A COMPARATIVE STUDY BETWEEN WEB SERVICE AND GRID SERVICE DEVELOPMENTS IN A MDA FRAMEWORK

Marcos López Sanz, Valeria de Castro, Esperanza Marcos
Kybele Research Group
Department of Information Systems and Languages
Rey Juan Carlos University
C/ Tulipan S/N – 28933 Móstoles
Madrid, Spain

José Luís Bosque
Department of Computer Architecture
Rey Juan Carlos University
C/ Tulipan S/N – 28933 Móstoles
Madrid, Spain

Keywords: Grid Services, Web Services, MDA, Platform Specific Models.

Abstract: The application of the MDA approach to the development of service-oriented systems facilitates the system migration over different platforms. The specific features of each platform are reflected in the PSM level of MDA. In this paper a comparison between the development of systems based on Web services and the development of systems based on Grid services is presented. The comparative study is carried out through a case study implemented on both a standard Web Service platform and a Grid platform based on the Globus Toolkit 4 middleware. After the study we conclude that a subdivision of the MDA's PSM level in two layers is needed: an upper layer with the characteristics shared by any service-based platform (a WSDL model and a model of the service code) and a lower layer with all the elements required to deploy the services successfully over the concrete execution platform.

1 INTRODUCTION

Current evolution of the information systems aims at the construction of systems in which the design, management, integration and evolution costs are minimized. This trend can be seen today in several research scopes of software development, from the definition of new design techniques to the emergence of new implementation approaches.

In the first case, approaches such as MDA (*Model Driven Architecture*) have come up. Following the idea of separating the specification of the system functionalities from the implementation details, MDA promotes the portability, interoperability and re-use degree of the systems (OMG, 2001). To achieve this, MDA specifies three levels of models to face the development of a system: CIM (*Computation Independent Models*), PIM (*Platform Independent Models*) and PSM (*Platform Specific Models*).

In the second case, the different technological approaches for system development, one of the most promising proposals at the moment is the service oriented paradigm (SOA – Service Oriented Architecture). This architectural approach is based on the interconnection of systems with light protocols such as SOAP (*Simple Object Access Protocol*) (Gudgin M. et al, 2003) and WSDL (*Web Service Description Language*) (W3C, 2003). With these protocols it is possible to communicate, through the Web and efficiently, software entities called "services". The main framework where this architecture is used is in the Web Services platforms because this is the environment which the previous protocols were developed for, although it is not the only one. Other field where this paradigm has been recently adopted is the *Grid*

López Sanz M., de Castro V., Marcos E. and Luís Bosque J. (2007).
A COMPARATIVE STUDY BETWEEN WEB SERVICE AND GRID SERVICE DEVELOPMENTS IN A MDA FRAMEWORK.
In *Proceedings of the Ninth International Conference on Enterprise Information Systems - ISAS*, pages 114-121
DOI: 10.5220/0002382301140121

Computing field (Foster I. and Kesselmann C., 1998) with the definition of OGSA –*Open Grid Service Architecture* (Foster I. et al., 2002) –. The OGSA architecture bases the creation of Grid platforms on Web Services, they are called generically as "Grid Services".

The combination of both approaches, MDA and SOA, results in the coming out of new methods for the development of systems which are versatile, service-oriented, and which development is based on the specification of models defining the system structure from several points of view and different abstractions levels.

The main objective of this paper is to compare the development of service-oriented systems in a MDA framework. In particular, we make a comparison between the development with Web Services and with Grid Services. To carry out this comparison we use, as case study, the development of a system for the management of medical images. Firstly, we use a method for the MDA development of Web Information Systems (WIS) based on services, and secondly, we implement that system over a Grid platform. After implementing the systems on both platforms, a standard-based Web Service platform (following the W3C recommendations and not taking into account other Web Service models such as REST (Fielding et al., 2000)) and a Grid platform, we make a study of the differences and similarities of both approaches. The interpretation of the comparison leads us to outline a proposal for the development of service-oriented systems based on MDA.

For a better comprehension we first explain the lessons learned, as a result of the study, and after that, we depict the concrete aspects of the study showing the aspects in common and main differences in the development with Web Services and Grid Services.

The rest of the article is organized as follows: in Section 2 we present an overview of the proposal for the subdivision of the MDA's PSM level for the development of service-oriented systems. In Section 3 the case study is described as well as the comparison of the development, at PSM level, with Web Services and Grid Services. Section 4 presents some related works describing the main benefits and contributions of our proposal and, finally, Section 5 concludes the paper and outlines some future works.

2 LESSONS LEARNED: A PROPOSAL FOR THE DEVELOPMENT OF SERVICE-ORIENTED SYSTEMS IN A MDA FRAMEWORK

The study we have carried out consists on the implementation of a service over both a standard Web Service platform and a Grid platform. After that work, we extracted and compared the main characteristics of each approach.

In this study we have detected several similarities in the development with both types of services (Web and Grid). Specifically, we have verified that both the WSDL description of the service and the code of the service itself are needed in both cases. These features are very similar for any type of service independently of the underlying technology used to implement the service.

However, there are some aspects that depend completely on the technology of the supporting infrastructure. This is the reason why we propose, for the MDA development of service-oriented applications, a separation of the PSM level in two sublayers that we name, alike other works (Muller P.A. et al., 2005): *Platform Dependent Layer* and *Technology Dependent Layer*. The specific characteristics that differentiate the development with Web Services and Grid Services and that have led us to suggest this subdivision are explained in Section 3.

The division of the PSM layer for both technologies can be seen in Figure 1. The main elements to be defined in each layer are:

- *Platform Dependent Layer (PD-PSM)*, at the upper level of the PSM layer: in this sublevel the models to be defined include, on the one hand, the WSDL description model (using guidelines such as the ones defined in (Marcos E. et al., 2003)), and, on the other hand, a model of the process that carries out the service. Both models should be obtained from the latest model defined in the PIM level.
- *Technology Dependent Layer (TD-PSM)*, at the lower level of the PSM layer: in this sublevel the elements to be defined include those related to the specific technology of the execution platform. In case of using a standard Web Services platform for example, the elements of this level would be the code of the service itself and the code of the WSDL description. In contrast, when using a Grid platform such as Globus Toolkit 4 (Foster I.,

2005) (GT4 from now onwards) there are some extra elements that need to be defined: a WSDD (*Web Service Deployment Descriptor*) file and JNDI (*Java Naming and Directory Interface*) file. Those elements are required because the development of Grid services over GT4 includes an explicit process of deployment (see Section 3.3 for more details). The preceding files need to be compiled together with the WSDL file and the service code itself prior to the deployment process.

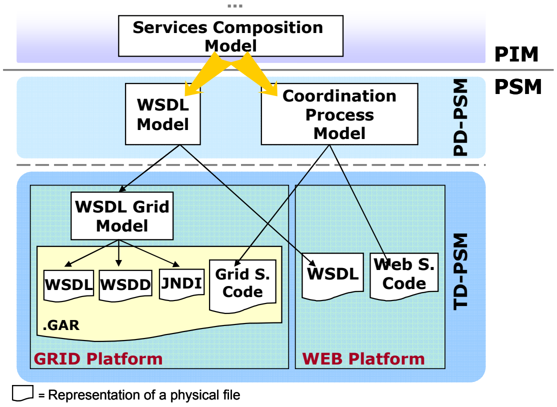

Figure 1: Proposal for the PSM level in the development of service-oriented systems.

3 COMPARATIVE AT PSM LEVEL: DEVELOPMENT WITH WEB SERVICES AND GRID SERVICES

As we have mentioned in the introduction, the comparative study starts from a MDA method for the development of WIS framed in the MIDAS methodology (Caceres P. et al, 2003). This method specifies the PIM models for a service-oriented system (De Castro V. et al, 2006) and the PSM models for the particular case of a system based on Web Services. Next, we depict the main features of the case study in which we centre the comparison, describing also the models created at PIM level for this case study. After that we explain the development at PSM level with both standard Web Services and Grid Services. Finally, we summarize the results of the comparative study by extracting the main similarities and differences.

3.1 Case Study

The case study we use as basis for the comparative study comes up from the necessity of extending the features offered by GESiMED, a WIS for the management of medical images (Acuña C. et al., 2004).

In particular, what we wanted to add to this system is the capability of performing, in a unique process, the acquisition of images from a database and its subsequent processing in order to be displayed on a Web page. At this moment, GESiMED offers both functionalities but in separated processes. So, the main goal of the new service ("to obtain images" from now onwards) will be to obtain the images and to process them for a correct visualization afterwards. The interest of using a Grid Platform for the case study is based on the opportunity of getting computational benefits when executing some parts of the developed service over a Grid environment. The processing of the images, for example, is a task whose algorithm can be easily parallelized or distributed; this also occurs with the extraction of the images from the original files.

In (De Castro V. et al, 2006), the PIM models for this case study are described in detail: user services model, extended use cases model, service process model and finally the service composition model. In this paper we reproduce only the last one, the services composition model (see figure 2) because it is the model from which all the models that will be defined at PSM level will be derived. This model is represented through an activity diagram. On it, the sequence of tasks that needs to be performed in order to achieve the objective of the service is identified. Apart from the tasks performed by the system itself, this model also reflects the collaborators of the system, which are external entities that perform specific tasks unable to be done by the system itself and required to complete the service.

The corresponding tasks of the "to obtain images" service can be seen in figure 2 and include the identification and validation of the user in the system, the payment of fees to perform the task selected by the user, the selection of the query criteria, the execution of the query in the database, the extraction of the images from the result files and, finally, the presentation of the results to the user offering the option of downloading the obtained result files.

In this point of the description we have to distinguish between two types of tasks: on the one hand, those ones carried out by the system itself (in Figure 2 those ones in the "Neuroscience researcher" swimlane) and, on the other hand, the tasks performed by the collaborators ("Medical Images Management

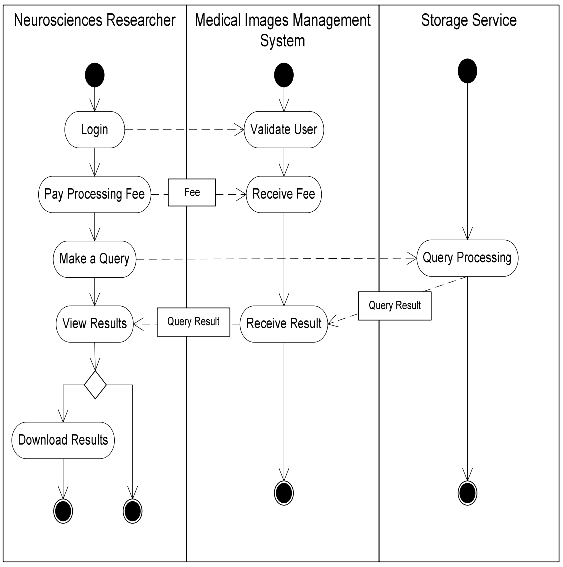

Figure 2: Services Composition Model.

System" and "Storage Service"). The part of this diagram on which we will focus at the PSM level will be the one related with the tasks of the system itself, that is, the composition of the invocations to the operations of the collaborators.

3.2 Web Service PSM

At the PSM level of the MDA approach we have to define all the models that reflect the specific features of the platform that will be used as execution support infrastructure for the developed service. When using a service-oriented approach, and especially an approach based on Web Services, a WSDL model describing the main operations offered by the service at the PSM level is needed and also a representation of the code of the service itself.

Because we start from a service which coordinates the invocations of several external services, the composition process can be represented, for example, using a language for the description of processes such as BPEL – *Business Process Execution Language* (Andrews T. et al., 2003).

Figure 3 shows a representation of the WSDL model corresponding to the service "to obtain images". This model has been built using the guidelines explained in (Marcos E. et al., 2003). From this model it will be possible to obtain a WSDL file that, together with the code of the service itself, represent the final elements of the development process of our case study with Web Services. The files with the WSDL description and the code of the service are the only ones needed to deploy and execute a service over a server which executes Web Services according to the WSDL and SOAP standards. In our case we have used the Apache AXIS framework as execution platform for the developed Web Services. The deployment over this kind of platform consist of just copying the Web Service files in the right server folder, consequently, we suppose that there are no needed specific elements for this purpose.

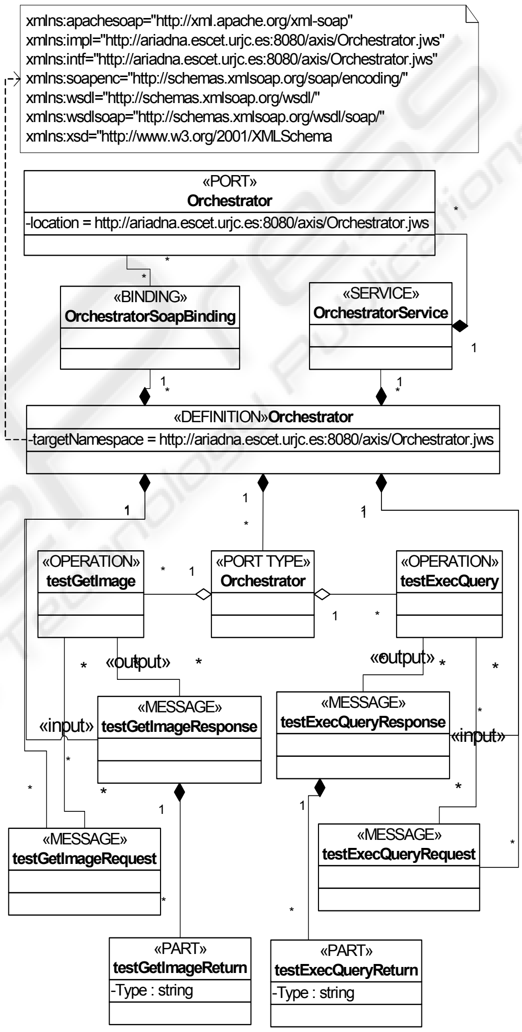

Figure 3: WSDL model of the Web Service.

3.3 Grid Service PSM

When facing the development of a system on a Grid platform there are several concepts that have to be understood previously, starting from the concept of "Grid Service" itself. As it is defined in OGSA, a Grid Service can be considered as a concrete application of a Web Service but in a Grid Computing environment. Similarly to Web Services, Grid Services are communicated through SOAP and are described with WSDL. The main difference is in the goal of the platforms on which both types of services are executed. On the one hand, the objective of the Web services is to put at the disposal of the users a concrete functionality available through the Web, on the other hand, in the Grid Services case, this availability is referred to the possibility of sharing software resources in a distributed virtual organization.

As stated previously, in both cases a WSDL description is needed; however, there are some restrictions to be taken into account when creating the WSDL description of the Grid Service:

- It is not necessary to specify neither *<binding>* nor *<service>* elements because in the Grid Service compilation process the compilation tool will add these elements automatically.
- Although in the WSDL description of the Web Service a definition of the data types used in the invocations and responses of the service operations is not always established, in the Grid Service case it is mandatory to define them by specifying a *<type>* element containing a fragment of XML Schema whose *targetNamespace* attribute be the WSDL file itself.
- To complete the definition it is necessary to add to the *<definition>* element those references to the namespaces used in the body of the WSDL file (that will be relative to the Grid platform)

The execution of Grid Services over the GT4 middleware involves a previous phase of compilation and deployment. In the compilation phase, apart from the WSDL description and the code of the service itself, two auxiliary files are needed:

- WSDD (*Web Service Deployment Descriptor*, 2005) file: contains a description of the deployment process and is used to communicate to the platform how the service should be published on the Grid. The WSDD code for our case study can be seen in figure 4.
- JNDI (*Java Naming and Directory Interface*, 2006) file: this file is related to the concept of resource associated to any Grid application. In the Grid Services case, the resource to be controlled consists on the maintenance of the service state between different invocations to the same service. Due to the properties of the case study, we don't need to maintain the state of our Grid Service in any case; but, and because of the characteristics of the platform, the existence of this file is mandatory. In our case, this file will contain exclusively a reference to the generic resource factory provided by GT4. The code of this file can be seen in figure 5.

```
<?xml version="1.0" encoding="UTF-
8"?>
<deployment
  name="defaultServerConfig"
  xmlns=http://xml.apache.org/axis
/wsdd/
  xmlns:java=http://xml.apache.org
/axis/wsdd/providers/java
  xmlns:xsd="http://www.w3.org/200
1/XMLSchema">

 <service
          name="prueba1/Ochestrator"
          provider="Handler"
          use="literal"
          style="document">
  <wsdlFile>share/schema/Orchestra
tor_instance/Orchestra-
tor_service.wsdl
   </wsdlFile>
  <parameter
      name="className"
  value="prueba1.impl.Orchestrator
"/>
  <parameter
         name="allowedMethods"
         value="*"/>
   <parameter
         name="handlerClass"
         value="org.globus.axis.
               provid-
ers.RPCProvider"/>
   <parameter
        name="scope"
        value="Application"/>
   <parameter
        name="providers"
         value="GetRPProvider"/>
   <parameter
         name="loadOnStartup"
         value="true"/>
 </service>
</deployment>
```

Figure 4: Content of the WSDD file.

```
<?xml version="1.0" encoding="UTF-
8"?>
<jndiConfig
  xmlns="http://wsrf.globus.org/jn
di/config">
 <service
name="prueba1/Orchestrator">
   <resource
        name="home"

  type="org.globus.wsrf.impl.
            ServiceResourceHome">
      <resourceParams>
    <parameter>
       <name>factory</name>
           <value>
org.globus.wsrf.jndi.BeanFactory
        </value>
    </parameter>
   </resourceParams>
  </resource>
 </service>
</jndiConfig>
```

Figure 5: Code of the JNDI file.

3.4 Summary of the Comparative Study

After completing the development of the case study with both Web services and Grid Services, we have detected that the main difference in the development with both types of services is found in the moment of deploying the service on a specific execution platform: while traditional Web Services doesn't need a explicit phase of deployment, Grid Services, on the contrary, demand a deployment process that involves other files apart from the ones defined because of its condition of Web Services. The need to create extra files and to adapt the WSDL description to the Grid platform leads to the necessity to define concrete models in the MDA's PSM level. Our proposal of models for this level is the one described in Section 2.

In relation to the code that orchestrates the composition in the "to obtain images" service, the differences between the Web Service code and the Grid Service code are negligible. The unique remarkable differences take place in the moment of referencing the libraries and data types used to perform the functionality of the service. These references will correspond to specific paths of the execution platform. The process of invocation of the collaborators is quite similar in both cases because they use standard protocols to call the external Web Services. Table 1 summarizes the main differences and similarities found in the comparative study.

4 RELATED WORKS

The application of the MDA approach to the design of Grid systems and applications is not so original. Recently, some proposals for the model driven development of Grid applications have come up.

Smith et al. (2006) suggest the division of the MDA's PSM level in two parts, separating the functionality of the application from the Grid logic concerns. The authors specify a UML profile for the description of Grid services at PSM level. Although their proposal separates the logic of the client who uses the service from the classes that form the Grid service itself, this separation is made in a unique model for the PSM level. In our case, we also propose a subdivision of the PSM level but with some slight differences. Since our proposal is framed in the MIDAS methodology, the interaction with the client of the system is modelled separately, specifically in the hypertext aspect of the development architecture proposed by MIDAS (see (Caceres P. et al., 2003) for more details). As a result, we only have to focus on the modelling of the system business logic. Moreover, we base our division of the PSM level on the identification of the service common features shared by the underlying implementation platforms.

In Manset et al. (2006), an architecture-centric approach for the automatic generation of Grid applications is presented. The authors define a model driven engineering process for the generation of applications based on a formal specification of the Grid architecture. Their proposal demands that the designers involved in the engineering process have a specific knowledge in architecture description languages. The main difference with our proposal is that the Manset et al. proposal focuses exclusively on the development of Grid applications over Grid platforms whereas ours aims to facilitate the migration of systems also over non-Grid platforms such as standard Web Service based platforms.

5 CONCLUSIONS AND FUTURE WORKS

In this paper we have presented a comparative study between the MDA development of systems based on Web services and the MDA development of systems

Table 1: Summary of the comparative study: Web Services vs. Grid Services.

	GRID SERVICES	WEB SERVICES
Goal of the execution platform	To create virtual organizations based on resource sharing	To have software resources available through a network
Service description	Modified WSDL	Standard WSDL
– Description of types used in operations	Mandatory	Optional
– *binding* and *service* elements	Established in a later compilation process	Established in the moment of creating the WSDL description
Service coding	Language supported by the platform	Language supported by the platform
Compilation and deployment process	Mandatory	Not needed
Auxiliary files	JNDI / WSDD	NO

based on Grid Services as implementation technology. To carry out this comparison we have used, as case study, the development of a service that allows to obtain a set of images from a database and that process them in order to be displayed on a Web page.

In the comparative study we have detected that, in the developments of systems with Web Services and Grid Services, there are some elements in common as well as other that depend completely on the underlying technology of the execution platform. Because of that, in this paper we have outlined a proposal for the division of the MDA's PSM level in two sublayers: an upper layer, PD-PSM, where all the shared aspects of the service-oriented development should be modelled; and a lower layer, TD-PSM, where the models with all the specific features and restrictions of the execution platform should be defined and which will be used to obtain the final code of the service.

At this moment there are guidelines for the definition of some PSM models, for example the ones described in (Marcos et al., 2003) to create the WSDL model. These definitions could be used also in the development process of Grid Services, both for the WSDL model of the PD-PSM layer and the WSDL Grid model of the TD-PSM layer. In the last case a definition of an extension to the UML-WSDL metamodel should be done in order to include the Grid characteristics.

There is a lot of work to be completed from the proposal depicted in this paper: first, all the models included in every PSM sublayer should be better defined; second, the guidelines for the transformation of the last model defined at PIM level to the PD-PSM models as well as the models transformations between the PD-PSM and TD-PSM level should also be defined. Finally, the automatic generation rules of executable code form the lowest models of the architecture should also be defined.

ACKNOWLEDGEMENTS

This research is partially granted by projects: GOLD (TIN2005-00010) and "Virtual trainers with low cost platforms" (TIC2003-08933-C02-00), financed by the Ministry of Science and Technology of Spain and "GATARVISA: Algorithms, Techniques and Applications of Virtual Reality and Advance Simulation" (S-0505/DPI/0235), financed by the Science and Technology Council of the government of Madrid (Spain).

REFERENCES

Acuña C., Marcos E., de Castro V. & Hernández J.A. 2004. A Web Information System for Medical Image Management. 5th International Symposium on Biological and Medical Data Analysis. Barcelona, Spain. Springer-Verlag. LNCS 3337. pp. 49-59.

Andrews, T., Curbera, F., Dholakia, H. et al. 2003. Business Process Execution Language for Web Services, Version 1.1. Specification, BEA Systems, IBM Corp., Microsoft Corp., SAP AG, Siebel Systems.

Cáceres, P., Marcos, E. & Vela, B. 2003. A MDA-Based Approach for Web Information System Development. Workshop in Software Model Engineering (WiSME@UML'2003) in conjunction with UML Conference. Octubre, 2003. San Francisco, USA.

De Castro V., Marcos E. & López M. 2006. A Model Driven Method for Service Composition Modeling: A Case Study. Int. Journal of Web Engineering and Technology, Vol. 2, No. 4, pp. 335-353.

Foster I. & Kesselman C. The Grid: Blueprint for a New Computing Infrastructure. Ed. Morgan Kauffmann, 1998. ISBN 1558604758.

Foster I. 2005. Globus Toolkit Version 4: Software for Service-Oriented Systems. IFIP International Conference on Network and Parallel Computing, Springer-Verlag LNCS 3779, pp 2-13.

Fielding R. T. and Taylor R. N. 2000. Principled design of the modern Web architecture. In Proceedings of the 2000 International Conference on Software Engineering (ICSE 2000), Limerick, Ireland, June 2000, pp. 407–416

Gudgin M., et al. 2003. Simple Object Access Protocol. W3C recommendation, W3C. Retrieved June, 2006, from http://www.w3.org/TR/soap12-part1

Java Naming and Directory Interface (JNDI). 2006. Retrieved November 2, 2006, from http://java.sun.com/products/jndi/.

Marcos, E., De Castro, V. & Vela, B. 2003. Representing Web Services with UML: A Case Study. The First International Conference on Service Oriented Computing (ICSOC03). M.E. Orlowska, S. Weerawarana, M.P. Papazoglou, J. Yang (Eds.). LNCS 2910, Springer Verlag, pp.15-27.

Manset D., Verjus H., McClatchey R. & Oquendo F. 2006. A Formal Architecture-Centric Model-Driven Approach For The Automatic Generation Of Grid Applications, Proceedings of the 8th International Conference on Enterprise Information Systems, Paphos, Cyprus, pp. 23 - 27.

Muller, P.-A, Studer, P., Fondement, F. & Bézivin, J. 2005. Platform Independent Web Application Modeling and Development with Netsilon. Journal on Software and Systems Modeling (SoSyM), Vol. 4 Nr. 4, Best papers of the <<UML>>'03 conference, Springer, pages 424 - 442

OMG, 2001. OMG Model Driven Architecture. Miller, J., Mukerji, J. (eds.). Document Nro. ormsc/2001-07-01. Available at: http://www.omg.com/mda.

Foster I., Kishimoto H, Savva A. et al. 2005. Open Grid Services Architecture version 1.0 – OGSA. Retrieved January, 2005, from http://forge.gridforum.org/ projects/ogsa-wg/document/draft-ggf-ogsa-spec/en/ 23.

Smith, M., Friese, T., and Freisleben, B. 2006. Model Driven Development of Service-Oriented Grid Applications. In Proceedings of the Advanced int'L Conference on Telecommunications and int'L Conference on Internet and Web Applications and Services (February 19 - 25, 2006). AICT-ICIW. IEEE Computer Society, Washington, DC, 139. DOI= http://dx.doi.org/BBA0CE2A-94C5-4D2C-BA23-ABE1A82CB45D

W3C Web Services Description Language (WSDL). 2003. Version 1.2. W3C Working Draft 3. Retrieved from: http://www.w3.org/TR/wsdl12/

Web Service Deployment Descriptor (WSDD). (2005). Reference Guide. Retrieved August 17, 2006, from http://ws.apache.org/axis/java/reference.html#DeploymentWSDDReference