A Quality-aware Approach to Web Services Procurement ⋆

Octavio Martín-Díaz, Antonio Ruiz-Cortés,
David Benavides, Amador Durán, Miguel Toro

Dpto. de Lenguajes y Sistemas Informáticos
E.T.S. de Ingeniería Informática, Universidad de Sevilla
41012 Sevilla, España - Spain
Phone: +34 95 455 3871 Fax: +34 95 455 7139
{octavio,aruiz}@lsi.us.es,
benavides@us.es,{amador,mtoro}@lsi.us.es

Abstract Web services bring programmers a new way to develop advanced applications able to integrate any group of services on the Internet into a single solution. Web services procurement (WSP) is focussed on the acquisition of web services, including some complex tasks such as the specification of demands, the search for available offers, and the best choice selection. Although the technology to support them already exists, there are only a few approaches wherein quality-of-service in demands and offers is taken into account, in addition to functionality. In this paper, we present some implementation issues on a quality-aware approach to WSP, whose solution is mainly based on using mathematical constraints to define quality-of-service in demands and offers.

Keywords: Software Procurement, Web Services, Quality-of-Service.

1 Introduction

The incredible successfulness of the Internet world has paved the way for a sub-industry devoted to developing and consuming web services, which is being considered as the core of the next-generation Internet. Web services bring programmers a new way to develop advanced applications which can integrate any group of services on the Internet into a single solution. It may involve, possibly, the use of web services provided by different organisations, cooperating in complex collaborations. Thus, there is a need of agreements in order to establish the obligations to both sides, i.e. customers which use web services, and providers which supply them.

Moreover, if we want to have a competitive technology based on web services, then one of challenges to be solved is quality-of-service owned by them [30]. Therefore, these agreements should include not only functional, but also quality-of-service obligations. All available web services may not be appropriate, only those fulfilling the demands on quality-of-service. Federated systems [2], cross-organisational workflows [13, 15] and multi-organisational web-based systems [5, 23] are several examples of this kind of systems.

⋆ Supported by the Spanish Interministerial Commission on Science and the Spanish Ministry of Science and Technology under grants TIC2000-1106-C02-01, FIT-150100-2001-78, and PCB-02-001.

In this context, software procurement [9, 10] becomes web services procurement (WSP): an activity focussed on the acquisition of web services which are required by a web-service-based system. In general, typical tasks involved in WSP are: i) the specification of demands, ii) the search for available offers, and iii) the best choice selection. Thus, WSP is a critical activity for current developers because the great number of available offers and quality-of-service parameters which can be involved in these tasks. Nowadays, there is a great effort from industry in supporting WSP-related tasks. However, most of approaches are based on functionality, and there are only a few which allow a limited expressiveness when specifying quality-of-service in demands and offers. Usually, some of their drawbacks are: i) specification of quality-of-service is only based on single quality-of-service parameters involved in simple expressions, or ii) specification of quality-of-service in offers is based on pairs parameter/value, or iii) unavailability of a solver able to process (some of) expressions, or iv) no optimization of search processes, or others.

In this paper, we present some implementation issues on a quality-aware approach for WSP. The proposed solution is based on using mathematical constraints to specify, in a declarative way, quality-of-service in demands and offers of web services. This allows a great deal of expressiveness and makes possible the implementation of WSP-related tasks by means of solving constraint satisfaction problems (CSP). Currently, we are working on a prototype which makes use of available technology: i) XML is used to specify quality-of-service in demands and offers, following the XML schema corresponding to QRL [23, 27], the language we have proposed for specifying quality requirements, ii) XSLT is used to transform XML documents into constraint satisfaction problems, and iii) ILOG's OPL Studio is the constraint solver.

The rest of this paper is structured as follows. First, Section 2 introduces the notion of web service procurement with a case of study. Next, Section 3 shows the use of mathematical constraints to specify quality-of-service, and implementation of WSP-related tasks by means of CSP. Then, Section 4 describes some implementation issues of the prototype, remarking on web services we have built so far. Finally, Section 5 reviews the related work, giving a brief comparative of existing approaches, and Section 6 will summarise the presented work and the immediate future.

2 WSP in a Nutshell

As introduced above, WSP is focussed on the acquisition of web services which are required by a web-service-based system. As an example, consider that someone is interested in setting up a web portal devoted to video broadcasting, so that it offers a catalogue of videos, and the same functionality just as a domestic video player. In order to achieve such goal, the system should include a service for streaming video through the Internet, a service for managing catalogues and keeping them up-to-date, and a service for managing virtual shops. Thus, the web portal becomes a composed service that integrates web services, possibly provided by other organisations. As well, the agreements for using these services should be established having their quality-of-service taken into account, including both the original demand and the selected offer.

Figure 1 shows a fragment of the components view of the multi-organisational web-based system which corresponds to this portal. The `IVideoServer` interface abstracts those operations that a component delivering video on demand should implement in order to be incorporated into the system. There are several notes attached to every architectural fragment. They are written in QRL (Quality Requirements Language) [23, 27], a language we have designed for specifying quality-of-service. A first note is associated with the `IVideoServer` interface: it states the demand for quality-of-service to be guaranteed by a web service which implements this interface so that it can be eventually used by the system. Remaining notes are associated to web services: they state the offer of quality-of-service which their providers guarantee when delivering video on demand.

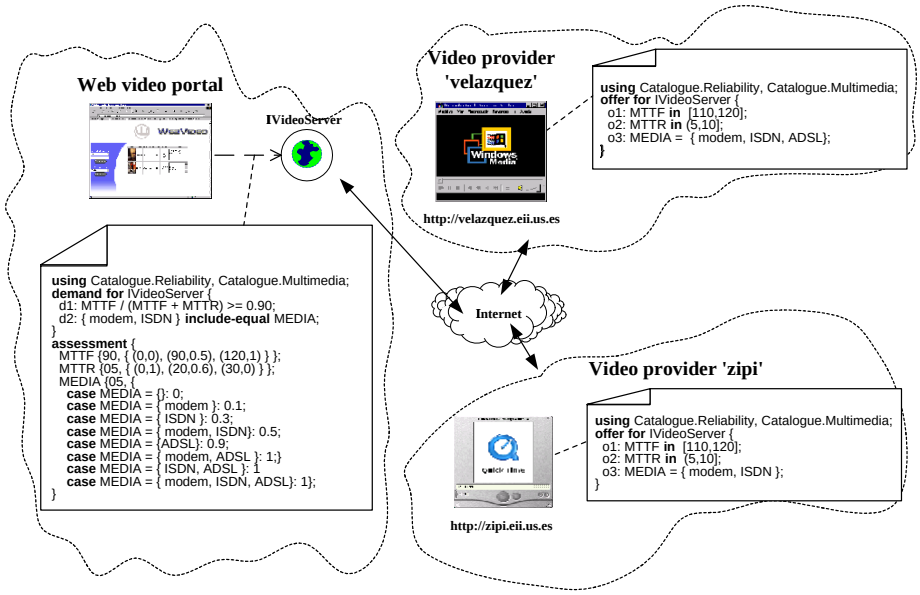

Figure 1. A components view of a video web portal.

In this case, the involved quality-of-service parameters are `Mean Time To Failure` (MTTF), `Mean Time To Repair` (MTTR), and `Media Support` (MEDIA). Whenever a new offer or demand is submitted to the system, it needs to be checked for consistency, that is to say, whether or not it contains inner contradictions. If we read both demand and offers in Figure 1, we will verify that they all are consistent. On the other hand, whenever new consistent demands on web services are submitted to the system, it needs to search those available offers in conformance with them. An offer is conformant to a demand if all quality-of-service values guaranteed by the offer fulfill the demand. If we read all the offers in Figure 1, we will verify they all are conformant to the demand which is needed to be subcontracted by the system.

Moreover, as different offers can be conformant to a given demand the best offer should be selected. This selection is based on assessment criteria which customers can attach to demands, containing their assessments regarding with values that quality-of-service parameters can take, together with their preferences among them. In this way, a web-service-based application is said to be optimum when it is composed of a set of web services so that their offers maximise the assessment criteria from a customer's point of view. These systems are also very flexible, because web services can be exchanged without unnecessary stops whenever new demands and offers are submitted, and/or better offers are found. According to assessment criteria included in Figure 1, provider `velazquez` is the best offer: both offers `velazquez` and `zipi` own the same values for `MTTF` and `MTTR`, but the first offers a better media support because it includes `ADSL`, which has a better assessment from the customer's viewpoint.

3 Supporting WSP with Constraint Programming

The core of the solution relies on the specification of quality-of-service in demands and offers by means of mathematical constraints. In this way, it is achieved a greater deal of expressiveness, and subsequent checking of properties, such as consistency and conformance, and computing of utility assessment of offers, can be implemented as a constraint satisfaction problem (CSP) [11, 14, 20]. A CSP is composed of a set of variables, each of which is given a domain which specifies the values it can take, and a set of constraints on values they can take in a concrete context. A CSP is said to be satisfiable whenever there exists (at least) one solution, i.e. all the variables can be given a value so that the constraints are fulfilled as a whole. In general, CSP-based modelling is quite simple and intuitive (in most cases) in the context of problems which we are dealing with.

3.1 Consistency and conformance

The consistency of every demand or offer which is submitted to the system needs to be checked, i.e., to compute whether or not the corresponding CSP, composed of all the mathematical constraints which are included in it, is satisfiable. In Figure 1, we can verify all CSP corresponding to demand and offers are satisfiable, so they all are consistent.

On the other hand, the conformance of an offer (by a provider) to a demand (from a customer) also needs to be checked, i.e., we are interested in determining whether or not each and every solution to the CSP corresponding to the offer is also a solution to the CSP corresponding to the demand. Formally [20]:

$$conformance(\omega, \delta) \Leftrightarrow sat(c_\omega \wedge \neg\, c_\delta) = false$$

where ω is the offer and c_ω its corresponding CSP, δ is the demand and c_δ its corresponding CSP, and sat is a function that we identify with the constraint solver we are using. It can be applied on a constraint c so that it returns one of the following results: $true$, if c is satisfiable, $false$ if not, and $\perp$ if the constraint solver cannot determine whether c is satisfiable or not. In Figure 1, we can verify both offers are conformant to the demand.

3.2 Optimality

As described above, it is possible to have several offers which are conformant to a demand for a web service. Therefore, a selection mechanism to choice the optimum service is needed. This selection is carried out according to the assessment criteria the customer includes in his or her demand. These criteria are given by means of utility functions [8, 18, 22] which, in general, have the signature $\mathcal{U} : \pi \rightarrow [0, 1]$, where π is the measuring domain of a quality-of-service parameter. In this way, the customer can define his/her assessment criteria regarding with a parameter by means of an utility function which assigns an utility assessment (ranging from 0 to 1) to every value it can take, so the greater the assessment, the better the consideration of the customer. Therefore, utility functions allow the establishment of an objective criteria, given by customers, in order to select the offers which better fulfill the demands. Figure 2 shows several utility functions corresponding to the demand in Figure 1.

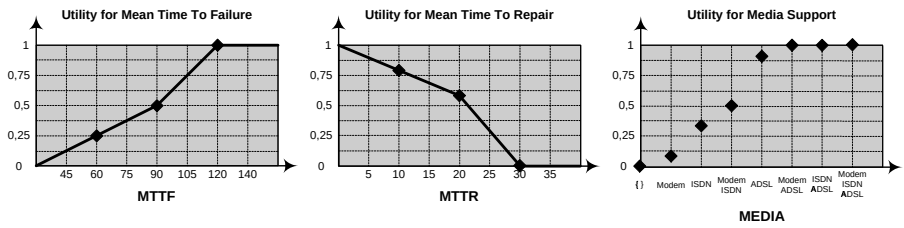

Figure 2. Utility functions for $MTTF$, $MTTR$, and $MEDIA$.

Moreover, we are not usually interested in utility functions with regard to lonely quality-of-service parameters, but on maximising the global assessment of offers in order to select the best one, being these offers conformant to the demand. Nevertheless, we can not compute the maximum utility assessment of offers when comparing them, because offers are guaranteeing the complete range, not only a particular quality-of-service value. Therefore, we compare the minimum utility assessments of offers. Formally:

$$\omega_S = \omega \in \Omega_\delta \cdot \ \forall \omega_i \in \Omega_\delta - \{\omega\} \ \mathcal{U}^\delta(\omega) \geq \mathcal{U}^\delta(\omega_i)$$

where ω and ω_i stand for offers in the set Ω_δ of conformant offers to the demand δ, and ω_S represents the selected offer. The utility function $\mathcal{U}^\delta(\omega)$ of an offer ω according to assessment criteria in demand δ is expressed as an optimization problem:

$$\mathcal{U}^\delta(\omega) = \underset{subject\ to\ c_\omega}{min} \sum_{\pi \in c_\omega} w_\pi^\delta U^\delta(\pi)$$

where π represents a quality-of-service parameter which is involved in the offer's CSP c_ω, and $U^\delta(\pi)$ its utility function, and w_π^δ its assigned weight, according to assessment criteria in demand δ. On the other hand, weights are needed to express that a quality-of-service parameter is preferred to another.

In Figure 1, since both offers are conformant to the demand, we will have to compute their utility assessments in order to compare them. In this way, both offers own $U(MTTF = 110) = 0.83$ and $U(MTTR = 10) = 0.8$, the `velazquez` offer owns $U(MEDIA) = 1$, and the `zipi` offer owns $U(MEDIA) = 0.5$. Therefore, utility assessment of `velazquez` is $0.9 * 0.83 + 0.05 * 0.04 + 0.05 * 1 = 0.84$, and utility assessment of `zipi` is $0.9 * 0.83 + 0.05 * 0.04 + 0.05 * 0.5 = 0.815$, so the best offer is `velazquez`.

4 Implementation Issues

4.1 Overview of the prototype's architecture

Currently, we are developing a prototype of the framework for management and execution of multi-organisational web-based systems. A preliminary version of the prototype is available at the web page `http://www.lsi.us.es/~octavio`, which shows some prepared examples using the web services which have been implemented so far. In fact, they will constitute the kernel of a future run-time framework whose architecture has been already defined and published in other works [21, 24, 28]. A components view of its architecture is shown in Figure 3.

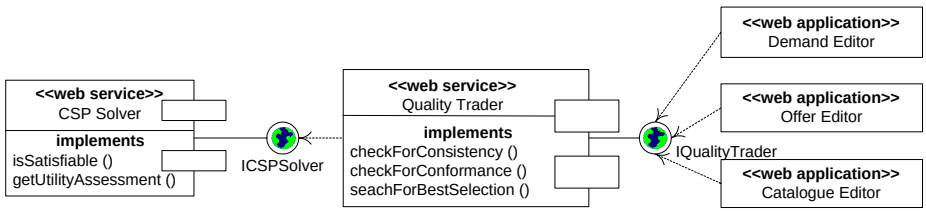

Figure 3. Architecture of the prototype's run-time framework.

One of design decisions we have made is to deploy all components as web applications and web services, so that the framework itself can be properly considered just as another multi-organisational web-based system:

- The `CSP Solver` web service is a wrapper to access the actual component which provides the solver for processing the incoming CSP. It provides the services `IsSatisfiable()` which it returns whether the CSP passed as a parameter is satisfiable or not, and `getUtilityAssessment()` which it computes the utility assessment of the optimization problem passed as a parameter.
- The `Quality Trader` web service provides the `checkForConsistency()`, `checkForConformance()`, and `searchForBestSelection()` services. All these functions have a similar operation: they take the involved demands and offers written in XML as parameters, the they invoke the appropriate XSLT transformations in order to generate automatically the corresponding CSP, which is processed by the CSP solver to get the result, which is finally returned.

4.2 XML schemas for QRL

We have decided to adopt XML as the language for exchanging QRL-based quality-of-service specifications, so we have defined several XML schemas corresponding to abstract semantics of QRL language. We have defined up to 291 elements in all, for the time being, so that any QRL-based document can be written in XML with no loss of original expressiveness.

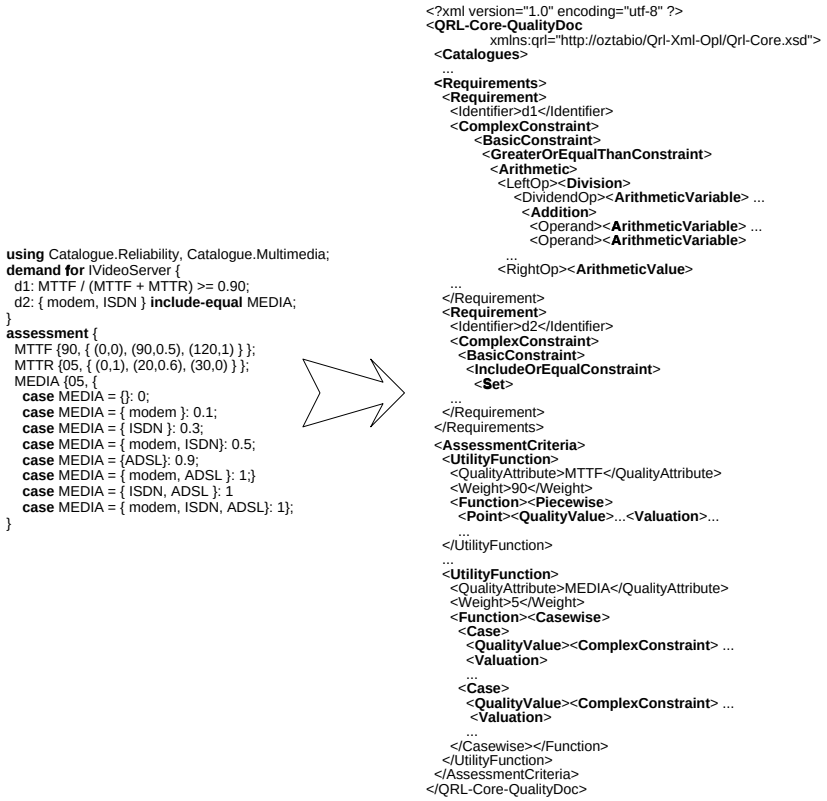

Figure 4. An example of a QRL-based demand written in XML.

As an example, Figure 4 shows partially a demand written in XML. The `QRL-Core-QualityDoc` XML-element is the root of specification, which includes names of catalogues which are being used and the requirements. Each requirement is given an identifier and a constraint. In turn, a constraint is expressed with a `ComplexConstraint` XML-element, which is the root of all mathematical constraints available in QRL, including logic, comparison, assignment, and arithmetic operators. Finally, the `AssessmentCriteria` XML-element is the root of specification of assessment criteria. Each inner `UtilityFunction` XML-element contains the name of a quality-of-service parameter and its weight of preference, and its proper specification.

4.3 CSP solver: ILOG's OPL Studio

The CSP solver which we have used is ILOG's OPL Studio [16]. Its language OPL (OPtimization Language) is easy to use, so we have considered it for solving the CSP corresponding to WSP-related tasks, such as checking of consistency and conformance, and best choice selection. An OPL model contains a CSP, and it is basically composed of a section for declaring variables, a maximise/minimise section to include an optimization function, and a section which includes the set of constraints.

```
range TYPE_MTTF 0..9999;
var TYPE_MTTF MTTF;

range TYPE_MTTR 0..9999;
var TYPE_MTTR MTTR;

enum TYPE_MEDIA
        {MEDIA_modem,MEDIA_ISDN,MEDIA_ADSL};
var int MEDIA[TYPE_MEDIA] in 0..1;

// IF SATISFIABLE, DEMAND IS CONSISTENT

solve {
  ( (MTTF * 100) / (MTTF + MTTR) >= 90 )
 &
  ( MEDIA[MEDIA_modem]=1 & MEDIA[MEDIA_ISDN]=1);
};
```

a) OPL model for consistency.

```
enum TYPE_MEDIA
        { MEDIA_modem,MEDIA_ISDN,MEDIA_ADSL};
var int MEDIA[TYPE_MEDIA] in 0..1;

range TYPE_TTF 0..120;
var TYPE_TTF TTF_MEAN;

range TYPE_TTR 0..30;
var TYPE_TTR TTR_MEAN;

// IF NO SATISFIABLE, THE OFFER IS CONFORMANT
// TO DEMAND

solve {
   ( // VELAZQUEZ'S IVIDEOSERVER OFFER
       (110 <= TTF_MEAN <= 120)
     & (5 < TTR_MEAN <= 10)
     & ( MEDIA[MEDIA_modem] = 1
         & MEDIA[MEDIA_ISDN] = 1
           & MEDIA[MEDIA_ADSL] = 1) )
   &
   not( // IVIDEOSERVER DEMAND
          ( (TTF_MEAN * 100) / (TTF_MEAN+TTR_MEAN) >= 90 )
        & ( MEDIA[MEDIA_ISDN] = 1
            & MEDIA[MEDIA_modem] = 1) );
};
```

b) OPL model for conformance.

```
enum TYPE_MEDIA { MEDIA_modem,MEDIA_ISDN,MEDIA_ADSL};
var int MEDIA[TYPE_MEDIA] in 0..1;
var int UTILITY_MEDIA_VALUE in 0..111;

range TYPE_MTTF 0..120;
var TYPE_MTTF MTTF;

range TYPE_MTTR 0..30;
var TYPE_MTTR MTTR;

minimize
  0.90 * piecewise{0.55->90;1.67->120;0} MTTF+
  0.05 * (100 - piecewise{2->20;6->30;0} MTTR) +
  0.05 * piecewise{1->1;3.22->10;20->11;0.45->100;10->101;0}
                UTILITY_MEDIA_VALUE

subject to {
  UTILITY_MEDIA_VALUE
     = sum(AUX_MEDIA in TYPE_MEDIA)
          MEDIA[AUX_MEDIA] * pow(10,ord(AUX_MEDIA));

  // VELAZQUEZ'S IVIDEOSERVER OFFER
  (  110 <= MTTF <= 120
   & 5 < MTTR <= 10
   & MEDIA[MEDIA_modem] = 1
     & MEDIA[MEDIA_ISDN] = 1 & MEDIA[MEDIA_ADSL] = 1 );
};
```

c) OPL model for computing the utility assessment.

Figure 5. OPL models for consistency, conformance, and utility assessment.

Nevertheless, although the OPL solver has demonstrated to be a good CSP solver, it presents some drawbacks, but none of them have demonstrated to be definitely unavoidable. These weaknesses have slightly restricted the original expressiveness of our solution, and made some implementation aspects harder as well. Figure 5 shows several examples of OPL models, referred to examples of Figure 1, according to definitions in Section 3: a) consistency of a demand, b) conformance of `velazquez`'s offer to a demand, and c) computing the utility assessment of `velazquez`'s offer with regard to a demand.

4.4 XSLT transformations to OPL models

XSLT (eXtensible Stylesheet Language Transformations) describes rules by means of templates for transforming a XML source into any arbitrary result. These transformations are not trivial at all, since XML schemas of QRL and structure of OPL models are very different:

```
<?xml version="1.0" encoding="utf-8" ?>
<Qrl-Conformance
      xmlns:qrl="http://oztabio/Qrl-Xml-Opl/Qrl-Core.xsd">
  <Catalogues></Catalogues>
  <Attributes></Attributes>
  <TheOffer></TheOffer>
  <TheDemand></TheDemand>
</Qrl-Conformance>
```

Figure 6. A XML template devoted to XSLT transformations.

Figure 6 shows the template which is needed to invoke our XSLT transformations. In general, a new XML document is created in order to get together the involved demand, offer and used catalogues. As we have used the ILOG's OPL Studio tool as the CSP solver, we have defined several XSLT transformations to get the OPL model which contains the appropriate CSP for checking the consistency and conformance, and computing the utility assessment.

5 Related Work

Several tools provide all necessary elements to implement, search and invoke web services using the current technology. However, these approaches have been focussed on functionality to be provided by web services, but not on quality-of-service. On the other hand, there are only a few proposals which allow a limited expressiveness to specify quality-of-service offered by/demanded from web services, such as *DARPA Agent Markup Language plus Services* (DAML+S) [3], *Web Services Outsourcing Manager* (WSOM) [6], and *UDDI extension* (UDDIe) [29]. In general, these proposals do not allow a symmetric way to specify quality-of-service, because demands are usually specified with a greater deal of expressiveness than offers, and in most of cases specification of quality-of-service in offers is only based on pairs parameter/value, but not any more complex expression. Figure 7 shows a comparative among current quality-aware approaches to WSP.

Currently, there are two approaches (as far as we know) which allow a greater deal of expressiveness when specifying quality-of-service in demands and offers, such as HP's *Matchmaking Engine* (MME) [12], which is based on the DAML semantic web language [4], and *Web Services Matchmaking Engine/Web Service Level Agreement* (WSME/WSLA) [15, 17, 19], which is an enhancement of the CORBA/ODP trader service and it has been integrated into IBM's *Web Services ToolKit* (WSTK) [7]. The former owns a great deal of expressiveness due to the use of the semantic web language

The Reference Model	Static View: The Lexicon					Dynamic View: The Process Model
	Stakeholders	Quality-of-Service Documents	Catalogues, Parameters & Measures			
			Data Structuring	Customer's	Provider's	
IBM's WSME MME	Providers Costumers	**A**dvertisements Queries **A**greements	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	*Blue Pages*	Single-Par. Constraints on Properties (Qualifiers)	Name-Value Pair Properties	Publishing Search and Discovery
QRL	Providers Costumers	Demands Offers **A**greements	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 7. Comparative of quality-aware approaches to WSP.

DAML, but there is currently no Description Logic's solver able to process some of most complex expressions which can be specified. The latter also owns a great deal of expressiveness: its specification is based on using rules written in a scripting language, wherein offers and demands are absolutely symmetric from both viewpoints: the demand can impose conditions on the offer, and viceversa. However, their results are the lists of all conformant offers to a demand but there is no optimization of searches.

6 Conclusions and Future Work

In this paper, we have presented some of implementation issues of our quality-aware approach to WSP. The proposed solution is based on using mathematical constraints in order to specify quality-of-service in demands and offers, so we have achieved a lot of interesting properties. First, it owns a great deal of expressiveness, including multiple parameters and non-linear expressions involving quality-of-service parameters. As the same expressiveness is allowed to specify quality-of-service in demands and offers, our approach can be said to be symmetric. As well, our approach includes the possibility to express the assessment criteria which is very important to select the best choice according to a demand.

We have developed a prototype of the run-time framework for management and execution of multi-organisational web-based systems. This prototype includes a quality trader web service as the main component, which offers services such as checking for consistency and conformance, and searching for the best choice. Of course, this web service is a cornerstone of the framework's kernel, which will be available in the near future. Among the main characteristics of implementation, we have used the QRL language to specify quality-of-service, and XML to specify QRL-based documents, the definition of XSLT transformations to get the appropriate CSP for carrying out the WSP-related tasks, and the use of a constraint solver as ILOG's OPL Studio. At the

moment of writing this paper, readers who are interested in our proposal can have an overview of our prototype in the web page `http://www.lsi.us.es/~octavio`.

Regarding with future work, we want to point out that our approach can be extended in several ways in order to include new characteristics: temporality clauses in constraints, negotiation clauses to improve the flexibility of the model whenever no solution can be initially found, and importance and soft clauses in order to enlarge the solution space of the search. In fact, definition of temporality and negotiation are currently in study [25, 26], so we are beginning the first phases of improvements of our prototype to include them.

Finally, the integration of our model on the current technology is also one of our pending work. We are aware of the uselessness of our approach if we do not have a working prototype integrated with any of them, such as UDDI or similar. In this way, our quality trader will be a component leveled at the top of a pyramid wherein lower levels would be devoted to functional-aspects of WSP [1]. This stage of development is currently starting, but we hope to have a completely functional prototype in the very near future.

References

1. A. Beugnard, J-M. Jézéquiel, N. Plouzeau, and D. Watkins. Making components contract aware. *IEEE Computer*, pages 38–45, July 1999.
2. P. Bhoj, S. Shingal, and S. Chutani. SLA management in federated environments. *Computer Networks*, 35:5–24, 2001.
3. DAML+S Coalition. DAML+S: Semantic markup for web services. In *Proc. of the Int'l Semantic Web Working Symposium SWWS01*, 2001.
4. Joint US/EU Agent Markup Language Committee. DARPA Agent Markup Language. Technical report, US's DARPA Defense Advance Research Projects Agency and EU's IST Information Society Technologies, 2000. `http://www.daml.org`.
5. R. Corchuelo, A. Ruiz-Cortés, J. Mühlbacher, and J.D. García-Consuegra. Object-oriented business solutions. In *Chapter 18 of ECOOP'2001 Workshop Reader, LNCS 2323*, pages 184–200. Springer-Verlag, 2001.
6. IBM International Business Machines Corporation. Web Services Outsourcing Manager overview (WSOM), 2002. `http://www.ibm.com`.
7. IBM International Business Machines Corporation. Web Services ToolKit (WSTK), 2002. `http://www.ibm.com`.
8. J.J. Dujmovic. A Method for Evaluation and Selection of Complex Hardware and Software Systems. In *Proceedings of the 22nd International Conference for the Resource Management and Performance Evaluation of Enterprise Computing Systems*, volume 1, pages 368–378, 1996.
9. B. Farbey and A. Finkelstein. Software acquisition: a business strategy analysis. In *Proc. of the Requirements Engineering (RE'01)*. IEEE Computer Society Press, 2001.
10. A. Finkelstein and G. Spanoudakis. Software package requirements and procurement. In *Proc. of the 8^{th} Int'l IEEE Workshop on Software Specification and Design (IWSSD'96)*. IEEE Press, 1996.
11. E.C. Freuder and M. Wallace. Science and substance: A challenge to software engineers. *Constraints IEEE Intelligent Systems*, 2000.
12. J. González-Castillo, D. Trastour, and C. Bartolini. Description logics for matchmaking of services. Technical Report HPL-2001-265, Hewlett-Packard, 2001.

13. P. Grefen, K. Aberer, Y. Hoffner, and H. Ludwig. CrossFlow: Cross-organizational workflow management in dynamic virtual enterprises. *International Journal of Computer Systems Science & Engineering*, 15(5):277–290, 2000.
14. P. Hentenryck and V. Saraswat. Strategic directions in constraint programming. *ACM Computing Surveys*, 28(4), December 1996.
15. Y. Hoffner, S. Field, P. Grefen, and H. Ludwig. Contract-driven creation and operation of virtual enterprises. *Computer Networks*, (37):111–136, 2001.
16. ILOG. OPL Studio. `http://www.ilog.fr`.
17. A. Keller and H. Ludwig. The WSLA framework: Specifying and monitoring service level agreements for web services. Technical Report RC22456 (W0205-171), IBM International Business Machines Corporation, 2002.
18. J. Koistinen and A. Seetharaman. Worth–based multi-category quality–of–service negotiation in distributed object infrastructures. In *Proceedings of the Second International Enterprise Distributed Object Computing Workshop (EDOC'98)*, La Jolla, USA, 1998.
19. H. Ludwig, A. Keller, A. Dan, and R.P. King. A service level agreement language for dynamic electronic services. Technical Report RC22316 (W0201-112), IBM International Business Machines Corporation, 2002.
20. K. Marriot and P.J. Stuckey. *Programming with Constraints: An Introduction.* The MIT Press, 1998.
21. O. Martín-Díaz, A. Ruiz-Cortés, R. Corchuelo, and A. Durán. A Management and Execution Environment for Multi-Organisational Web-based Systems. In *ZOCO: Métodos y Herramientas para el Comercio Electrónico*, pages 79–88, San Lorenzo del Escorial, Spain, 2002.
22. L. Olsina, D. Godoy, G. Lafuente, and G. Rossi. Specifying Quality Characteristics and Attributes for Websites. In *Proceedings of the Web Engineering Workshop, in conjunction with 21^{st} International Conference on Software Engineering (ICSE)*, pages 84–93, May 1999.
23. A. Ruiz-Cortés. *A Semi-qualitative Approach to Automated Treatment of Quality Requirements (in Spanish).* PhD thesis, E.T.S. de Ingeniería Informática. Dpto. de Lenguajes y Sistemas Informáticos. Universidad de Sevilla, 2002.
24. A. Ruiz-Cortés, R. Corchuelo, and A. Durán. An automated approach to quality-aware web applications. In *Enterprise Information Systems IV*, pages 237–242. Kluwer Academic Publishers, 2003.
25. A. Ruiz-Cortés, R. Corchuelo, A. Durán, and M. Toro. Enhancing Win–Win requirements negotiation model. In *Applied Requirements Engineering*. Catedral, 2002.
26. A. Ruiz-Cortés, R. Corchuelo, A. Durán, and M. Toro. Automated negotiation of quality requirements. In *VII Jornadas de Ingeniería del Software y Bases de Datos (JISBD'02)*, 2002.
27. A. Ruiz-Cortés, A. Durán, R. Corchuelo, and M. Toro. Specification of Quality Requirements in Multi-Organisational Web-based Systems (in Spanish). In *Sextas Jornadas de Ingeniería del Software y Bases de Datos JISBD'01*, pages 615–629, Almagro, Spain, 2001.
28. A. Ruiz-Cortés, R. Corchuelo, A. Duran, and M. Toro. Automated support for quality requirements in web-services-based systems. In *Proc. of the 8th IEEE Workshop on Future Trends of Distributed Computing Systems (FTDCS'2001)*, Bologna, Italy, 2001. IEEE Press.
29. A. ShaikhAli, R. Al-Ali O. Rana, and D. Walker. UDDIe: An extended registry for web services. In *Proc. of the IEEE Int'l Workshop on Service Oriented Computing: Models, Architectures and Applications at SAINT Conference*. IEEE Press, January 2003.
30. Gerhard Weikum. The Web in 2010: Challenges and opportunities for database research. *Lecture Notes in Computer Science n^o 2000*, 2001.