

A Framework For Classifying and Comparing Web Services Procurement Platforms

Octavio Martín-Díaz, Antonio Ruiz-Cortés, Rafael Corchuelo, Miguel Toro
Dpto. Lenguajes y Sistemas Informáticos
E.T.S. Ingeniería Informática, Universidad de Sevilla
41012 Sevilla, Spain
{octavio,aruiz,corchu,mtoro}@lsi.us.es

Abstract

The procurement of web services (WSP) shifts the focus of architects and integrators from functional to quality-of-service (QoS) aspects. A number of platforms have been proposed as supporting infrastructures to assist WSP activities. Notwithstanding, there is a lack of consensus on (i) what expressiveness degree in QoS specifications should be offered, and (ii) what activities should be supported by these platforms. This paper attempts to provide an answer to both questions, thus it motivates and presents a classification framework for WSP platforms. This framework is used to compare several existing platforms and to identify some key properties and deficiencies, which might be considered as a research agenda for the future.

1 Introduction

The incredible successfulness of the Internet world has paved the way for a sub-industry devoted to developing and executing web services, which some authors view as the core of the next-generation Internet. Built on the standard integration fabric of XML and widely-accepted Internet protocols, web services bring programmers a new way to develop advanced applications that can integrate any group of services on the Internet into a single solution.

In this context, the WSP, considered as a particular case of software procurement [5], is focussed on acquisition of web services that applications require, so it is becoming a critical point for current web-service-based system developers.

There are a number of reasons that also lead us to automate those tasks with regard to WSP:

- The increasing size of web service repositories, whose contents can be also subject to unexpected changes.

- The increasing number of QoS parameters to have taken into account to select the best web service. QoS parameters often bias customers against using a web application that does not live up to the hype, even though it is fully functional.
- The need of carrying out this activity while the application works without shutdowns, so applications are able to use the best web services, according to their needs, at any time.

The WSP shifts the focus of architects and integrators from functional to QoS aspects. Although WSP is being supported by several platforms from industry and academic [1, 2, 9, 13], they have been mainly focussed on functionality provided by web services, but not on their QoS. Nowadays, there is a new emerging generation of platforms which are proposed to support infrastructures to assist WSP activities in a quality-aware manner [4, 6, 7, 8, 11].

Notwithstanding, there are proposals with a very limited expressiveness when describing QoS. All proposals allow demands to be specified with (more or less) complex expressions, but [4] and [11] only allow offers to be specified by means of name-value pairs, whereas [6] and [7, 8] do allow offers to be specified with the same expressiveness as demands. As well, [4] and [7, 8] allow offers to specify conditions on demands, and [7, 8] allow demands to include assessment criteria to optimise the search of offers. On the other hand, there is also a lack of consensus on terminologies which is used to name both elements and activities involved in WSP, and the way these activities are supported by platforms.

This paper attempts to provide an answer to these points, thus it motivates and presents a classification framework for WSP platforms. This framework is used to compare several existing platforms and to identify some key properties and deficiencies, which might be considered as a research agenda for the future.

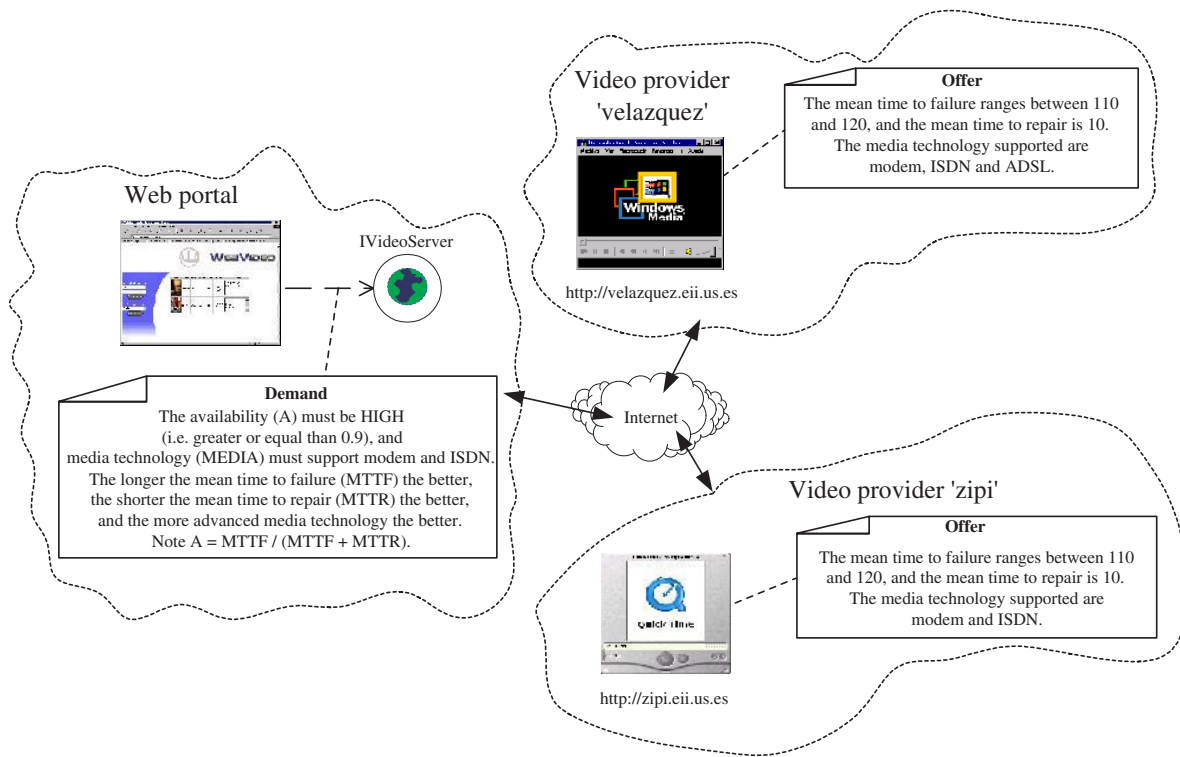


Figure 1. A fragment of the web video portal architecture.

The rest of the paper is organised as follows. First, section 2 introduces a case of study to illustrate the WSP and its main elements and activities. Next, section 3 describes our classification and comparison framework for WSP. Then, section 4 summarises some related proposals according to this framework and show some examples regarding with the case study in section 2. Finally, section 5 presents our conclusions and future research.

2 A Case of Study

Consider, for instance, that someone is interested in setting up a web portal specialised in films, so that it offers a potentially infinite catalogue of films and the same functionality as a domestic video player. In order to achieve such goal, the system should include, at least, a service for streaming videos on the Internet, a service for managing catalogues and keeping them up-to-date, and a service for managing virtual shops. These services should be procured having their QoS taken into account. In this way, the web portal becomes a composed service that integrates web services provided by several organisations.

Figure 1 shows a fragment of the component view corresponding to this portal. The IVideoServer interface abstracts those operations a component able to deliver video on demand needs to have if we want to incorporate it into

the system at run-time. There is also a note associated with the IVideoServer interface, stating the demand on QoS that a web service should fulfill so that it can be eventually used by the portal, and the remaining notes are associated with web services Microsoft Windows Media Server and Apple Quick Time Server, stating the offers on QoS their providers guarantee. In this case, the involved QoS parameters are *MTTF* (Mean Time to Failure), *MTTR* (Mean Time To Repair), and *MEDIA* (Media Support). QoS parameters can be derived from others: as an example, the Availability QoS parameter is usually defined as $MTTF / (MTTF + MTTR)$.

If we read all demand and offers, we can check that all of them do not contain inner contradiction; in this case, we say they all are consistent. On the other hand, if we read the offers, we can check that all of them fulfill the demand which is needed to be contracted by the portal; in this case, we say they all are conformant. As both offers are conformant to the given demand, then we should select the optimum offer according to the assessment criteria included in the demand [3]. As an example, figure 1 also includes several statements which denote such assessments: the longer *MTTF* the better, the shorter *MTTR* the better, and the more advanced media support is used the better. According to them, the provider whose web service owns the best offer is Velazquez.

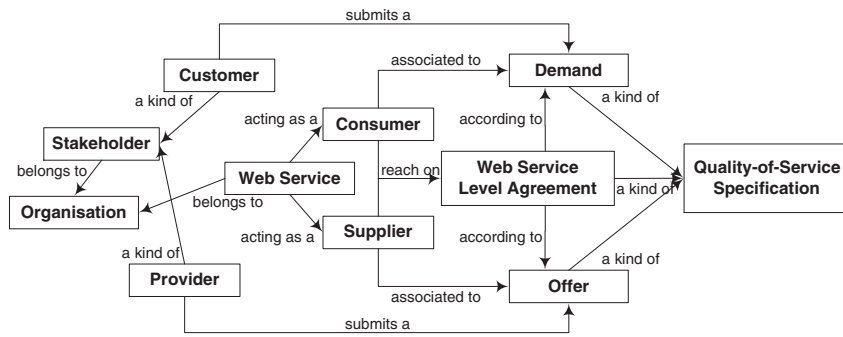


Figure 2. WSP reference model: process model static view.

3 An “Ideal” Platform for WSP

In this section, we introduce our classification and comparison framework for WSP platforms. This framework is composed of a reference model and a set of features we think any proposal should own.

3.1 A reference model

The main elements (from a static point of view) and activities (from a dynamic point of view) involved in WSP are introduced in the following:

3.1.1 Static view - the lexicon

Figures 2 and 3 show a conceptual map describing the relationships among the main elements involved in WSP, which are defined in the following:

- *Organisations, stakeholders & web services.* An organisation stands for any entity involving stakeholders and web services. In turn, a stakeholder stands for any people which belongs to an organisation, be either a customer or a provider. On the other hand, a web service stands for any software accessed via the Web which belongs to an organisation, be either a consumer or a supplier. A customer submits demands associated to a service which is acting as a consumer, and a provider submits offers associated to a service which is acting as a supplier.
- *Quality-of-service specifications.* A QoS specification is referred to a demand, an offer, or an agreement with regard to a web service. In general, it is defined in terms of some catalogues, and basically it contains a set of clauses stating (i) the requirements/provisions upon QoS parameters, and (ii) the assessment criteria. Briefly:

- *Demands & offers.* A demand states those conditions on QoS regarding with a web service which a customer is asking for. On the other hand, an offer states the QoS that a provider guarantees.
- *Web service level agreements (WSLA).* A WSLA stands for a kind of contract between a customer and the provider of a web service, so that the latter is pledged to fulfill it when the first uses it.
- *Catalogues, parameters & measures.* A parameter stands for any observable characteristic of a service which may affect the consideration of stakeholders. Each parameter is associated to a measure, which basically defines (i) the domain of values it can take, and (ii) the way it is measured. Services can be characterised by current measures of QoS parameters. A parameter is said to be basic if it can be directly measured (as an example, *MTTF* and *MTTR*), whereas it is said to be derived if it is computed from others (as an example, Availability). On the other hand, a catalogue stands for a (hierarchical) organisation which groups related QoS parameters. As an example, a catalogue devoted to reliability includes parameters as *MTTF*, *MTTR*, and so on.
- *Assessment criteria.* QoS specifications may contain the criteria which stakeholders have regarding with QoS parameters. As an example, a customer can demand a $MTTF \geq 90$, but he/she can fix some preferences on QoS values when searching for a web service, establishing that a value of $MTTF = 95$ is preferred to $MTTF = 90$, even though both QoS values are valid the same.
- *Negotiation clauses.* A QoS specification may contain alternate options in demands or offers to be applied in case of arising conflicts. As an example, a negotiation clause can include the renunciation to several QoS clauses, or a trade-off so that some QoS clauses are left out, in exchange for other less restrictive ones.

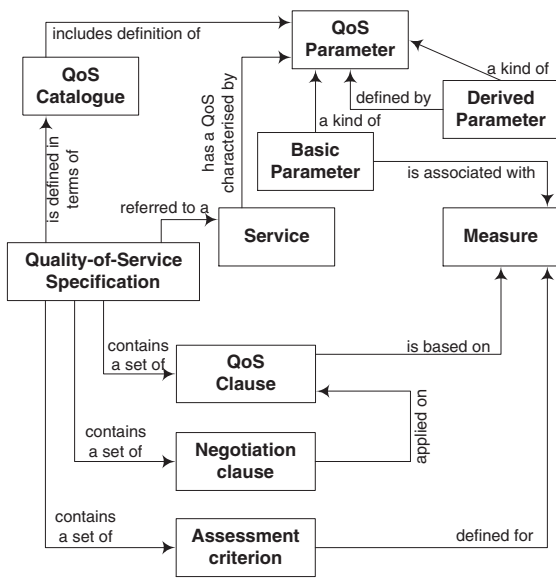


Figure 3. WSP reference model: QoS specification static view.

3.1.2 Dynamic view - the process model

The main activities involved in the WSP process model are defined in the following:

- *Creation & submission of catalogues.* It carries out the definition of catalogues containing descriptions of QoS parameters. It includes these tasks: (i) an analysis for inconsistencies, (ii) a verification for an appropriate construction, and (iii) a final validation for submission and usage.
- *Creation & submission of demands/offers.* It carries out the definition of QoS specifications, be either demands or offers, to be submitted to the system. It includes these tasks: (i) an acquisition of catalogues, (ii) an analysis for inconsistencies, (ii) a verification for an appropriate construction, and (iii) a final validation for submission.
- *Matchmaking.* It carries out the search of the web service which better fulfills every customer's demand, in order to attain a WSLA between the customer and its provider.

3.2 Target Features

We also have need of defining a criteria set that allows us to compare some interesting capabilities of current proposals, as objective as possible. In order to do so, we have identified three major features, namely: expressiveness, formality and viability.

- *Expressiveness.* This is a highly subjective concept depending on the context it is used. We consider expressive that proposal whose reference model allows us to express the most interesting problems in its domain without using tricky solutions. We distinguish several sub-items:

- *Complex clauses.* Most of proposals have limited the provider's offers to be specified by means of simple name-value pairs, whereas customer's demands can be defined by means of conditions on QoS parameters. As an example, we can specify the offer $MTTF = 90$ and the demand $MTTF \geq 90$, but not an offer as $90 \leq MTTF \leq 120$, nor even a demand as $MTTF / (MTTF + MTTR) \geq 0.975$. It would be desirable that expressions involving multiple parameters were allowed to specify QoS demands and offers. On the other hand, nearly all the proposals rely on several types, from simple datatypes to complex data structures, which are used to build the QoS clauses.
- *Uncertainty clauses.* There are reasons to allow a degree of uncertainty when defining a QoS clause. The inclusion of statisticals and ranges in these clauses are examples of such uncertainty. As an example, an offer as $90 \leq MTTF \leq 120$ specifies the range of QoS values it is guaranteeing. This simple expression is not allowed in lots of proposals because they only allow offers to be specified by means of name-value pairs.
- *Symmetry.* There are proposals that allow both demands and offers to be expressed the same, but not all. As an example, it is a typical solution that offers have to be defined by means of name-value pairs, whereas demands can be defined in a less restrictive way, by using simple conditions on QoS parameters. Those proposals that do not allow demands and offers to be expressed the same are said to own an asymmetric QoS specification, whereas those proposals that allow it are said to own a symmetric one.
- *Bilaterality.* There are proposals that also allow offers to specify requirements with regard to the customers which submit demands in order to accept an agreement for usage. As an example, if a web service hosted in the US offered several 128-bit cryptography functions, then it could not serve those demands coming from outside the US, due to current US exportation laws.
- *Assessment criteria.* There are proposals that allow stakeholders to define their criteria on QoS parameters, just as described in section 3.1.

The Reference Model	Static View: The Lexicon					Dynamic View: The Process Model
	Stakeholders	Quality-of-Service Documents	Catalogues, Parameters & Measures			
			Data Model	Customer's	Provider's	
IBM's WSME MME	Providers Costumers	Advertisements Queries Agreements	Data Dictionary: pre-def. basic types sequences records	Name-Value Pair Properties Static/Dynamic Binding Scripts for Rule-based Reqs.		Advertisement/Submission Query/Submission Matchmaking Selecting Providers' Offers
HP's MME Service	Advertisers Requestors	Service Offers & Requests	DAML+OIL Ontology: datatypes and types subsumption	Composition Single-Parameter Constraints on Parameters of Service (expandable)		Advertising Querying Browsing
UDDIe	Providers Consumers	Publishing Inquiry	<i>Blue Pages</i>	Single-Par. Conditions on Properties (Qualifiers)	Name-Value Pair Properties	Publishing Search and Discovery
QRL	Providers Costumers	Demands Offers Agreements	Catalogues: pre-def. basic types catalogue extension basic and derived p.	Composition Multiple-Parameter Constraints on Parameters of Service		Creating Catalogues Offers Submission Demands Submission Matchmaking

Figure 4. Comparison of WSP related proposals with regard to the reference model.

- *Formality.* The criteria to evaluate this feature are the following:

- *Formal description of semantics & checking of properties.* The proposal allows a complete and rigorous description of constituent elements. Thus, checking of consistency, conformance, and optimality are made possible.
- *Usability.* The proposed QoS specification does not make difficult (in excess) the tasks of costumers and providers to specify demands and offers.

- *Viability.* The criteria to evaluate this feature are the following:

- *Availability of a prototype.* There are a lot of features to compare the proposals, but perhaps one of most interesting is the implementation of a run-time system.
- *Reusability.* The different elements (as an example, the catalogues) that have been already defined can be reused.
- *Efficiency.* The involved processes in the run-time framework owns an usage of available computational resources within reason.

4 A Comparative Study

In this section, we go through some proposals according to the comparison framework for WSP we have just defined in section 3. As far as we know, there is no survey on WSP, so that we think it would be an interesting matter to make here a comparative study, reviewing the way these proposals currently own the target features and their implementation aspects.

Figure 4 is devoted to compare them, providing an overall summary at first sight. Figures 5, 6, 7, and 8 show the way the example about a video web portal in figure 1 can be specified by using IBM's WSME [4], HP's MME [6], UDDIe [11], and QRL-based WSP [7, 8, 10], respectively.

The next table shows a comparison between proposals regarding with the target features.

	Expressiveness				Formality	Viability
	Complex	Uncertainty	Symmetry	Bilaterality		
IBM's WSME MME	~	~		V	~	V
HP's MME Service	V	~	V	~	V	V
UDDIe					~	V
QRL	V	V	V	V	V	V

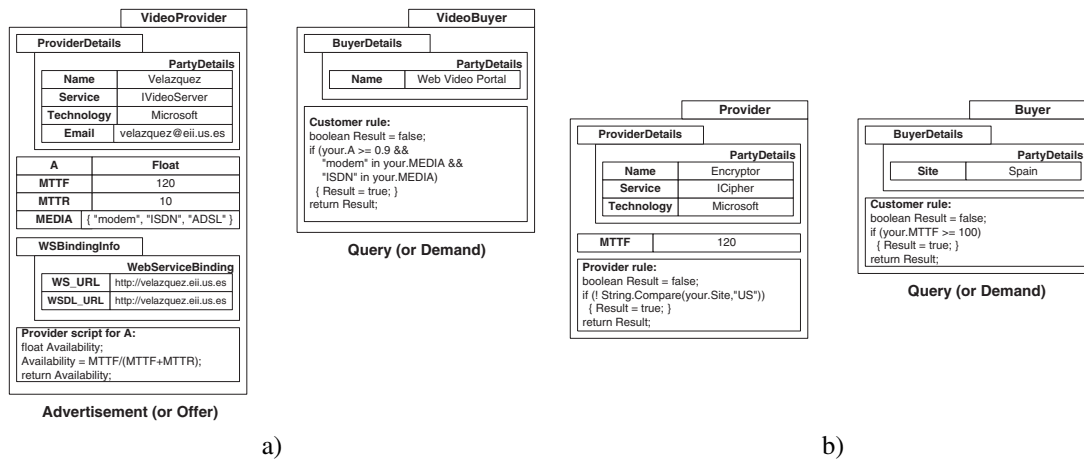
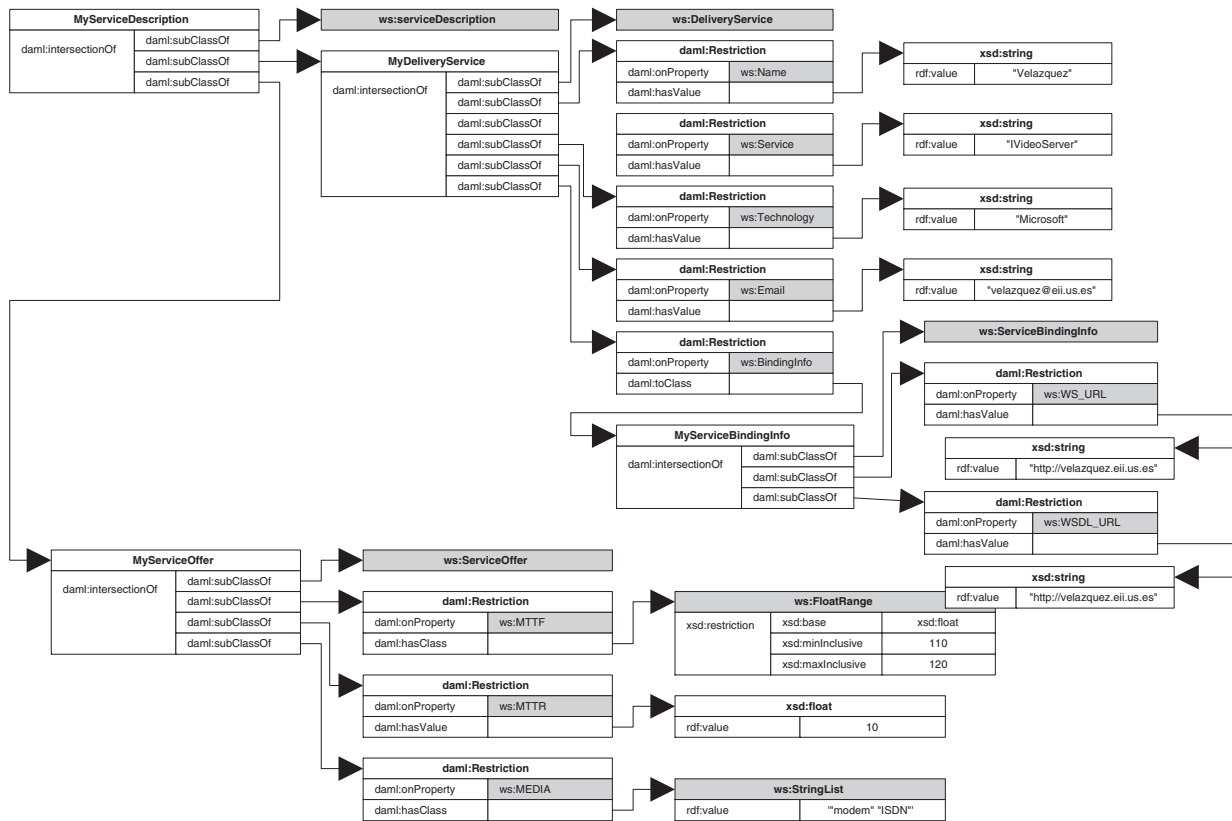


Figure 5. Advertisements and queries in IBM's WSME.

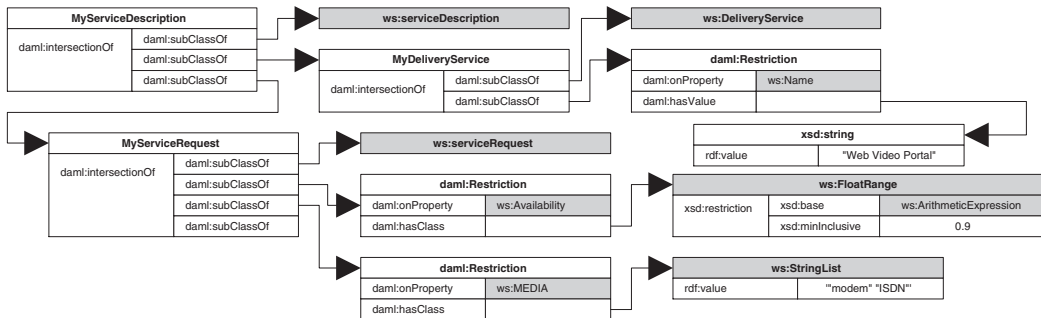
4.1 Expressiveness

- *Complex clauses.* We consider a proposal owns this feature if it allows the composition of multiple-parameter expressions with little or none adaptation when specifying QoS. In this way:
 - UDDIe does not own this feature, because QoS parameters in offers are only expressed by means of name-value pairs, where demands are specified by means of simple conditions on such parameters, which are named qualifiers, as it can be seen in figure 7. On the other hand, the only allowed datatypes are numericals and strings.
 - IBM's WSME does not own this feature in offer's side, as it can be seen in figure 5. QoS offers are only described by means of name-value pairs, but demands contain conditions which are specified by means of a scripting language, so we can specify complex clauses by programming them. These scripts are given by means of programmatic rules which are activated whenever a demand and an offer are computed for checking the conformance.
 - HP's MME does own this feature. It is based on DAML+OIL semantic web ontology: both demands and offers can be expressed by means of single multi-value parameter restrictions by using the XML xsd:restriction derived subtyping, just as it is shown in figure 6. In addition, provided that new ontologies are defined, this approach will become one of most expressive approaches. As an example, if a mathematical ontology of arithmetics were defined, then lots of multiple-parameter expressions could be also specified.

- QRL allows QoS specifications in demands and offers to be expressed by means of complex multi-value parameter mathematical constraints, so that non-linear and other complex expressions are made possible, as it is shown in figure 8.
- *Uncertain clauses.* We consider a proposal owns this feature if specifications of ranges in offer and statisticals are made possible with little or none adaptation. Note both offers in figure 1 include the same range $110 \leq MTTF \leq 120$ expressing a guarantee on a QoS parameter. However, not all the proposals are able to specify it:
 - Both UDDI and IBM's WSME do not allow uncertain clauses because their specification of offers is based on name-value pairs. On the other hand, IBM's WSME does allow uncertain clauses in case of specifying demands because conditions are expressed by means of scripting rules so that ranges and statisticals can be directly programmed *ad hoc*, whereas UDDI have a very limited expressiveness based on simple conditions on single QoS parameters which allow some ranges to be specified, but not statisticals.
 - HP's MME allows the use of ranges to define QoS offers of web services, but the use of statisticals would need the definition of new ontologies.
 - QRL allows both ranges of values and statisticals, as it is shown in figure 8.
- *Symmetry.* We consider a proposal owns this feature if both demands and offers can be expressed the same with little or none adaptation. Thus, HP's MME and QRL can be considered as symmetric, but UDDIe and IBM's WSME not.



Service Offer



Service Request (Demand)

Figure 6. Service requests and offers in DAML+OIL-based HP's MME.

```

<save_service generic="2.0" xmlns="urn:uddie-org:api_v2">
<authInfo>velazquez@eii.us.es</authInfo>
<businessService businessKey="" serviceKey="">
<name xml:lang="en">Velazquez</Name>
<bindingTemplates>
<bindingTemplate bindingKey="">
<accessPoint URLType="http">http://velazquez.eii.us.es</accessPoint>
<tModelInstanceDetails>
// Information about IVideoServer
</ModelInstanceDetails>
</bindingTemplate>
</bindingTemplates>
<propertyBag>
<property>
<propertyName>MTTF</propertyName>
<propertyType>number</propertyType>
<propertyValue>120</propertyValue>
</property>
<property>
<propertyName>MTTR</propertyName>
<propertyType>number</propertyType>
<propertyValue>10</propertyValue>
</property>
</propertyBag>
</businessService>
</save_service>

```



A published web service with QoS properties

```

import org.uddi4j.*;
import ...

class ... {
public static void main(String[] args) {
try {
UDDIProxy proxy = new UDDIProxy();
proxy.setInquiryURL("...");
proxy.setPublishURL("...");

AuthToken token = proxy.get_authToken("...", "...");

Name name = new Name("IVideoServer");
Vector names = new Vector();
names.add(name);

Property property1 = new Property("MTTF", "number", 110);
property1.setPropertyFindQualifier.GREATER_THAN);

Property property2 = new Property("MTTR", "number", 10);
property2.setPropertyFindQualifier.EQUAL_TO);

Vector properties = new Vector();
properties.add(property1);
properties.add(property2);

PropertyBag bag = new PropertyBag();
bag.setPropertyVector(properties);

FindQualifier findQualifier1 = new FindQualifier("ExactPropertyMatch");
FindQualifier findQualifier2 = new FindQualifier("ExactPropertyMatch");
FindQualifiers qualifiers = new FindQualifiers();
Vector qualifiersVector = new Vector();
qualifiersVector.add(findQualifier1);
qualifiersVector.add(findQualifier2);
qualifiers.setFindQualifierVector(qualifiersVector);

ServiceList list = proxy.find_service(null, names, null, bag, null, qualifiers);

... // Now the list of services can be processed.
}
}
}

```



Using the Inquiry API Set to search the appropriate web services

Figure 7. Publications and inquiries of web services in UDDI.

```

// A catalog for Reliability-related QoS parameters
catalog Cat-Reliability {
MTTF {
description: "Mean Time to Failure";
domain: real [0, +inf] minute;
statistic: variance;
};
MTTR {
description: "Mean Time To Repair";
domain: real [0, +inf] minute;
statistic: variance;
};
Availability {
description: "Readiness to Use";
relies: TTF, TTR;
stereotypes {
HIGH: MTTF / (MTTF+MTTR)>=0.9;
LOW: MTTR / (MTTF+MTTR)<0.9;
}
}
}

// A catalog for Network-related QoS parameters
catalog Cat-Network {
MEDIA {
description: "Media Support";
domain: set { modem, ISDN, ADSL };
}
}

// Demands for IVideoServer
using Cat-Reliability, Cat-Network;
demands for IVideoServer {
Availability = HIGH;
MEDIA includes {modem, ISDN};
}
assessment {
MTTF (90, { (0,0), (90,0.5), (120,1) });
MTTR (05, { (0,1), (20,0.6), (30,0) });
MODEM {05,
case { } = MEDIA: 0;
case {modem} = MEDIA: 0.1;
case {ISDN} = MEDIA: 0.3;
case {ISDN,modem} = MEDIA: 0.5;
case {ADSL} = MEDIA: 0.9;
case MEDIA includes-not-equal {ADSL}: 1;
}
}
}
b) A demand.

// Service that uses Windows Media Server
using Cat-Reliability, Cat-Network;
offer for IVideoServer {
MTTF >= 110 and MTTF <= 120;
MTTR = 10;
MEDIA = {ADSL, ISDN, modem};
}

// Service that uses Apple QuickTime Server
using Cat-Reliability, Cat-Network;
offer for IVideoServer {
MTTF >= 110 and MTTF <= 120;
MTTR = 10;
MEDIA = {ISDN, modem};
}

```

a) Two catalogs of quality-of-service parameters.

c) Two offers.

Figure 8. Specification of catalogues, demands and offers in QRL.

In this cases, their adaptations are not easy because both platforms are based on programming languages whose operators are able to check if any value meets a condition, but they do not have complex operators such as computing the inclusion of multi-dimensional spaces or similars, which are needed if offers are also specified with complex expressions.

- *Bilaterality*. We consider a proposal owns this feature if offers can impose some restriction on the demands which intend to use them. IBM's WSME is the only proposal which owns this feature. In figure 5.b, an offer is imposing a condition so that any demand outside the US is ignored. On the other hand, QRL-based WSP has been recently adapted to own this feature, so that the conformance is checked in both directions: demands onto offers, and offers onto demands, too.
- *Assessment criteria*. We consider a proposal owns this feature if specification of assessment criteria is possible with little or none adaptation. If new ontologies were created, HP's MME could be adapted to own this feature. QRL allows assessment criteria to be defined by means of utility functions, as shown in figure 8.

4.2 Formality

As a scripting language is used for QoS specifications in IBM's WSME, it is difficult to extract formal descriptions from programming code, so checking of properties based on them is a hard task. On the other hand, this proposal is easy to use, because of familiarity the expertised people have of programming languages. The same, UDDIe relies on functionality provided by publishing and inquiry API to deal with the underlying UDDI model, but there is no formalisation with regard to QoS. On the contrary, as HP's MME is based on ontologies from the semantic web, checking of properties is made possible. Unfortunately, it is a research just starting walking, and it is difficult to use. QRL is based on constraints, thus the checking of QoS specifications by means of constraint satisfaction problems.

4.3 Viability

We can easily get a prototype of UDDIe. The same, IBM's WSME is an application supplied as part of the IBM's WSTK, and a prototype of a QRL-based WSP runtime framework can be also tested. On the contrary, HP's MME lacks of implementation of Description Logics (DL) reasoners which are needed in the matchmaking. On the other hand, UDDIe and IBM's WSME are based on imperative programming languages, so their efficiency is higher than HP's MME and QRL whose search algorithms based on formal descriptions have very high computational costs.

5 Conclusions and Future Work

A classification and comparison framework for WSP platforms has been presented in this paper. As well, we have used it to present a brief comparison of some quality-aware approaches for WSP. The research in this area is just starting walking, and we think there will be interesting advances in the near future. As claimed by [12], if we want to have a competitive technology based on web services, quality is one of challenges to be solved before 2010.

Acknowledgment

This work was funded by the Spanish Central Government under grants TIC2000-1106-C02-01 and TIC2003-02737-C02-01, and the Castilla-La Mancha Local Administration under grant PCB-02-001.

References

- [1] UDDI technical white paper, 2000. <http://uddi.org>.
- [2] D. Coalition. DAML+S: Semantic markup for web services. In *Proc. of the Int'l Semantic Web Working Symposium SWWS01*, 2001.
- [3] R. Corchuelo, A. Ruiz-Cortés, J. Mühlbacher, and J. García-Consuegra. Object-oriented business solutions. In *Ch. 18 of ECOOP'01 Workshop Reader, Springer LNCS 2323*, 2001.
- [4] S. Field and Y. Hoffner. Web services and matchmaking. *Int'l. J. Networking and Virtual Organisations*, 2003.
- [5] A. Finkelstein and G. Spanoudakis. Software package requirements and procurement. In *8th Int'l Workshop on Software Specification and Design (IWSSD)*. IEEE Press, 1996.
- [6] J. González-Castillo, D. Trastour, and C. Bartolini. Description logics for matchmaking of services. Technical Report HPL-2001-265, Hewlett-Packard, 2001.
- [7] O. Martín-Díaz, A. Ruiz-Cortés, D. Benavides, A. Durán, and M. Toro. A quality-aware approach to web services procurement. In *4th Intl. VLDB Workshop Technologies for E-Services, Springer LNCS 2819*, Berlin, Germany, 2003.
- [8] O. Martín-Díaz, A. Ruiz-Cortés, A. Durán, D. Benavides, and M. Toro. Automating the procurement of web services. In *1st Intl. Conference on Service-Oriented Computing, Springer LNCS 2910*, Trento, Italy, 2003.
- [9] OASIS and UN/CEFAT. Electronic business using XML (ebXML), 1999. <http://www.ebxml.org/>.
- [10] A. Ruiz-Cortés, R. Corchuelo, and A. Durán. An automated approach to quality-aware web applications. In *Enterprise Information Systems IV*. Kluwer Academic Pub., 2003.
- [11] A. ShaikhAli, O. Rana, R. Al-Ali, and D. Walker. UDDIe: An extended registry for web services. In *Int'l SAINT Workshop on Service Oriented Computing*. IEEE Press, 2003.
- [12] G. Weikum. The Web in 2010: Challenges and opportunities for database research. *Springer LNCS 2000*, 2001.
- [13] L. Zhang, H. Li, H. Chang, and T. Chao. XML-based advanced UDDI search mechanism for B2B integration. In *4th Int'l Workshop on Advanced Issues of E-Commerce and Web-based Information Systems*. IEEE Press, 2002.