-Oriented Architecture: Concepts, Technology, and Design*. Prentice Hall, 2005.

[FMB+03] Dieter Fensel, Enrico Motta, V. Richard Benjamins, Monica Crubezy, Stefan Decker, Mauro Gaspari, Rix Groenboom, William Grosso, Mark Musen, Enric Plaza, Guus Schreiber, Rudi Studer, and Bob Wielinga. The Unified Problem-solving Method Development Language UPML. *Knowledge and Information Systems*, 5(1):83–131, 2003.

[GB05] Luis Gomes and Joao Paulo Barro. Structuring and Composability Issues in Petri Nets Modeling. *IEEE Transactions on Industrial Informatics*, 1(2):112–123, 2005.

[GHS95] Dimitrios Georgakopoulos, Mark F. Hornick, and Amit P. Sheth. An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure. *Distributed and Parallel Databases*, 3(2):119–153, 1995.

[ISO02] ISO/IEC 15909-1. High-Level Petri Nets - Concepts, Definitions and Graphical Notation, 2002.

[Jen03] Kurt Jensen. *Coloured Petri Nets : Basic Concepts, Analysis Methods and Practical Use. Volume 1 (Monographs in Theoretical Computer Science. An EATCS Series).* Springer, April 2003.

[MS06] Boris Motik and Ulrike Sattler. A Comparison of Reasoning Techniques for Querying Large Description Logic ABoxes. In *Proceedings of the 13th International Conference on Logic for Programming Artificial Intelligence and Reasoning*, volume 4246 of *LNCS*, pages 227–241, Phnom Penh, Cambodia, 2006. Springer.

[Ope07] OpenRDF.org. *Sesame 2.0*, March 2007. Available at http://www.openrdf.org.

[vdA98] W. M. P. van der Aalst. The Application of Petri Nets to Workflow Management. *The Journal of Circuits, Systems and Computers*, 8(1):21–66, 1998.

[vdAvH04] W. M. P. van der Aalst and K. van Hee. *Workflow Management : Models, Methods, and Systems (Cooperative Information Systems).* The MIT Press, March 2004.

[VLB06a] Juan C. Vidal, Manuel Lama, and Alberto Bugarín. A High-level Petri Net Ontology Compatible with PNML. *Petri Net Newsletter*, 71:11–23, 2006.

[VLB06b] Juan C. Vidal, Manuel Lama, and Alberto Bugarín. A Workflow Modeling Framework Enhanced with Problem-Solving Knowledge. In *Proceedings of the 10th International Conference on Knowledge-Based Intelligent Information and Engineering Systems*, volume 4253 of *LNAI*, pages 623–632. Springer, 2006.

[WK03] Michael Weber and Ekkard Kindler. *Petri Net Technology for Communication-Based Systems*, volume 2472 of *Lecture Notes in Computer Science*, chapter The Petri Net Markup Language, pages 124–144. Springer-Verlag, 2003.

[Wor99] Workflow Management Coalition. The Workflow Reference Model, 1999. Available at http://www.wfmc.org/standards/referencemodel.htm.