WORKSHOP

ERO AMERICANO

E ENGENHARIA DE REQUISITOS

E AMBIENTES DE SOFTWARE

M. Lencastre, J. F. Cunha, A. Vallecillo (Eds.)

IDEAS 2008

Proceedings of the

11th Iberoamerican Workshop on
Requirements Engineering and Software
Environments

Recife, Pernambuco, Brazil
February 11-15, 2008

Editors

Maria Lencastre
Departamento de Sistemas Computacionais
Universidade de Pernambuco
Recife, PE, Brasil
maria@dsc.upe.br

João Falcão e Cunha
Departamento de Engenharia Industrial e Gestão
Faculdade de Engenharia. Universidade do Porto
Rua Dr. Roberto Frias, s/n.
4200-465, Porto, Portugal.
jfcunha@fe.up.pt

Antonio Vallecillo
Departamento de Lenguajes y Ciencias de la Computación
Universidad de Málaga
Bulevar Luois Pasteur, 35.
29071 Málaga, Spain
av@lcc.uma.es

FICHA CATALOGRÁFICA

I12p	Iberoamerican Workshop on Requirements Engineering and Software Environments (11.: 2008 : Recife, PE) Proceedings of the 11 th Iberoamerican workshop on requirements engineering and software environments : IDEAS 2008, Recife, February 11-15, 2008. – Recife : FASA, 2008. xv, 380 p.
	1. Engenharia de software – Congressos. 2. Fórum (Debates). I. Título.
ISBN	9788570841346 CDU 004.41

Organized by Fernanda Alencar

Prefacio

El escribir un prefacio significa que se ha llegado al final de un largo camino, plagado tanto de algunos obstáculos como de gratificantes recompensas. Es para nosotros por tanto un placer darles la bienvenida a IDEAS 2008 mediante estas palabras.

El presente volumen contiene las actas con los artículos que presentado en el undécimo Workshop Americano sobre Ingeniería de Requisitos y Ambientes Software (IDEAS 2008), que se celebra este año en Recife, Pernambuco, Brasil, del 11 al 15 de Febrero de 2008.

Pernambuco ha mantenido en toda su historia una fuerte identidad, que ha contribuido decisivamente en la cultura y la política brasileña. Pernambuco fue un estado tradicionalmente centrado en la explotación de la caña de azucar. Sin embargo, en los últimos tiempos la capital de Pernambuco, Recife, se está consolidando como uno de los grandes centros tecnológicos de Brasil.

IDEAS 2008 es la undécima Conferencia de la serie IDEAS que, desde finales de los años 90 proporciona un foro para la presentación y el intercambio de resultados de la investigación y experiencias industriales en los campos de la Ingeniería de Requisitos y Ambientes de Software. En el año 2008 esta conferencia la organiza el Departamento de Sistemas Computacionales (DSC) de la Universidad Estatal de Pernambuco, junto con el Laboratorio de Ingeniería de Requisitos (LER) de la Universidad Federal de Pernambuco, en Recife, Brasil. Esta es la segunda vez que Brasil acoge a la conferencia IDEAS, tras la celebración de la primera edición en Torres, Rio Grande do Sul, en 1998.

La conferencia IDEAS trata de favorecer y promover el intercambio de conocimiento y experiencias entre profesores, estudiantes y profesionales del ámbito académico y empresarial iberoamericano, estrechando las relaciones entre los diferentes grupos de estos países que trabajan en los temas de interés de la conferencia.

Este año la conferencia recibió 74 artículos para su revisión, entre los cuales el Comité de Programa decidió seleccionar 22 para su presentación en la conferencia. Esto ha supuesto un ratio de aceptación del 29%, lo que demuestra el arduo proceso de revisión y selección al que fueron sometidos los artículos, así como la calidad de los finalmente seleccionados. Además de estos artículos, otros 12 fueron seleccionados para participar en la conferencia como artículos cortos, con la idea de favorecer y estimular el debate científico entre los asistentes y dar cabida a la presentación de trabajos incipientes. Todos los artículos fueron revisados siguiendo un sistema de peer-review por al menos 2 revisores (en media 2.84) de entre los miembros del Comité de Programa de IDEAS 2008, que estuvo compuesto por expertos internacionales de reconocido prestigio.

El programa resultante refleja perfectamente el hecho de que tanto la Ingeniería de Requisitos como los Ambientes Software involucran diferentes aspectos, tanto técnicos como de índole humana y de organización, en cuanto a recursos y a procesos. Estos aspectos incluyen los procesos de desarrollo software, los requisitos de seguridad, el uso de las ontologías en la ingeniería del software, la calidad del software, el modelado conceptual, la gestión de los requisitos, y los casos de uso y experiencias en ingeniería de software. Estos temas constituyen precisamente las sesiones del programa de la conferencia.

Por otro lado, el éxito de la conferencia IDEAS también se refleja en el número de eventos que suceden a su alrededor. IDEAS 2008 cuenta con cuatro tutoriales, dos mesas redondas, y el tercer Workshop Internacional sobre i* (istar'08). Además, este año hemos contado con tres conferenciantes invitados de primer nivel: el profesor John Mylopoulos (de las universidades de Toronto, Canada, y Trento, Italia) que impartió la charla "Goal-Oriented Requirements Engineering"; el profesor Oscar Pastor López (de la Universidad Politécnica de Valencia,España) que impartió la charla "Web Engineering: Present, Past and Future"; y el profesor José Carlos Maldonado (de la Universidad de São Paulo, Brasil) que impartió la charla "Software testing in the Context of Qualipso Project and National Perspectives". Nuestro agradecimiento más sincero por su disponibilidad para aceptar la invitación y venir a Recife a impartir sus conferencias.

También queremos expresar nuestro agradecimiento a los miembros del Comité de Programa por su tiempo y dedicación a la hora de revisar los artículos y seleccionar los artículos aceptados para su presentación, que han permitido confeccionar un año más un programa de altísima calidad y nivel. También queremos agradecerles a los organizadores locales del Departamento de Sistemas Computacionales (DSC) de la Universidad Técnica de Pernambuco todo su esfuerzo y trabajo, que han permitido hacer realidad esta conferencia. Mención especial requiere a Profa. Fernanda Alencar, que fue la encargada de confeccionar estas actas y a Prof. Jaelson Castro por su apoyo constante y ayuda.

Finalmente, nos gustaría mencionar nuestro agradecimiento explícito a los patrocinadores del evento: El Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), la Fundação de Amparo à Ciência e Tecnologia do Estado de Pernambuco (FACEPE), la Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), la Pro-reitoria para Assuntos de Pesquisa e Pós-Graduação (Propesq-UFPE), y el Departamento de Sistemas Computacionais (DSC/POLI/UPE) que hicieron posible que la conferencia fuera todo un éxito. También mencionar el sistema de revisión de artículos que utilizamos, EasyChair, que fue de una utilidad y ayuda inestimable. Nos gustaría por tanto agradecer a su creador, Andrei Voronkov, por toda su ayuda y eficiente soporte durante el proceso de revisión y la preparación de las actas.

Muchas gracias a todos los asistentes y participantes a IDEAS 2008, y esperamos verles de nuevo en Colombia en el próximo IDEAS 2009.

Diciembre 2007

Maria Lencastre
João Falcão e Cunha
Antonio Vallecillo

Conference Organization

General Chair

Maria Lencastre

Programme Chairs

João Falcão e Cunha
Antonio Vallecillo

Programme Committee

Fernanda Alencar
João Paulo Almeida
Carina Alves
João Araújo
Alvaro Arenas
Marcio Barros
Nelly Bencomo
Pere Botella
Regina Braga
Antonio Brogi
Coral Calero
Rafael Calvo
Carlos Canal
Jaelson Castro
Alejandra Cechich
Luca Cernuzzi
Marcio Delamaro
Isabel Diaz
Amador Duran
Sandra Fabbri
Ricardo Falbo
Xavier Franch
Marcelo Frias
Alessandro Garcia
Jesús García Molina
Itana Maria de Souza Gimenes
Silvia Gordillo
Juan Hernandez
Miguel Katrib
Nora Koch
Julio Cesar Leite
Maria Lencastre
José Carlos Maldonado
Esperanza Marcos
Emilia Mendes
Jonas Montilva
Ana Moreira
Nuno Nunes
Hanna Oktaba
Luis Olsina
Oscar Pastor López
Vicente Pelechano
Ernesto Pimentel
Francisco Pinheiro
Claudia Pons
Ruben Prieto-Díaz
Daniel Riesco
Gustavo Rossi
Francisco Ruíz
Victor Santander
Ernest Teniente
Miguel Toro Bonilla
Guilherme Travassos
Alexandre Vasconcelos
Marcello Visconti

Local Organization

Alex Sandro Gomes
Carina Alves
Cristine Gusmão
Fernanda Alencar

Genésio Neto
Jaelson Castro
Luis Soares
Márcia Lucena
Márcio Cornélio
Ricardo Massa
Ricardo Ramos
Tiago Massoni
Sérgio Soares

External Reviewers

Marcio Barros
Regina Braga
Guillermo Juan Covella
Maria Istela Cagnin
Rafael Calvo
Valter Vieira de Camargo
Pedro J. Clemente
Jose Maria Conejero
Javier Cubo
Javier Cámara
Leandro Da°on
Amador Duran
Isabel Diaz
Maria Jose Escalona
Sandra Fabbri
Sandra Ferrari
Andrés Flores
Fred Freitas
Thaizel Fuentes
Roxana Giandini
Itana Gimenes
itana Gimenes
Silvia Gordillo
Francisco Gutiérrez
Francisco Hernández-Quiroz
Maria de los Angeles Martin
Hernan Melgratti
Hernan Molina
German Montejano
Ana Moreira
Hanna Oktaba
J.L. Ortega-Arjona
Joaquin Peña
Antonia M. Reina Quintero
Fernando Rincón
Gustavo Rossi
Gwen Salaui
Marisol Sanchez-Alonso
Laura Semini
Flavio Signorelli Mendes
Simone do Rocio Senger de Souza
Rosana Teresinha Vaccare Braga
Jo Ueyama
Rafael Valencia
David Benavides
Valeria de Castro

Table of Contents

Short papers

An Ontology for the WSRP Standard

Mª Ángeles Moraga[1], Ignacio García-Rodríguez de Guzmán[1], Coral Calero[1], and Mario Piattini[1]

[1] Alarcos Research Group. UCLM-INDRA Research and Development Institute. University of Castilla-La Mancha. Paseo de la Universidad 4 –13071 –Ciudad Real –Spain {MariaAngeles.Moraga,Ignacio.GRodriguez,Coral.Calero,Mario.Piattini}@uclm.es

Abstract. The use of Web portals continues to rise, showing their importance in the current information society. Specifically, this work focuses on portlet-based portals. Recently, the WSRP (Web Services for Remote Portlets) standard has come into existence. Its aim is to provide a common interface in order to allow the communication between portal and portlets. Bearing all in mind, in this work we propose an ontology for this standard. The ontology leads both portlet and portal developers to focus their effort on developing the portlet domain logic instead of implementing its communication.

Keywords: portlets, web portals, WSRP standard, ontology

1 Introduction

Today, the emergence of portlets is leading to develop web portals as an aggregation of portlets. Portlets are used by portals as pluggable user interface components that is, they are displayed in the portal and provide content for it [4].

When portlets came into existence, their main problem was the lack of interoperability. Portlets had to be custom-developed for each portal server because the API was different for each server [5]. However, this problem was eliminated when the WSRP (Web Services for Remote Portlets) standard appeared.

In spite of the existence of the WSRP standard, there are different companies which offer portlets that do not adhere to this standard (see http://www.jahia.net/jahia/571). These portlets present several problems between them such as non-existence of interoperability between portlets and portal; and impossibility of offering these portlets to different portal developers.

Currently it is possible to find portlet repositories where portal developers can acquire a portlet which satisfies their's needs [6]. Indeed, the Open Source Portlet Repository Project has been recently launched [1] to foster the free and open exchange of portlets. Other similar portlet-sharing sites include Portlet Swap (jboss.org) and Portlet Exchange (portletexchange.com). The majority of portlets, which are offered in repositories, adhere to the WSRP standard. However, its specification is not easily understood.

On the other hand, ontologies allow us to share knowledge and facilitate the communication between people and/or systems [10]. Thereby, using an ontology both

portlet and portal developers can focus their effort on developing the portlet domain logic instead of understanding the standard. In addition, it makes easier the creation of new portlets as well as the adaptation of portlets that do not adhere to the WSRP. For all these reasons, we have created a specific ontology (WSRP-O) for WSRP.

This paper is structured as follows. Section 2, the ontology for the WSRP is shown whereas in section 3 conclusions and future work are drawn.

2 An ontology for the WSRP standard: WSRP-O

The main objective of this work is to develop an ontology for the WSRP standard. For this reason, first of all, general information about WSRP should be given.

The goal of the WSRP standard is to enable an application designer or administrator to pick from a rich choice of compliant remote content and application providers (portlets), and integrate them with just a few mouse clicks and no programming effort [8]. The standard defines different interfaces and operations for carrying out the specified functionality. It may be worth emphasizing that the WSRP standard is independent of the programming language.

However, the use of a standard is not a trivial task, the vocabulary is complex and generally is hard to understand on your own. In view of this situation, a tool which simplifies the standard, such as an ontology, would be necessary.

On the other hand, we should provide a definition of ontology: "Defines the basic terms and relations comprising the vocabulary of a topic area as well as the rules for combining terms and relations to define extensions to the vocabulary [7]".

Nonetheless, different definitions about ontologies can be found in the literature. The majority of them provide a set of common elements. According to [3], these elements are: *Classes, Relations, Formal axiom* and *Instances*.

According to [2] the ontology representation language should have rich and formal abstractions. UML-based ontologies have the obvious advantage of being more widely understandable [9]. For this reason, in order to describe the ontology proposed in this work, UML class diagrams are used. In addition, the proposed ontology (figure 1) gives support to develop new portlets as well as to adapt existents portlets to the standard. The UML diagrams are accompanied by some constraint defined using OCL.

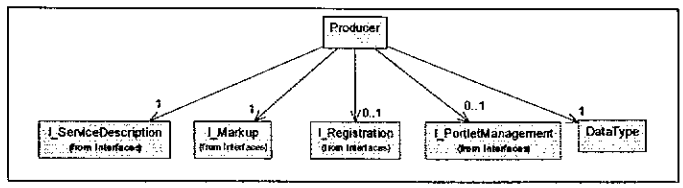

Fig. 1. A global overview of the ontology

As we can see, the ServiceDescription and Markup interfaces are compulsory whereas Registration and PortletManagement interfaces are optional. All these interfaces contain different operations. These operations have two common concepts:

fault and *result*. For this reason we have created a base class (named Operation) representing these common concepts (figure 2 (a)).

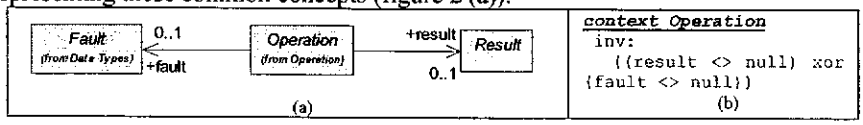

Fig. 2. (a)"basicPortletOperation" as a base class for all the operations of the ontology; (b) constraint for the abstract *Operation* class

The *Fault* class represents the possible error message that can arise as a result of the operation, whereas the response represents the result of the operation when any error has happened. The error messages as well as the data type of the response depend on the operation. For this reason, a constraint for each operation will be created to specify the possible faults and results of an operation. The constraint indicates the kind of allowed error messages along with the allowed data type of the response for each operation. Finally, it may be worth emphasizing that an operation has to return a fault or a response but it is not possible to return both of them. In order to represent this restriction, the constraint of the figure 2 (b) has been defined.

The ontology is broken down according to interfaces decomposition into subontologies. In each subontology the different operations, which are allowed, are represented. Thus, in order to represent the allowed data types another subontology for data types has been included. Next, the different subontologies are presented.

Firstly, we are going to present the ServiceDescription subontology, this subontology represents the ServiceDescription interface. This interface defines an operation for acquiring the Producer's metadata. (figure 3)

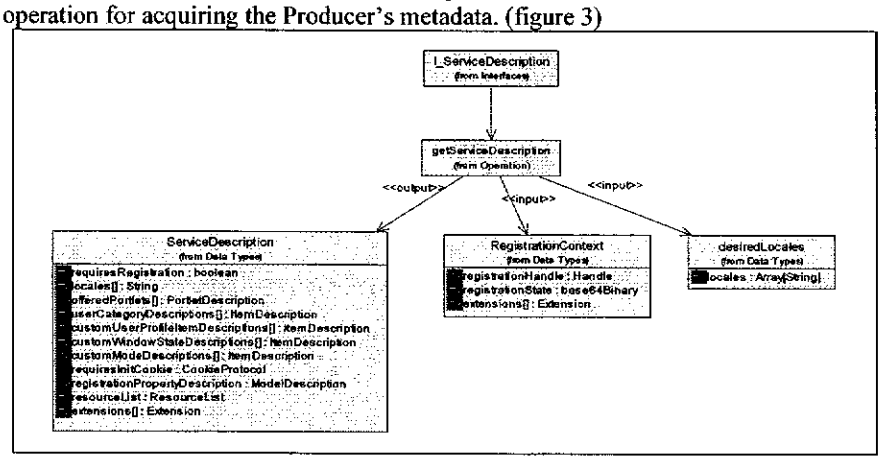

Fig. 3. ServiceDescription subontology

The well-formedness rules that complement the *ServiceDescription* subontology are the following:

```
context getServiceDescription
   inv allowedFaults:
(self.fault.name="InvalidRegistration"
or self.fault.name="OperationFailed")
```

```
context getServiceDescription
      inv resultKind:
self.result.isKindOf(ServiceDescription)
```

Secondly, the markup subontology is presented (Figure 4). The Markup interface has operations to generate markup and process interactions with that markup.

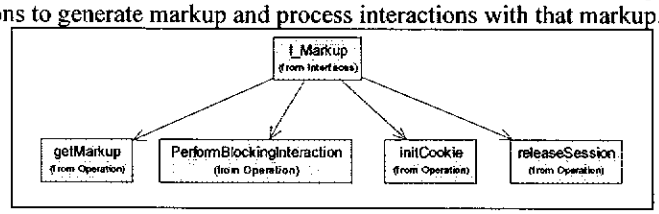

Fig. 4. Markup subontology

This subontology is composed of four different operations: a) *getMarkup*: renders the current state of a portlet (figure 5); b)*performBlockingInteraction*: sends user interactions to the producer; c) *initCookie*: provides assistance to initialize the cookies; d) *releaseSession*: informs the producer that it will no longer be using a set of sessions. The figures for the last three operations are not shown in this paper.

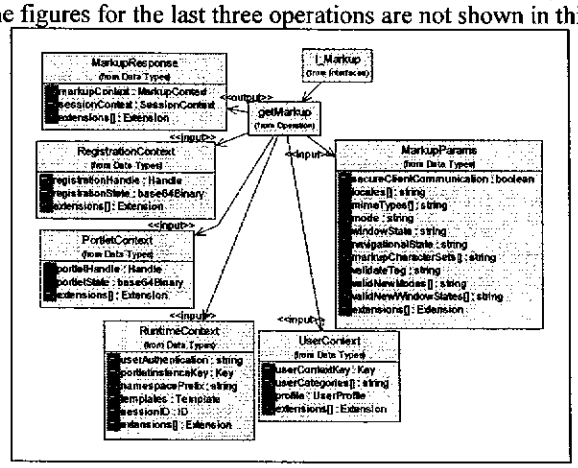

Fig. 5. getMarkup operation (it is part of Markup subontology)

The well-formedness rules for the initCookie operation are (it is similar for the rest of operations):

```
context getMarkup
inv allowedFauls:
 (self.fault.name="AccessDenied) or
self.fault.name="InconsistentParameters
)) or
 self.fault.name="InvalidRegistration"
or
 self.fault.name="MissingParameters" or
 self.fault.name="OperationFailed" or
 self.fault.name="InvalidUserCategory"
or
self.fault.name="InvalidHandle" or
 self.fault.name="InvalidCookie" or
 self.fault.name="InvalidSession" or
 self.fault.name="UnsupportedMode" or
self.fault.name="UnsupportedWindowState
" or
 self.fault.name="UnsupportedLocale" or
self.fault.name="UnsupportedMimeType")
context getMarkup
inv resultKind
 self.result.isKindOf(MarkupResponse)
```

Thirdly, the registration subontology is shown. The registration interface provides the necessary operations to allow a consumer to register with a producer [8].

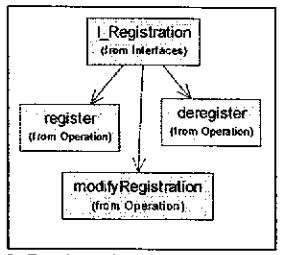

Fig. 9. Registration subontology

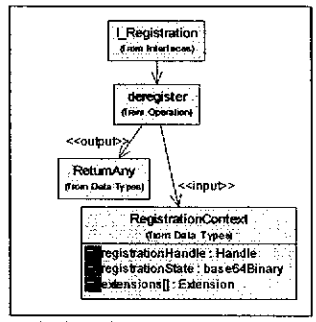

Fig. 10. deregister operation (it is part of Registration subontology)

As we can see in the subontology (figure 9), the operations for the Registration interface are: a) *register*: lets a consumer register with a producer, b)*modifyregister*: lets a consumer modify an existing relationship with a producer; c) *deregister*: lets a consumer terminate a registration (figure10). The register and modifyregister operations are not presented here.

The well-formedness rules for the deregister operation are (the rules for the rest of operations are built in a similar way):

```
context deRegister
 inv allowedFauls:
 (self.fault.name="InvalidRegistration"
or
   self.fault.name="OperationFailed")
```

```
context deregister
 inv resultKind:
   self.result.isKindOf(ReturnAny)
```

In addition, the PortletManagement subontology has been built. The purpose of the PortletManagement interface is to let consumers manage the persistent state and lifecycle of portlets explicitly [8].

Finally, a DataType subontology focused on the allowed data types has been included into the ontology for WSRP. The objective of this subontology is to provide and constrain the different data types which can be used and specify their structure. Due to the lack of space, the last two subontologies are not presented in this paper.

4 Conclusions and future work

In this paper an ontology for WSRP standard, namely WSRP-O, has been proposed. Its objective is to help portlet developers to create or adapt a portlet according to WSRP. Thus, the portlet developer can easily establish the specific elements for portlets which have to be included. Moreover, the portlet developer can concentrate on the logic domain instead of the portlet characteristics.

It must be emphasized that the WSRP standard is generic, it is not specific of any programming language. All portlets must be conformed to the WSRP standard (independently of its programming language), therefore the WSRP-O ontology (which is independent of the programming language) can be used to ensure this fact. For this reason, different ontologies are under development. Each one of these ontologies will be specific for a given programming language.

As a result, we will have the ontology of the WSRP standard which will be generic. Using this ontology the portlet developer can establish the necessary generic elements. Next, taking into account the programming language in which the portlet has to be developed, the ontology specific to this programming language has to be used.

Acknowledgments. This work is part of the DIMENSIONS (PBC-05-012-1), ENIGMAS (PBI-05-058), FAMOSO (2007: FIT-340000-2007-71) and ESFINGE (TIM2006-15175-C05-05) projects and CALIPSO network (TIN2005-24055-E).

References

1. Blattman, J., N. Krishnan, D. Polla, and M. Sum. Open-Source Portal Initiative at Sun, Part 2: Portlet Repository. http://developers.sun.com/prodtech/portalserver/reference/techart/portlet-repository.html#2 (2006)
2. Calero, C., F. Ruiz, A. Baroni, F. Brito e Abreu, and M. Piattini, An Ontological Approach To Describe the SQL:2003 Object-Relational Features. Accepted in "Computer Standards and Interfaces", vol. 28(6): pp. 695-713. (2006)
3. Corcho, O., M. Fernández-López, and A. Gómez-Pérez, Chapter 1: Ontological Engineering: Principles, Methods, Tools and Languages., in Ontologies for Software Engineering and Software Technology, C. Calero, Ruiz, F., Piattini, M., (Ed)., Springer-Verlag: New York (2006)
4. Java Community Process. JSR 168 portlet specification. http://www.jcp.org/en/jsr/detail?id=168 (2003)
5. Linwood, J. and D. Minter, Building Portals with the Java Portlet API. New York: Apress.(2004)
6. Montejava. Los Portales Empresariales sacan el máximo partido a las aplicaciones de negocio. http://www.montejava.es/articulo3.asp, (2006)
7. Neches, R., R.E. Fikes, T. Finin, T.R. Gruber, R. Senator, and W.R. Swartout, Enabling technology for knowledge sharing. AI Magazine, vol.12(3): pp. 87-127 (1991)
8. OASIS. Web Service for Remote Portlets (WSRP). Version 1.0. http://www.oasis-open.org/commitees/wsrp/ (2003)
9. PlanetMDE. MDE (Model Driven Engineering). Community portal. http://www.planetmde.org (2005)
10. Ruiz, F. and J.R. Hilera, Chapter 2. Using Ontologies in Software Engineering and Technology, in Ontologies for Software Engineering and Software Technology, C. Calero, F. Ruiz, and M. Piattini, (Eds), Springer Verlag: New York (2006)