Flexible Service Offering with Semantic Policies

Marco Comerio[1], Flavio De Paoli[1], Matteo Palmonari[1], and Ioan Toma[2]

[1] University of Milano - Bicocca, viale Sarca 336, 20126 Milano, Italy
{comerio,depaoli,maurino,palmonari}@disco.unimib.it
[2] STI Innsbruck - ICT - Technikerstrasse 21a, 6020 Innsbruck, Austria
ioan.toma@sti2.at

Abstract. One of the challenges in Service Oriented Computing consists in supporting service discovery and composition according to the so-called non functional properties (NFP). In this paper we propose a linguistic framework to describe NFPs of services independently of the grounding technology by means of semantic Web technologies. In particular, the approach is based on an extension of the Web Service Modeling Language (WSML). The linguistic framework supports flexible NFP representations by allowing (i) to reuse existent models and ontologies, (ii) to represent both the definition of single NFPs and clusters of NFPs grouped into policies. A policy collects a set of related and possibly interdependent properties to form a single entity that can be referenced and discovered. As a result, definitions of NFPs can be independent of specific services, and vice versa, so to support flexible associations.

Key words: Non-functional properties, Policy, Semantic Web Services, Web Service Modeling Ontology, service level agreements.

1 Introduction

Service-based interactions to support activities of organizations and end-users require the retrieval, composition and execution of services. One of the major challenges is to support service discovery according to the so-called non functional properties (NFP). NFPs are related to technical issues (e.g., quality of service (QoS) such as performance, availability, security), and also to business, social, legal and economic issues, just to mention a few. It is likely that users will be interested in selecting the desired services among a large set of available services, each one providing the same or similar functionalities.

To achieve the goal, a user can issue a request that can be split into functional and non-functional constraints. From the point of view of service discovery and selection, which is our main perspective, the role of functional constraints is to filter eligible services, while the role of non-functional constraints is to compute *degrees of satisfaction* for that set of services (i.e., NFP constraints could be fully, partially or not satisfied). Therefore, NFPs could be quite relevant to match a service request and a service description. In fact, even if a service matches the requested functionalities, it could be unacceptable in terms of NFPs (e.g., if availability is not sufficient, performance is too poor, cost is too expensive).

A model to support a scenario in which a client wants to select the "best" available service needs to address both provider and client perspectives: providers need to define and publish descriptions of NFPs associated with candidate services; clients need to specify the criteria for service selection. The Web Service Modeling Ontology (WSMO) [15] is a conceptual model for Semantic Web services that, differently e.g. from OWL-S [13], explicitly takes into account clients and providers. In particular, WSMO defines four top-level elements: (1) *ontologies*, as formal explicit specifications of share conceptualizations providing the terminology used to describe services, (2) *goals*, as descriptions of the objectives a client may have when consulting a service, (3) *web services*, as descriptions of functionalities and (4) *mediators*, as means to address the heterogeneity problem that occurs between descriptions at different levels (i.e. data, protocol and process level).

In this paper we present a linguistic framework for the semantic representation of (Web) services' NFPs in WSMO. The framework extends the Web Service Modeling Language (WSML) [6], the language used by WSMO. The aim of semantic descriptions of NFP is to achieve a new level of automation for service related tasks like discovery, selection, composition, negotiation, and monitoring, by supporting automated reasoning about NFPs. Moreover, the framework aims to achieve flexibility by providing users with constructs to accommodate NFP descriptions at different level of details. It supports simple descriptions that deal with individual properties by means of logical expressions; as well as composite descriptions that are defined by the *policy* construct, which collects a set of related and possibly interdependent properties in a single entity that can be referenced and discovered.

A definitions of non-functional properties can be independent of any specific service, and vice versa. The advantage is to be able to publish the same service associated with different NFP policies to address the needs of different clients; and to publish a policy that can be associated with different services to facilitate the discovery process and composition. The separation between services and descriptions opens the opportunity to build predefined, possibly standard, policies that can be reused to facilitate the task of publishing, retrieving and composing services.

The linguistic constructs discussed in this paper have been implemented into semantic Web search engines to support NFP aware WS discovery and selection. In particular, the ranking engine discussed in [17] supports the computation of individual properties; the GLUE2 discovery engine [3] deals with composite properties as described by the Policy-Centered Meta-model (PCM) [14]. PCM provides for an ontology-based model to describe single properties to be composed into *policies* as discussed in this paper.

The paper is organized as follows. Section 2 describes semantic and non-semantic approaches for NFP descriptions. Section 3 shows our approach for modeling NFPs and policies. Examples of flexible NFP publishing are reported in Section 4. Finally, Section 5 draws some conclusions.

2 Related Work

Developing models and languages for non-functional properties has been a challenging problem in many research areas. In software engineering especially NFPs models and languages were developed to support the description of NFP aspects and requirements of software systems. In [16] for example, a language for NFPs called Process^{NFL} is proposed to express correlations and conflicts between NFPs and also compositional aspects and strengths. In [8] the ODRL-S language is proposed to describe economical and legal aspects of a service as license clauses. In [1] the CQML modeling language is proposed for specification of Quality of Service (QoS) (i.e., the particular kind of NFPs related to technological characteristics of a Web services) allowing QoS characteristics to be refined and aggregated. The QML [7] language and model is a domain independent approach to describe QoS providing a refinement mechanism that allows QoS aspects to be defined as refinements of existing ones. [5] provides a formal machine processable description of software and services by means of constraints instead of ontologies. In addition, QRL provides a mechanism to express alternative NFPs for the same service by means of the so-called negotiation clauses. In the Web services area some efforts to describe NFPs resulted in the WS-* specifications. Considerable related work has been done also in the context of Service Level Agreements.

One important limitation of the former approaches is the lack of machine processable descriptions. Formal machine processable descriptions of software and services would enable a higher degree of automation required to solve complex problems. With the emergence of semantic technologies as solutions for the previous limitation, several approaches for semantic NFP descriptions have been proposed. A model that facilitates providers in expressing policies and consumers in expressing preferences on QoS is proposed in [11]. A model to establish a set of rules that are used to represent QoS characteristics of Web services along with the relationships among them is introduced in [18].

Current standards for semantic descriptions of services (e.g., WSMO [15] and OWL-S [13]) cover only marginally the specification of NFPs. OWL-S [13] basically adopts attribute-value descriptions. The OWL-S profile only includes an expandable list of NFPs (e.g., average completion time, classification, quality of service) expressed as service parameters. Some papers propose to fill the gap by complementing OWL-S with models for NFP description. Kritikos and Plexousakis [10] propose to enrich OWL-S with an extensible model for describing QoS. Giallonardo and Zimeo [9] define a model that allow service providers advertising offered QoS, and service consumers specifying QoS requirements associated to an OWL-S profile. Comerio et al. in [4] introduce the concept of NFP Policy, with a meta-model providing structure to NFP descriptions to support service selection. The early versions of WSMO [15] adopted as well an attribute-values approach to describe NFPs. As attributes non-functional property identifier are provided. Values can be any identifier and thus it can be an IRI, a data value, an anonymous identifier or a list of the former. The recom-

mended common set of non-functional properties was mainly based on Dublin Core Metadata Initiative [20]. Other approaches such as

In this paper, we present a comprehensive approach to NFP description compliant with different NFP models instead of a new NFP model. The ideas behind the logical expression-based approach to NFP description, together with a NFP-aware discovery algorithm for WS described with WSMO were introduced in [17]. The concept of NFP Policy was first introduced within the Policy-Centered Meta model (PCM) [4]; the refinement and formalization of PCM is described in [14], while a NFP-aware discovery engine based on such model is presented in [3]. Here, elaborating on these preliminary works, we present a unified approach within the WSMO framework, discussing in particular the main advantages expected from the approach in a future Internet business scenario.

3 Representing Non Functional Properties and Policies

The definition and management of NFPs is a complex task due to the nature of the involved properties. NFPs concern a variety of service aspects, ranging from technical issues to pricing policies to legal aspects [12]. Similar properties may have different names (e.g., in different languages or domains) or the same name may refer to different properties (e.g., in different domains a property may have different implications). Numeric values can be expressed in different units (e.g., price in Euro or in USD), or properties can be purely qualitative (e.g., the usability is "good", trust is "high", software is "open source").

Moreover, sometimes the value of a NFP is not fixed but needs to be calculated at runtime when the context or the request provide additional information (e.g., the bandwidth a service is provided with). Nonetheless, as discussed in [14], it could be helpful for the discovery process to publish some information about the possible values these NFPs might assume (e.g. a range), or about the methods the values are calculated with. Moreover, clusters of NFP offers and requests need to be considered because of technological or business interdependencies among NFPs (e.g. a higher price for the service that guarantee a certain bandwidth).

In more general terms, a web service can provide the same functionality with different non-functional property levels. Several real world services provide evidence of this consideration. For example, the Italian railway company Trenitalia web site[3] offers different fares (e.g, Flexi, Standard, Amica, Blu card rail plus[4]) for booking ticket services. Each fare is characterized by a set of non-functional properties such as *discount*, *advantages* and constraints; some property, such as *allowed trains*, specifies the applicability conditions. Therefore, it is possible to buy a transit pass for a destination under different conditions (e.g., weekly or monthly one-way or return pass for IC/EC trains). Service descriptions should reflect these clustering of related NFPs. Moreover, these descriptions should be

---

[3] http://www.ferroviedellostato.it/ferrovie/util/inglese.jsp
[4] The description of the several fares offered by Trenitalia can be found at http://www.trenitalia.com/en/oraribiglietti/infotariffe.html

independent of the functional description of a service since they are subject to change more frequently. For example, the price may change due to market conditions or special discount campaign may be introduced anytime.

In this paper we propose an approach to handle the heterogeneity of NFP descriptions, support the clustering of NFP offered and requested, and finally decoupling FP and NFP specifications. The last feature allows for a more flexible provision of services, where descriptions of NFPs can be changed modularly over a set of persistent FPs.

Single NFPs and NFP policies can be attached to WSs and goals. The NFP-aware discovery process is based on the evaluation of a matching degree between a goal and a number of WS which takes into account NFPs and policies attached to these elements. Since NFP and policy descriptions are attached to goals in the same way they are attached to WS, in the rest of the paper we will show only the WS-side of NFP and policy descriptions.

3.1 Representation of individual NFPs

The proposed solution is based on semantic technologies and starts from two considerations: providing a unique and shared model of all the possible NFPs is unfeasible; nonetheless, there is the need of a systematic approach to NFP description to support selection activity beside the traditional technological perspective.

A first step towards the definition of a self-contained semantically enabled modeling framework for NFPs and policies is the creation of ontological models for specifying non-functional properties. Providers and clients should be made able to refer to different NFPs by adopting existent ontologies or even defining a set of new ones. As part of our initial work (i.e. [17]) we have developed a set of ontologies to provide a formal model for the most common non-functional properties like availability, security, etc.; other models, e.g. those mentioned in Section 2, can be used as well if represented in WSML.

Once domain ontologies have been defined, they can be used to describe NFPs within WSMO service descriptions through identifiers followed by logical expressions. A simplified model of a WSMO service following this approach is illustrated in Listing 1.1[5]:

Listing 1.1. Service description structure

```
webService
    capability idCapability
        precondition definedBy axiomPrecond
        postcondition definedBy axiomPostcond
        assumption definedBy axiomAssumption
        effect definedBy axiomEffect
    nonFunctionalProperty
        idNFP hasValue valuelistnfp definedBy axiomNFP
```

---

[5] Please notice that in WSMO the default cardinality for each statement is zero-to-many, which holds for the listings presented along the paper.

According to this approach one or more NFP can be attached to service descriptions in a similar way as capabilities are attached. The WSMO model has been extended with the class nonFunctionalProperty, characterized as illustrated in Listing 1.2.

Listing 1.2. Non-functional property class

```
Class nonFunctionalProperty
    hasAnnotations type annotations
    hasDefinition type logicalExpression
```

In the NFP specification, ontological references are first introduced as set of values. Such values can be instances of domain ontologies (e.g., a price defined by amount and currency) or variables defined by an axiom. The exploitation of axioms allows for a great freedom in NFP definition, allowing for description such as "if the client is older than 60 or younger than 10 years old the invocation price is lower than 10 euro"; an example is given in Section 4.

3.2 Representation of NFP clusters with NFP Policies

Besides modeling single NFPs, we have extended the WSMO model with the concept of Policy. A NFP policy (policy for short) is *a cluster of non functional properties* offered or requested (e.g., a specification of price, of a security level that is guaranteed and so on). More than one policy can be associated with a service and each policy has an *applicability condition* specifying when it can be accessed (e.g., the *premium* policy can be accessed only by clients older then 60).

In WSMO, a policy collects a set of NFPs that are elements of the class nonFunctionalProperty. Policy conditions are described by axioms; a WS can have one of more conditions; when more conditions are specified their conjunction is taken to hold, as it is defined for similar WSMO elements. Formally, the WSMO conceptual model can be extended by the class Policy characterized as follows (Listing 1.3).

Listing 1.3. WSMO conceptual model extensions

```
Class policy
    hasAnnotations type annotations
    importsOntology type ontology
    usesMediator type ooMediator
    hasCondition type Axiom
    hasNfp type nonFunctionalProperty
```

Imported ontologies allows for reuse of existing ontologies in the policy descriptions; mediators can be exploited to support matching between offered and requested policies, as described in [3]. Policies are attached to service descriptions as done for single NFPs. A simplified model of a WSMO service resulting from this approach is illustrated in Listing 1.4:

Listing 1.4. Service description structure with policies

```
webService
    capability idCapability
        precondition definedBy axiomPrecond
        postcondition definedBy axiomPostcond
        assumption definedBy axiomAssumption
        effect definedBy axiomEffect
    nonFunctionalProperty
        idNFP hasValue valuelistnfp definedBy axiomNFP
    policy idPolicy
            policyCondition definedBy axiomPolicy
                nonFunctionalProperty
                idNFP hasValue valuelistnfp definedBy axiomNFP
                ...
                nonFunctionalProperty
                idNFP hasValue valuelistnfp definedBy axiomNFP
```

The advantage is to give developers the freedom to model non functional properties as single NFPs or clustered NFPs defined by policies. However, it should also be stated that the the use of both policies and single NFPs together should be avoided.

The policies that are included in service descriptions can be extracted to form independent entities identified by URIs, supporting policy reuse. In fact, the specification of the fields *idPolicy* and *valuelistnfp* may refer to instances of existing ontologies. This is a key factor to address flexibility, since the same policy can be associate with more services. A positive side effect is to reduce the effort in writing semantic policy descriptions, which could be a non trivial task for complex specifications (e.g., legal clauses). In particular, this opens to future solutions in which policy descriptions are collected in repositories to address NFP management, reuse and evolution, separately from WS and functional specification management. Repositories will be able to support the creation of standard and shared policies to address specific needs (e.g., to conform business rules of an organization or to fulfill legal prescriptions).

In principle, single-NFP and policy-based approaches are equally expressive; however, the policy-based approach provides a more direct, explicit, and compact way to cluster NFPs; when such clusters are taken into account the policy-based approach is easier to read and manipulate due to the explicit linguistic structure; when more freedom in modelling is needed the single NFP approach can be adopted.

4   Examples of flexible NFP publishing

In this section, we present some examples to illustrate the proposal[6]. Listing 1.5 shows how NFPs can be individually attached to service descriptions by exploiting WSML axioms. The axiom states that a given price (lower than 10 euro) is for clients older than 60 or younger than 10 years.

---

[6] Please notice that the WSML syntax used in this section is not yet supported by WSMO/WSML tools (e.g. WSMT). Future extensions of WSMO and WSML towards better support of non-functional properties and policies, as well as the associated tool support will be most probably based on the proposal described in this paper.

Listing 1.5. Example of NFPs descriptions based on the proposed extensions

```
webService _"http://example.org/ws"
importsOntology {_"http://www.wsmo.org/ontologies/nfp/priceNFPOntology",
                 _"http://www.example.org/ontologies/humansOntology",
                 _"http://www.example.org/ontologies/clientsOntology" }

    nonFunctionalProperty

        price hasValue ?price
            definedBy ?client[age hasValue ?age] memberOf hu#human and ?age[amount hasValue ?years,
                    units hasValue hu#YearsDuration] memberOf hu#age and (greaterEqual(?years, 60) or
                    lessEqual(?years,10)) implies ?price[hasAmount hasValue ?amount, hasCurrency
                    hasValue cur#Euro] memberOf po#AbsoultePrice and lessEqual(?amount, 10).

    capability wsCapability
    interface wsInterface
```

In the next example (Listing 1.6), a service offers two different policies, one for users applying for repetitive travels, and one for 'Gold' subscribers users. The conditions associated with the policies state the client status to access the policy and consequently the corresponding property values. In order to show the modularity of the approach, the example exploits the PCM external ontology [14] to describe the involved NFPs. Moreover, the Gold policy refers to the external ontology PolicyRepository where this policy is described (Listing 1.7).

Listing 1.6. Example of Web Service description including policy descriptions

```
webService WSOmniTransport

importsOntology{ _"http://www.siti.disco.unimib.it/ontologies/LOOntology#LOOntology",
               _"http://www.siti.disco.unimib.it/ontologies/PCMOntology#PCMOntology",
               _"http://www.siti.disco.unimib.it/ontologies/NFPOntology#NFPOntology",
               _"http://www.siti.disco.unimib.it/ontologies/PolicyRepository#PolicyRepository" }

    policy policyForRepetitiveTravel

        policyCondition repetitiveTravelCondition
            definedBy pcm#appliable(policyForRepetitiveTravel) :− _#[lo#numberOfTravelsSubscribed
                    hasValue ?n and ?n > 1] memberOf lo#LogisticOperatorClient.

        nonFunctionalProperty
            paymentMethod hasValue offeredPaymentMethodOT
            definedBy offeredPaymentMethodOT[constraintOperator hasValue exact
                    pcm#parameters hasValue nfpo#carriageForward] memberOf nfpo#ListPaymentMethod.

        nonFunctionalProperty
            price hasValue offeredBasePriceOTrepetitive
            definedBy offeredBasePriceOTrepetitive[pcm#constraintOperator hadValue pcm#equal,
                    pcm#parameter hasValue 20, pcm#unit hasValue nfpo#euro]
                    memberOf nfpo#SingleValueBasePrice.

        nonFunctionalProperty
            deliveryDuration hasValue offeredHoursToDeliveryOTrepetitive
            definedBy offeredHoursToDeliveryOTrepetitive[pcm#constraintOperator hasValue pcm#interval,
                    pcm#minParameter hasValue 24, pcm#maxParameter hasValue 48,
                    pcm#unit hasValue nfpo#hour] memberOf nfpo#RangeHoursToDelivery

    policy pr#policyGold

    capability wsCapability
    interface wsInterface
```

Three NFPs are represented in the example, namely, Payment Method (the method accepted for paying the service), Base Price (the starting price upon which total price is calculated taking into account size, weight and distance), and Hours To Delivery (the time spent by the providers before shipping the goods). The value of each NFP is an instance in the ontology; the properties of this instance are specified by the axiom immediately following the WSML construct "defined by". In the example the two policies offer the same payment methods, but associated with different base price and duration of the delivery. Observe that the delivery duration cannot be defined with a fixed value; therefore a range is published according to PCM range constraint operators.

Listing 1.7. Example of independent policy definition

```
ontology PolicyRepository

importsOntology{ _"http://www.siti.disco.unimib.it/ontologies/LOOntology#LOOntology",
                 _"http://www.siti.disco.unimib.it/ontologies/PCMOntology#PCMOntology",
                 _"http://www.siti.disco.unimib.it/ontologies/NFPOntology#NFPOntology" }

        policy policyGold

        policyCondition goldCondition
            definedBy pcm#appliable(policyGold) :− _#[lo#userTypology hasValue lo#subscribed]
                memberOf lo#LogisticOperatorClient.

        nonFunctionalProperty
            paymentMethod hasValue offeredPaymentMethodOT

        nonFunctionalProperty
            price hasValue offeredBasePriceOTgold
            definedBy offeredBasePriceOTgold[pcm#constraintOperator hadValue pcm#equal,
                pcm#parameter hasValue 15, pcm#unit hasValue nfpo#euro]
                memberOf nfpo#SingleValueBasePrice.

        nonFunctionalProperty
            deliveryDuration hasValue offeredHoursToDeliveryOTgold
            definedBy offeredHoursToDeliveryOTgold[pcm#constraintOperator hasValue pcm#interval,
                pcm#minParameter hasValue 12, pcm#maxParameter hasValue 32,
                pcm#unit hasValue nfpo#hour] memberOf nfpo#RangeHoursToDelivery
```

5 Conclusions and Future Work

This paper has introduced a modeling approach for non-functional properties and policies. Both service provider and consumer perspectives are supported allowing the specification of what, how and in which way non-functional properties and policies are provided and requested. Distinctive features are *(i)* the semantic description of non-functional properties and policies that heavily use ontologies to define terminology and concepts; *(ii)* the declarative approach to model non-functional properties and policies using axioms, and *(iii)* the definition of policies as independent entities that clusters non-functional properties and related applicability condition.

Currently the proposal is under evaluation to become part of the official definition of WSMO/WSML. Future plans deal with the development of a lighter

version of the descriptions. This is motivated by emerging lighter modeling frameworks for Semantic Web service, such as WSMO-Lite[19] and SA-WSLD[7]. The aim is to deliver lighter ontologies describing a reduced but sufficient (most common used) set of non-functional properties and policies. A key aspect of the current and future activity is the development of tools to evaluate the practical applicability of the proposal. In the framework of the sponsoring projects we are developing a discovery engine and case studies. Moreover, these topics will be part of next Semantic Challenge proposals [2].

Acknowledgements

References

1. J. Aagedal and J. Earl F. Ecklund. Modelling QoS: Towards a UML Profile. In *UML '02: Proceedings of the 5th International Conference on The Unified Modeling Language*, pages 275–289, London, UK, 2002. Springer-Verlag.
2. A. Carenini, D. Cerizza, M. Comerio, E. D. Valle, F. D. Paoli, A. Maurino, M. Palmonari, M. Sassi, and A. Turati. Semantic web service discovery and selection: a test bed scenario. In *In proc of. Sixth International Workshop on Evaluation of Ontology-based tools and the Semantic Web Service Challenge (EON & SWS-Challenge 2008)*, 2008.
3. A. Carenini, D. Cerizza, M. Comerio, E. D. Valle, F. D. Paoli, A. Maurino, M. Palmonari, and A. Turati. Glue2: a web service discovery engine with non-functional properties. In *ECOWS*. IEEE Computer Society, 2008.
4. M. Comerio, F. DePaoli, A. Maurino, and M. Palmonari. Nfp-aware semantic web services selection. In *Proc. of the Enterprise Computing Conference (EDOC).*, Annapolis, Maryland, USA, 2007.
5. A. R. Cortes, O. M. Diaz, A. D. Toro, and M. Toro. mproving the Automatic Procurement of Web Services Using Constraint Programming. *Int. J. Cooperative Inf. Syst.*, 14(4):439–468, 2005.
6. J. de Bruijn, H. Lausen, R. Krummenacher, A. Polleres, L. Predoiu, M. Kifer, and D. Fensel. The Web Service Modeling Language WSML. Technical report, DERI. WSML Final Draft D16.1v0.21. http://www.wsmo.org/TR/d16/d16.1/v0.21/.
7. S. Frølund and J. Koistinen. Qml: A Language for Quality of Service Speicifcation. Techincal report, Hewlett Packard, 02 1998. Available from http://www.hpl.hp.com/techreports/98/HPL-98-10.html.
8. G. Gangadharan, M. Comerio, H.-L. Truong, V. D'Andrea, F. D. Paoli, and S. Dustdar. License-aware Service Selection. In *IEEE 2008 Conference on Enterprise Computing, E-Commerce and E-Services (EEE '08)*, Crystal City, Washington, D.C., USA, July 2008. IEEE Computer Society.

---

[7] http://www.w3.org/2002/ws/sawsdl/

9. E. Giallonardo and E. Zimeo. More semantics in qos matching. In *Proc. of International Conference on Service-Oriented Computing and Application (SOCA07)*, pages 163–171, Newport Beach, CA, USA, 2007.

10. K. Kritikos and D. Plexousakis. Semantic qos metric matching. In *ECOWS '06: Proceedings of the European Conference on Web Services*, pages 265–274, Washington, DC, USA, 2006. IEEE Computer Society.

11. E. Maximilien and M.P.Singh. A framework and ontology for dynamic web services selection. *IEEE Internet Computing*, 08(5):84–93, 2004.

12. J. O'Sullivan, D. Edmond, and A. H. ter Hofstede. Formal description of non-functional service properties. Technical report, Queensland University of Technology, Brisbane, 2005. Available from http://www.service-description.com/.

13. OWL-S. *Semantic Markup for Web Services*. Available at: http://www.daml.org/services/owl-s/1.0/owl-s.html, 2003.

14. F. D. Paoli, M. Palmonari, M. Comerio, and A. Maurino. A meta-model for non-functional property descriptions of web services. In *ICWS*, pages 1048–1055. IEEE Computer Society, 2008.

15. D. Roman, H. Lausen, , and U. Keller. *Web Service Modeling Ontology (WSMO)*. www.wsmo.org, 2004.

16. N. S. Rosa, P. R. Cunha, L. Freire, and G. R. Justo. Process NFL: A language for describing non-functional properties. In *Proceedings of the 35th Annual Hawaii International Conference (HICSS), March 29, 2005, Hawaii, USA*, pages 3676–3685, 2002.

17. I. Toma, D. Roman, and D. Fensel. On describing and ranking services based on non-functional properties. In *NWESP '07: Proceedings of the Third International Conference on Next Generation Web Services Practices*, pages 61–66, Washington, DC, USA, 2007. IEEE Computer Society.

18. D. Tsesmetzis, I. Roussaki, I. Papaioannou, and M. Anagnostou. Qos awareness support in web-service semantics. In *In AICT/ICIW '06: Proceedings of the Advanced International Conference on Telecommunications and International Conference on Internet and Web Applications and Services*, pages 128–135, Guadeloupe, French Caribbean, 2006.

19. T. Vitvar, J. Kopecký, M. Zaremba, and D. Fensel. Wsmo-lite: Lightweight semantic descriptions for services on the web. In *ECOWS '07: Proceedings of the Fifth European Conference on Web Services*, pages 77–86, Washington, DC, USA, 2007. IEEE Computer Society.

20. K. J. L. C. Weibel, S. and M. Wolf. Rfc 2413 - dublin core metadata for resource discovery. Technical report, 1998.