Stanford Linear Accelerator Center (SLAC)
Menlo Park, California
Main Program: **July 12-14, 2006**
Workshops and Tutorials: **July 10-11, 2006**

Links:

Workshops

SMIWEP-MATeS'06

Joint Workshop on Web Services Modeling and Implementation Using Sound Web Engineering Practices$_1$ and Methods, Architectures & Technologies for e-Service Engineering$_2$

Joint Workshop on Web Services Modeling and Implementation Using Sound Web Engineering Practices[1] and Methods, Architectures & Technologies for e-Service Engineering[2]

(SMIWEP-MATeS 2006)

Workshop held in conjunction with the 6th International Conference on Web Engineering (ICWE2006)

Palo Alto, California, USA. July 10th, 2006

[1]Geert-Jan Houben (Vrije Universiteit Brussel)
[1]Oscar Pastor (Universidad Politécnica de Valencia)
[1]Daniel Schwabe (PUC-Rio)
[2]Oscar Corcho (University of Manchester)
[2]Pedro Sampaio (University of Manchester)
[2]Andreas Wombacher (University of Twente)

ABSTRACT

This workshop is the result of joining the two workshops "Web Services Modeling and Implementation Using Sound Web Engineering Practices" and "Methods, Architectures & Technologies for e-Service Engineering", which were originally proposed in conjunction with the ICWE 2006 conference. As a result of the selection process of both workshops, six papers were accepted and will be presented on July 10th, 2006.

Categories and Subject Descriptors

H.3.5 [Online Information Services]: Data sharing, Web-based services

Keywords

Web services, e-services, methods, models, architectures, technologies, implementations

1. INTRODUCTION

The joint workshop will cover best practices for engineering Web applications; the discussion of systematic, disciplined and quantifiable approaches for the cost-effective development and evolution of Web-based systems; and the comparison of best practices for modeling and implementation of web services.

The workshop will also discuss and disseminate the current state-of-the-art and state-of-the-practice in e-service engineering; The articulation and alignment of the methods, architectures and technologies underpinning the e-service engineering research domain; The development of new links and research collaborations between research and practitioner groups working in e-service engineering.

The topics will cover most of the aspects involved in web service engineering (development case studies, enterprise modeling, integration technologies, privacy and security, requirements engineering) and will also focus on novel aspects such as patterns and reference architectures, design methodologies and process and quality models.

2. WORKSHOP PROCEEDINGS

The call for papers and common proceedings for this joint workshop can be found in the URLs of each of the original workshops: SMIWEP[1] and MATeS[2]. The common workshop proceedings are published in both of the previous URLs as well.

The list of accepted papers is as follows:

- Extended Web Services Framework to Meet Non-Functional Requirements. Zafar U. Singhera and Abad Shah.
- A state propagation method for consistency checking of Web service function invocations in Web applications. Tomohiro Kaizu, Tomoya Noro and Takehiro Tokuda.
- Integrating Bioinformatic Data Sources over the SFSU ER Design Tools XML Databus. Yan Liu, Sorna Vincent, and Marguerite C. Murphy
- Ontology Driven Definition of a Usability Model for Second Generation Portals. Mª Ángeles Moraga, Coral Calero and Mario Piattini.
- A High-Level Specification for Semantic Web Service Discovery Services. Andreas Friesen and Egon Börger.
- Towards Semantic Service Selection for B2B Integration. Andreas Friesen and Kioumars Namiri

3. ACKNOWLEDGMENTS

We would like to give thanks to those who submitted their work in the form of papers, for their interest in the workshops, to the programme committee members of both workshops (they are listed in the workshop Web sites) and to the ICWE workshop organisers, Luis Olsina and Nora Koch, for their support during the workshop preparation.

[1] http://wise.vub.ac.be/smiwep2006/organizers.html

[2] http://www.informatics.manchester.ac.uk/mates06/

Ontology Driven Definition of a Usability Model for Second Generation Portals

Mª Ángeles Moraga
Alarcos Research Group.
UCLM-SOLUZIONA Research and Development Institute.
University of Castilla-La Mancha
Paseo de la Universidad,4 13071
Fax. 34 926 295 354
Tf. 34 926 295 300
MariaAngeles.Moraga@uclm.es

Coral Calero
Alarcos Research Group.
UCLM-SOLUZIONA Research and Development Institute.
University of Castilla-La Mancha
Paseo de la Universidad,4 13071
Fax. 34 926 295 354
Tf. 34 926 295 300
Coral.Calero@uclm.es

Mario Piattini
Alarcos Research Group.
UCLM-SOLUZIONA Research and Development Institute.
University of Castilla-La Mancha
Paseo de la Universidad,4 13071
Fax. 34 926 295 354
Tf. 34 926 295 300
Mario.Piattini@uclm.es

ABSTRACT

Second generation portals tend to be constructed by means of portlets, i.e. a multi-step, user-facing application to be delivered through a Web application. For this reason, the portal developers have to face up to the problem of choosing the best portlet among a set with the same functionality in order to construct portals with a good level of quality. This requires the existence of appropriate models to assess portlet's diverse quality characteristics. One of these quality characteristics is usability. In this paper we present deeply this portlet quality characteristic by means of a software measurement ontology. The use of the ontology avoids misunderstanding the concepts since it provides a consistent (i.e. both generally agreed, with consensus, and coherent (without conflicts and contradictions)) terminology. In such a manner, the usability model presented can be easily understood and applicable by any person because each element of the model is correctly and precisely defined.

Categories and Subject Descriptors

D.2.9 [**Software Engineering**]: Management – *Software Quality Assurance*

General Terms

Algorithms, Management, Measurement, Security.

Keywords

Ontology, quality model, measures, portals, portlets

1. INTRODUCTION

Over time web portals have undergone an evolution. They have grown from mere collections of static HTML pages to full-fledged business applications [12]. Two generations of portals can be distinguished: the first-generation portals tended to present a monolithic software architecture that compromises portal development and maintenance while the second-generation portals let users create one or more personal pages composed of personalizable *portlets* – interactive web miniapplications, local or remote to the portal, that render markup fragments (news, weather, sports and so on) that the portal can aggregate into a page [1].

In the second-generation portals if we want "good" portals, we must select the most appropriate portlets for building on them. For this reason, the portlet market requires the existence of a portlet quality model where all the characteristics related to the portlet quality will be defined and which could be used in the selection of the best portlet (among a set of portlets with the required functionality).

In [10] we have developed a portlet quality model. This model has been constructed taking into account works on web quality, software components quality and product software quality because portlets can be considered as a combination of them.

Among the characteristics of the portlet quality, we have identified the usability defined as the capability of the portlet to be understood, learned or used under specified conditions.

In this paper we present the development of the usability model, going from the definition of the characteristic to the definition of measures for calculating the usability level of a given portlet.

For presenting all this information we have used a software measurement ontology (SMO) presented in [3]. Several reasons moved us to use the SMO for defining the usability model. Firstly, vocabulary conflicts and inconsistencies can be frequently found amongst the many sources and references commonly used by software measurement researchers and practitioners. Secondly, the use of the ontology allows us to present in a formal way all our measures, using concepts and terminology that have been reached by consensus.

This paper is structured as follows. Section 2 gives a brief on software measurement ontology. Section 3 explains the model for usability characteristic. Last section summarizes this paper and proposes future work.

2. SOFTWARE MEASUREMENT ONTOLOGY (SMO)

Although software measurement plays an increasingly important role in Software Engineering, there is no consensus yet on many of the concepts and terminology used in this field, such as, 'measurement', 'measure', 'metric', 'measurable attribute', etc. Even worse, vocabulary conflicts and inconsistencies can be frequently found amongst the many sources and references commonly used by software measurement researchers and practitioners.

The situation is not much better if we take a look at the current software engineering international standards, such as IEEE, ISO and IEC. Inconsistencies and terminology conflicts appear not only between standards from different bodies, but also within those from the same organization.

With the goal of contributing to the harmonization of the different software measurement standards and research proposals, in [3] the authors propose a Software Measure Ontology. This ontology brings together and unifies a great amount of these proposals.

In concrete, the ontology was organized around four sub-ontologies:

- Software Measurement Characterization and Objectives, which includes the concepts required to establish the scope and objectives of the software measurement process.
- Software Measures, which aims at establishing and clarifying the key elements in the definition of a software measure.
- Measurement Approaches. This sub-ontology introduces the concept of measurement approach to generalize the different 'approaches' used by the three kinds of measures for obtaining their respective measurement results.
- Measurement. It establishes the terminology related to the act of measuring software.

In Figure 1, the UML diagram of the software measurement ontology is presented (each colour of the figure represents each one of the four subontologies), while in Table 1, Table 2, Table 3, Table 4, the concepts defined in the sub-ontologies are shown.

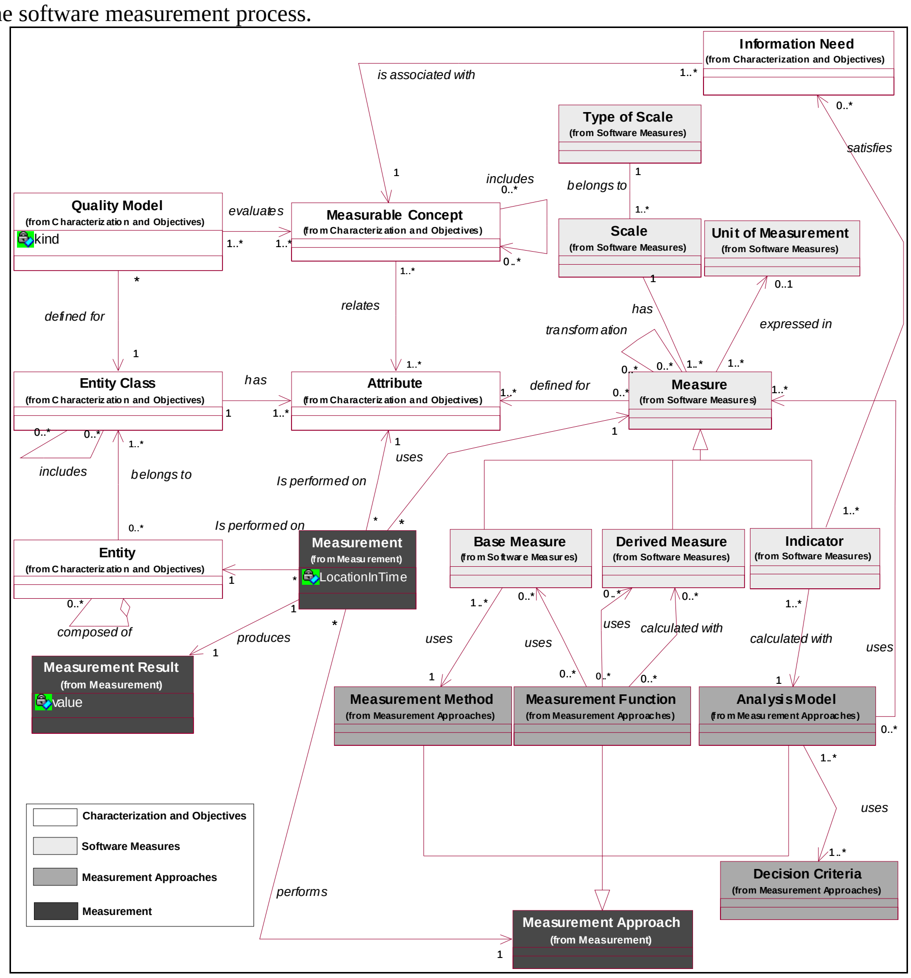

Figure 1. UML diagram of the Software Measurement Ontology.

Table 1. Concepts Table of the Sub-Ontology Characterization and Objectives

Term	Definition
Information Need	Insight necessary to manage objectives, goals, risks, and problems
Measurable Concept	Abstract relationship between attributes of entities and information needs
Entity	Object that is to be characterized by measuring its attributes
Entity Class	The collection of all entities that satisfy a given Predicate
Attribute	A measurable physical or abstract property of an entity, that is shared by all the entities of an entity class
Quality Model	The set of measurable concepts and the relationships between them which provide the basis for specifying quality requirements and evaluating the quality of the entities of a given entity class

Table 2. Concepts Table of the Sub-Ontology Software Measures

Term	Definition
Measure	The defined measurement approach and the measurement scale. (A measurement approach is either a measurement method, a measurement function or an analysis model)
Scale	A set of values with defined properties
Type of Scale	The nature of the relationship between values on the scale
Unit of Measurement	Particular quantity, defined and adopted by convention, with which other quantities of the same kind are compared in order to express their magnitude relative to that quantity
Base Measure	A measure of an attribute that does not depend upon any other measure, and whose measurement approach is a measurement method
Derived Measure	A measure that is derived from other base or derived measures, using a measurement function as measurement approach
Indicator	A measure that is derived from other measures using an analysis model as measurement approach

Table 3. Concepts Table of the Sub-Ontology Measurement Approaches

Term	Definition
Measurement Method	Logical sequence of operations, described generically, used in quantifying an attribute with respect to a specified scale. (A measurement method is the measurement approach that defines a base measure)
Measurement Function	An algorithm or calculation performed to combine two or more base or derived measures. (A measurement function is the measurement approach that defines a derived measure)
Analysis Model	Algorithm or calculation combining one or more measures with associated decision criteria. (An analysis model is the measurement approach that defines an indicator)
Decision Criteria	Thresholds, targets, or patterns used to determine the need for action or further investigation, or to describe the level of confidence in a given result

Table 4. Concepts Table of the Sub-Ontology Measurement

Term	Definition
Measurement Approach	Sequence of operations aimed at determining the value of a measurement result. (A measurement approach is either a measurement method, a measurement function or an analysis model)
Measurement	A set of operations having the object of determining the value of a measurement result, for a given attribute of an entity, using a measurement approach
Measurement Result	The number or category assigned to an attribute of an entity by making a measurement

In these tables, columns one and two show the term being described and its definition in the SMO, while figures show the graphical representation of the SMO terms and relationships, using UML.

3. MODEL FOR USABILITY CHARACTERISTIC.

In this section, the model for portlet usability will be presented in detail. Portlet usability is defined as the capability of the portlet to be understood, learned or used under specified conditions [2]. As pointed out in the previous section, we make use of the Software Measurement Ontology to use a consistent terminology. Thereby, the understanding of our model is easier since all the terms have a precise meaning. Next, the terms of the SMO for our context are defined.

The Entity Class is defined as the "set of portlets which meet the desirable functionality by the user" while an Entity is "a specific portlet which is measured".

Our Information Need is "to evaluate the usability of a set of portlets which provide similar functionality with the aim of choosing the best".

The Quality Model is the model that is being presented in this section.

This model evaluates the Measurable Concepts: understandability, learnability, customizability and compliance, all of them identified in [9] as subcharacteristics of the usability. In the next subsections we will present the study of each of these Measurable concepts.

3.1 Study of the measurable concept: understandability

Understandability mensurable concept is defined as the capability of the portlet to enable the user to understand what the portlet is about. This measurable concept relates the attributes: interface language, documentation, documentation language and description. All the attributes have been defined taking into account that portlets are like black boxes. Moreover, we consider that portlets adhere to the WSRP (Web Service for Remote Portlets) standard whose goal is to enable an application designer or administrator to pick from a rich choice of compliant remote content and application providers, and integrate them with just a few mouse clicks and no programming effort [11]. This standard defines standard interfaces, so, the only attributes definable are those that can be calculated using information obtained from their interfaces or information provided by the vendors. The meaning of the attributes is explained as follows: Interface language (the portlet interface supports different languages), Documentation (the portlet vendor provides the portlet with documentation on line. Hence additional information, which can help the portal administrator to understand the portlet, is provided), Documentation language (the documentation is provided in several languages), Description (this refers to the existence of a description of the portlet functionality, helping the end-user to understand it).

Next, let's define measures to measure each attribute. In Table 5, these measures along with their scale, type of scale and unit of measure are shown.

Table 5. Measures defined for attributes of the understandability measurable concept.

Measurable Concept	Attribute	Measure			
		Base Measure	Scale	Type of Scale	Unit of Measurement
Understan-dability	Interface language	Number of languages supported by the Interface	Natural Number	Ratio	Languages
	Documentation	The portlet vendor provides documentation	Boolean (0/1)	Nominal	Documentation
	Documentation language	Number of languages in which the documentation is written	Natural Number	Ratio	Languages
	Description	The portlet specifies its functionality	Boolean (0/1)	Nominal	Description

3.2 Study of the measurable concept: understandability

The learnability measurable concept is defined as "the capability of the portlet to enable the user to learn how the portlet achieves its aim".

This measurable concept relates the following attributes: help, documentation, predictability, screens and structured presentation. The meaning of them is:

- Help: the portlet supports *help mode* (i.e. the portlet may provide help screens that explains the portlet and its expected usage [11]).
- Documentation: the portlet vendor provides the portlet with documentation on line. So additional information is provided.
- Predictability: portlet interface icons are easily related to the actions the portlet performs.
- Screens: it refers to the number of screens for achieving one functionality.
- Structured presentation: the presentation of the portlet is structured and easy to understand.

The different measures defined for these attributes are presented in Table 6.

Table 6. Measures defined for attributes of the learnability measurable concept.

Measurable Concept	Attribute	Measure			
		Base Measure	Scale	Type of Scale	Unit of Measurement
Learnability	Help	The portlet provides *help mode*	Natural number (Among 0 and 2)	Ratio	Helps
	Documentation	The portlet vendor provides documentation	Boolean (0/1)	Nominal	Documentation
	Predictability	Portlet predictability	List (1...5): very difficult, difficult, normal, easy, very easy	Enumerate	Difficulty level
	Screens	Number of screens in order to achieve one functionality	Natural Number	Ratio	Screens
	Structured presentation	Structured presentation degree	List (1...5): very difficult, difficult, normal, easy, very easy	Enumerate	Structure Degree

3.3 Study of the measurable concept: customizability

Customizability measurable concept refers to the attributes of software that enable the portlet to be customized by the user, to reduce the effort required to use it and also to increase satisfaction with the software.

The attributes which have been identified are: location, localization, time, device, network, user, windowStates, CSS, edit mode, necessary parameters, categories of users and content depends on configuration.

The meaning of the attributes is explained as follows:

- Location: the portlet captures information about the location from which it is accessed.
- Localization: is the capacity to tailor one portlet to the idiosyncrasies of a given culture- this is becoming an increasing concern. The aspects of cultural diversity that need specific support are normally arranged around two features, namely, *language* and *country*.
- Time: the portlet allows the adaptation of the application with respect to certain timing constraints.
- Device: this attribute discusses the demand of ubiquitous Web applications for any media, in terms of multi-channel delivery, and it provides basic information about the hardware and software capabilities of the device accessing the application.
- Network: the portlet can adapt itself to different networks. This attribute considers adaptation from the network

viewpoint, and whether network context information, e. g. bandwidth or package losses, affects the application.

- User: the portlet takes into account the personal characteristics of the user. This attribute regards the need for personalization, i.e. how the user profile (e. g. demographic data, knowledge, skills and capabilities, interests and preferences, goals and plans) is considered by the application.
- WindowStates: space left for portlet rendering. WSRP defines five windowStates: *normal,* indicates the portlet is in all likelihood sharing the aggregated page with other portlets; *minimized*, the portlet should not render visible markup, but it is free to include non-visible data such as JavaScript or hidden forms; *maximized,* specifies that the portlet is probably the only one being rendered in the aggregated page, or that the portlet has more space compared to other portlets in the aggregated page; *solo,* denotes that the portlet is the only portlet being rendered in the aggregated page; *custom,* for consumers to declare additional custom windowStates.
- CSS: the portlet considers aesthetic guidelines for preserving the identity of the portal.
- Edit mode: the portlet provides the end-user with a mode, namely *edit mode*, for configuring the portlet. Within this mode, a portlet should provide content and logic that let a user customize the behaviour of the portlet.
- Necessary parameters: relationship between the number of parameters which are requested of the end-user and the number of parameters that the portlet actually uses in order to adapt the portlet to him/her.
- Categories of users. The portlet supports communities: the content generated depends on the category of the user who is interacting with the portlet.
- Content depends on configuration: The portlet can tailor its generated content (in the *mode view* – used to render markup reflecting the current sate of the portlet [11]) to specific users depending on the configuration (windowState, categories of users, user profile, user's preferences, etc.).

Table 7 shows the measures defined for the attributes outlined above.

Table 7. Measures defined for attributes of the customizabity measurable concept.

Measurable Concept	Attribute	Measure				
		Derived Measure	Base Measure	Scale	Type of Scale	Unit of Measurement
Customi-zability	Location	--------	Location availability	Boolean (0/1)	Nominal	Location
	Localization	Total of languages and countries to which the portlet can be tailored.	Number of languages	Standard language list	Nominal	--------
			Number of countries	Standard country list	Nominal	--------
	Time		Time adaptation availability	Boolean(0/1)	Nominal	Adaptation to the time
	Device	Total of markup and user agents that the portlet supports.	Markup types that the portlet supports.	List	Nominal	--------
			User agents that the portlet supports.	List	Nominal	--------
	Network	--------	Network adaptation availability	Boolean (0/1)	Nominal	Adaptation to the network
	User	--------	Number of user profile characteristics that the portlet stores.	Natural number	Ratio	User profile characteristics
	WindowStates	--------	Number of additional windowStates supported by the portlet	Natural number	Ratio	WindowStates
	CSS	--------	CSS availability	Boolean(0/1)	Nominal	CSS
	Edit mode	--------	Edit mode availability	Boolean(0/1)	Nominal	Edit mode
	Necessary parameters	Ratio of necessary parameters	Parameters_user_fill_in	Natural number	Ratio	Parameters filled in by the user
			Parameters_portlet_user	Natural number	Ratio	Parameters used by the portlet
	Categories of users	--------	Number of categories of users	Natural number	Ratio	Categories of users
	Content depends on configuration	--------	Content generated by the portlet depends on the configuration	Boolean (0/1)	Nominal	Depends on the configuration

It may be worth emphasized that three of the previous measures are derived measures. Consequently they are derived from other base measures.

For example, the derived measure "Ratio of necessary parameters" will be calculated using the results of the base measures "Parameters_user_fill_in" and "Parameters_portlet_use" and a measurement function (to combine the base measures). The measurement function is:

$$RNP= \frac{Parameters\_user\_fill\_in}{Parameters\_portlet\_use}$$

Where:

- RNP: is Ratio of necessary parameters.
- Parameters_user_fill_in: is the number of parameters that the user must provide to the portlet.
- Parameters_portlet_use: is the number of parameters that the portlet actually uses.

The ideal case is when "necessary parameters" takes the value 1 because this means that the portlet uses all the parameters it asks the user for.

The information about the derived measures (i.e. its measurement function, scale, type of scale and domain) is shown in Table 8.

Table 8. Information related to the derived measures.

Derived Measure	Measurement Function	Scale	Type of Scale	Unit of Measurement
Total of languages and countries to which the portlet can be tailored.	Number of languages + Number of countries	Natural number	Ratio	Languages-Countries
Total of markup and user agents that the portlet supports.	Markup types that the portlet supports + User agents that the portlet supports	Natural number	Ratio	Markup-UserAgents
Ratio of necessary parameters	$\frac{Parameters\_user\_fill\_in}{Parameters\_portlet\_use}$	Decimal number	Ratio	Necessary parameters

3.4 Study of the measurable concept: compliance

Compliance measurable concept is the capability of the portlet to adhere to standards, conventions or regulations in laws and similar prescriptions relating to usability.

This measurable concept relates the following attribute: standard compliance. The meaning of this attribute is:

- Standard compliance: the portlet adheres some of the different usability standards which have been proposed in the literature. Specifically, these standards are: ISO 9241 [7], ISO 14915 [8], IEC CDV TR 61997 [5], ISO DTS 16071 [6] and Usability Engineering Process Model [4].

The measure defined for this attribute is shown in Table 9

Table 9. Measure defined for the attribute of the compliance measurable concept.

Measurable Concept	Attribute	Measure			
		Base Measure	Scale	Type of Scale	Unit of Measurement
Compliance	Standard compliance	The portlet adheres to usability standards	Natural number	Ratio	Standards

3.5 Indicators for usability

Once derived and base measures have been identified, the reusability indicator can be calculated. Firstly, indicators for the different measurable concepts will be defined. Then, these indicators will be transformed (by using an analysis model) into others to obtain the usability indicator. In Figure 2, the different indicators that must be defined and its transformations are shown.

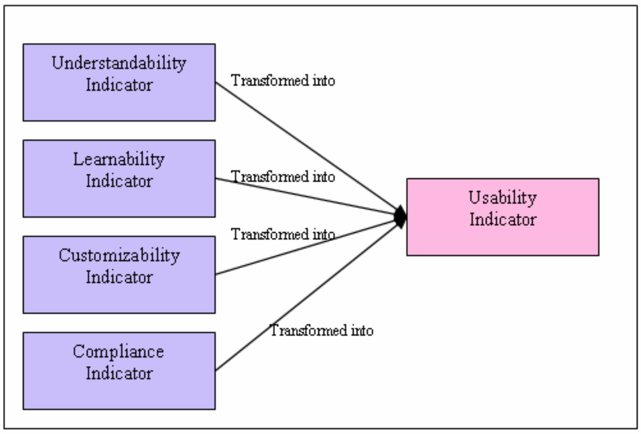

Figure 2. Indicators to obtain the Usability Indicator

Next, the process to calculate the value of the different indicators is explained. In general, it is necessary to use an analysis model as measurement approach. In concrete, our analysis model is a vector composed of different elements, where each element represents one measure.

For example, the vector for the **understandability indicator** (UND-IND) is:

V(UND-IND)= (Number of languages supported by the Interface, The portlet vendor provides documentation, Number of languages in which the documentation is written, The portlet specifies its functionality)

Therefore, each element is one understandability measure (defined in section 3.1). The next step consists in defining decision criteria to describe the level of portlet understandability (for the case of the understandability indicator), or portlet learnability (for the case of the portability indicator), and so on. Decision criteria are thresholds, targets, or patterns used to determine the need for action or further investigation, or to describe the level of confidence in a given result. Finally, the indicator type scale, as well as its values, is defined. Depend on each case, the values can be: "excellent, high, middle, acceptable and non-acceptable" or "excellent, acceptable, non-acceptable". In all cases the type scale is ordinal.

In Table 10, the understandability indicator values and decision criteria are defined.

Table 10. Values and decision criteria for understandability indicator

UNDERS-TANDABILITY INDICATOR VALUES	*Decision criteria*			
	Number of languages supported by the Interface	The portlet vendor provides documentation	Number of languages in which the documentation is written	The portlet specifies its functionality
Excellent	2^+	1	2^+	1
Acceptable	2	1	1	1
Non-accept.	Rest			

We can notice that we have identified one decision criterion for each element of the vector. Likewise, it may also be worth underlining that a "+" as superscript of the number indicates that the value for this element is this or greater. In addition, the [X-Y] expression, which will be used in the next table, indicates that the value is a number between X and Y. Therefore, depending on the value of each element of the vector V(UND-IND) and taken into account the decision criteria, a portlet can obtain an understandability value among excellent and non-acceptable. For example, the understandability level obtains an excellent value if the number of languages supported by the interface of the portlet is two or more and the portlet provides documentation which is written in two or more languages and a description of its functionality.

Next, the different functions for the rest of indicators, as well as its decision criteria are presented.

The **learnability indicator** (LEARN-IND) vector is:
V(LEARN-IND)= (The portlet provides help, The portlet vendor provides documentation, Portlet predictability, Number of screens in order to achieve one functionality, Structured presentation degree)
The values and decision criteria for learnability indicator are presented in Table 11.

Table 11. Values and decision criteria for learnability indicator

LEARNA-BILITY INDICATOR VALUES	*Decision criteria*				
	The portlet provides help	The portlet vendor provides documentation	Portlet predictability	Number of screens in order to achieve one functionality	Structured presentation degree
Excellent	2	1	Very easy	[1-3]	Very easy
High	2	1	Easy	[1-4]	Easy
Middle	1	1	Normal	[1-4]	Normal
Acceptable	0	1	Difficult	[1-5]	Difficult
Non-accept.	Rest				

The vector for the **customizability** indicator (CUS-IND) is:
V(CUS-IND)= (Location availability, Total of languages and countries to which the portlet can be tailored, Time adaptation availability, Total of markup and user agents that the portlet supports, Network adaptation availability, Number of user profile characteristics that the portlet stores, Number of additional windowStates supported by the portlet, CSS availability, Edit mode availability, Ratio of necessary parameters, Number of categories of users)
In Table 12, the values as well as the decision criteria for customizability indicator are shown.

Table 12. Values and decision criteria for customizability indicator.

CUSTOMIZA-BILITY INDICATOR VALUES	*Decision criteria*											
	Location availability	Total of languages and countries to which the portlet can be tailored	Time adaptation availability	Total of markup and user agents that the portlet supports	Network adaptation availability	Number of user profile characteristics that the portlet stores.	Number of additional windowStates supported by the portlet	CSS availability	Edit mode availability	Ratio of necessary parameters	Number of categories of users	Content generated by the portlet depends on the configuration
Excell.	1	6^+	1	6^+	1	22^+	4	1	1	1	1	1
High	1	4	1	4^+	0	[20-22]	3	1	1	1.25	0	0
Middle	1	2	0	2	0	[17-20]	3	1	1	1.5	0	0
Accept.	1	2	0	2	0	17	3	1	1	2	0	0
Non-acc	Rest											

The vector for the **compliance** indicator (COM-IND) is:
V(COM-IND)=(The portlet adheres to usability standards)
The values and the decision criteria for compliance indicator are defined in Table 13.

Table 13. Values and decision criteria for compliance indicator.

COMPLIANCE INDICATOR VALUES	*Decision criteria*
	The portlet adheres to usability standards
Excellent	The portlet adheres to 4 standards and to the model
High	The portlet adheres to four
Middle	The portlet adheres to three
Acceptable	The portlet adheres to one or two
Non-acceptable	Rest

Finally, the understandability, learnability, customizability and compliance indicators are transformed into the **usability indicator**.In Table 14, the decision criteria to carry out this transformation are presented.

Table 14. Values as well as decision criteria for usability indicator

USABILITY INDICATOR VALUES	***Decision criteria***
Excellent	All the indicators have obtained the excellent level.
High	All the indicators have obtained the high level. There exists a combination of excellent and high levels. There exists a combination of excellent and middle levels where the number of excellent levels is equal or bigger than the number of middle levels
Middle	All the indicators have obtained the middle level. There exists a combination of high and middle levels. There exists a combination of excellent and middle levels where the number of middle levels is bigger than the number of excellent levels. There exists a combination of excellent (or high) and acceptable levels where the number of excellent (or high) levels is bigger than the number of acceptable levels.
Acceptable	All the indicators have obtained the acceptable level. There exists a combination of excellent (or high or middle) and acceptable levels where the number of acceptable levels is bigger than the number of excellent (or high or middle) levels.
Non-acceptable	Rest of the cases

4. CONCLUSIONS AND FUTURE WORK

Portals have undergone an evolution in such a way that, nowadays, second-generation portals are constructed by means of portlets. For this reason, the portal developers have to face up to the problem of choosing the best portlet among a set of portlet with the same functionality.

With the aim of helping portal developers, a portlet usability model has been presented. But, as it is well known, the lack of a common terminology and inconsistencies between the different standards and proposals, related to software measurement, makes difficult to develop an understandable model.

For this reason and with the objective of using a consistent terminology (both generally agreed and coherent), we have made use of the software measurement ontology to develop our model.

In future work, we have to validate through surveys done by experts, both the attributes and the measures proposed for the usability, but especially the decision criteria used to obtain the portlet usability value.

Also, we plan to define formally the other characteristics identified for portlet quality until we will have dealt with all characteristics that affect that quality.

The final goal is to have a quality model that could be used, on one hand, to decide on the best portlet among a set of portlet with similar functionality, and, on the other hand, to identify possible improvements in the quality of a given portlet. In addition, we will automate the ontology in a system and we will develop a tool on top of the ontology to automate the assessment method.

5. ACKNOWLEDGMENTS

This work is part of the CALIPO project (TIC 2003-07804-C05-03) and the CALIPSO network (TIN2005-24055-E) supported by the Spanish Ministerio de Educación y Ciencia and by the DIMENSIONS project (PBC-05-012-1) supported by FEDER and Junta de Comunidades de Castilla-La Mancha.

6. REFERENCES

[1] Bellas, F., Standards for Second-Generation Portals. IEEE Internet Computing., 2004. 8(2): pp. 54-60.

[2] Diaz, O., C. Calero, M. Piattini, and A. Irastorza. Portlet usability model. 2004. IBM Research Report. RA221(W0411-084). ICSOC 2004.pp. 11-15.

[3] García, F., M.F. Bertoa, C. Calero, A. Vallecillo, F. Ruíz, and M. Genero, Towards a consistent terminology for software measurement. Information and Software Technology, 2005 To be published.

[4] Granollers, T., J. Lorés, and F. Perdrix. Usability Engineering Process Model. Integration with Software Engineering. 2003. Procs. of HCI-Intl'03. Crete-Greece

[5] IEC CDB TR 61997, Guidelines for the user interfaces in multimedia equipment for general purpose use. 2000.

[6] ISO, ISO DTS 16071: Guidance on accessibility for human-computer interfaces. 2000.

[7] ISO/IEC, ISO 9241. Ergonomic requirements for office work with visual display terminals. 1999.

[8] ISO/IEC, ISO 14915. Software ergonomics for multimedia user interfaces. 2000.

[9] Moraga, M.Á., C. Calero, M. Piattini, and O. Diaz, Improving a Portlet Usability Model. Submitted to: Software Quality Journal, 2005.

[10] Moraga, M.Á., C. Calero, M. Piattini, and O. Diaz. PtSM: A Portlet Selection Model. 2006. Submitted to: The 7th International Conference on Web Information System Engineering.

[11] OASIS. Web Service for Remote Portals (WSRP) Version 2.0. 2005. Available on: http://www.oasis-open.org/commitees/wsrp/. Accessed: June 2005.

[12] Pettersson, U. and S. Jarzabek. Industrial experience with building a web portal product line using a lightweight, reactive approach. 2005. ACM SIGSOFT Software Engineering Notes, Proceedings of the 10th European software engineering conference held jointly with 13th ACM SIGSOFT International Symposium on Foundations of software engineering ESEC/FSE-13 Vol.30(5).pp. 326-335.