A Meta-Model for Non-Functional Property Descriptions of Web Services

Flavio De Paoli, Matteo Palmonari, Marco Comerio, Andrea Maurino
Dipartimento di Informatica Sistemistica e Comunicazione
Università di Milano - Bicocca,
Viale Sarca 336/14, 20126 Milano Italy
Email: depaoli, palmonari,comerio,maurino@disco.unimib.it

Abstract

In this paper we propose a meta-model for non-functional property descriptions targeted to support the selection of Web Services. The approach is based on the explicit distinction between NFP offered by providers and requested by users, on the concept of policy that aggregates NFP descriptions into single entities with an applicability condition, and finally on a set of constraint operators, which is particularly relevant for NFP requests. The semantic meta-model embracing the above perspective is defined by a BNF syntax whose semantics is formalized by an ontology. The ontology has been formalized in OWL-DL and WSML to provide for logical syntaxes. The logic upon which the meta-model supports NFP-based selection is discussed in the paper.

1 Introduction

Service-Oriented Computing (SOC) is a computing paradigm that considers services as basic constructs for the development of rapid, low-cost and easy-to-compose distributed applications within and across organizational boundaries [16]. Web services technology enables the implementation of services that are distributed over the network and are available via standard protocols.

Nevertheless, there is a growing consensus that pure functional descriptions of Web Services (WS) are inadequate to develop valuable processes due to the high degree of heterogeneity, autonomy, and distribution of the Web [2]. Moreover, the increasing availability of Web services that offer similar functionalities with different characteristics increases the need for more sophisticated discovery processes to match user requests [14]. A way to address this problem is to combine Semantic Web and Web service technologies. The focus is on the discovery and selection activity that is requested to identify the needed Web services and supply the business information to include them in a business process. Today, there is still not a clear agreement on the definition of terms discovery and selection. In this paper we call discovery the activity of locating a machine-processable description of a SWS that meets certain functional criteria [18]; and selection the activity of evaluating the discovered services in order to identify the ones that fulfill a set of non-functional properties requested by the actual user [10]. The final output is an ordered set of services that satisfy both functional and non-functional requirements.

In this paper, we propose the Policy Centered Meta-model (PCM) that provides developers, providers and users with a frame to describe non-functional properties (NFP) that can be exploited to match service requests and service offers to support service selection and composition. The approach is based on (i) the explicit distinction between NFP offered by providers and requested by users, (ii) the concept of policy that aggregates NFP descriptions into a single entity with an applicability condition, and finally (iii) a set of constraint operators, which is particularly relevant for NFP requests. The meta-model embracing the above perspective is formalized by an ontology that provides a good balance between expressiveness and complexity in the resulting NFP descriptions. The conceptual syntax of the meta-model is defined by a BNF grammar, whose logical grounding is given by formalizations in both OWL-DL [8] and WSML [6]; the WSML formalization has been exploited to develop a discovery engine to evaluate the approach.

The paper is organized as follows. Section 2 motivates the definition of a meta-model for non-functional property description. Section 3 defines the meta-model. Section 4 discusses an example of a NFP-based service selection process in a logistic scenario, and finally, Section 5 discusses the related works. Section 6 draws conclusions and shows some relevant open problems.

2 Issues in NFP-based selection

In a service oriented domain, the properties that completely specify a service can be classified as (1) functional,
(2) behavioral and (3) non-functional [4]. Functional properties (FPs) represent the description of the service tasks in terms of operation signatures (operation names and input/output schema) or more sophisticated descriptions (e.g., finite state automata specifications [17]).

Non-functional properties (NFPs) represent the description of the service characteristics that are not directly related to the functionality provided. Nevertheless, in the literature there is not yet an agreement about what properties should be classified functional or non functional. From our point of view this is a consequence of the fact that functional or non functional is not an intrinsic qualification of a property, but it depends on domain and context. For example, the service location is classified as a FP for a logistic service and as a NFP for a payment service. From a requester perspective, non-functional properties can be classified hard or soft constraints to distinguish between mandatory properties and optional ones. Even if NFPs are soft constraints, they 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 for a specific user in terms of NFPs (e.g., if availability is not sufficient, performance is too poor, cost is too expensive).

The definition and management of NFPs is a complex task due to the nature of the involved properties. 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). Moreover, 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’). Finally, some NFPs reveal inter-dependencies, therefore they cannot be independently assigned.

Therefore, there is the need for a systematic approach to NFP description to support selection activity that involves business properties besides the traditional technological ones. Our proposal is to define a semantic meta-model that provides a sound and robust base to formally define non-functional properties. These definitions will augment the functional descriptions of Web services (e.g., WDSL descriptions). Ontologies are the means to implement such definitions.

2.1 Why a meta-model

Many Web service models specify non-functional properties by means of attribute-value clauses (e.g., [21, 15, 6]), where attribute identifies the involved non-functional property and value specifies the value of the property offered by the associated service. Shortcomings of such approaches are:

1. Attributes are usually labels that are defined by textual descriptions in natural language; there is not a formal definition of measurement methods and units. Moreover, qualitative dimensions are not standardized. Therefore, automatic selection is difficult to achieve due to the high probability of semantic misunderstandings.

2. Interdependencies among quality dimensions cannot be expressed since every dimension is considered in isolation.

3. Properties cannot be easily clustered to form a joint offer.

4. Offered properties are represented by single static values: range-based offers are not easily expressed. Notice that existing semantic models (OWL-S [15] and WSMO [20]) also do not support range descriptions.

5. The representation of non-functional properties whose values are computed dynamically are not directly supported. In fact, neither the possible values of the properties, nor the methods used to assign them can be specified by means of attribute-value clauses. In semantic based models (e.g., OWL-S) dynamic properties could be defined by variables which are further included in axiom specifications; however, only few existing semantic languages directly support variables.

6. The capability to express NFP requested by users is limited. In practice typical non-functional property requirements require advanced mathematical operators (e.g., ‘≤’, ‘≥’, range) that are not usually supported.

2.2 Characteristics of PCM

A meta-model for non-functional properties should be general and expressive enough to address the most significant issues to support the matching of requested and offered properties. The primary goal is to support service description and selection according to the following characteristics:

Support for sophisticated descriptions. Requests, and also offers, should refer to a full set of constraint operators, including ‘≤’, ‘≥’ and range. Moreover, relevance attributes should specify the importance that requesters give to each requested NFP. The use of relevance attributes and operators in constraint expressions (e.g., the cost of the service must be ≤ 3 Euro) enhance the expressivity of the descriptions to support non-boolean matching. In practice, as far as FPs are concerned, a boolean match between requests and offered services is reasonable: a service that does not exactly fulfill the requested functionalities should be discarded. On the contrary, matching between
offered and requested NFPs should be considered not crispy and degrees of satisfaction should be evaluated (i.e., a NFP request can be fully, partially or not satisfied).

Support for offer clustering. A Web service can be offered with different levels of NFPs. Clustering NFPs in sets called policies is one of the most important features of PCM to capture business scenarios by aggregating interdependent properties. For example, a SMS service offered by a telephone company can be characterized by different policies: one offering a price of 0.10 Euro for SMS and a maximum of 100 SMS per day; another offering a price of 0.15 Euro and no restriction on the number of SMS per day. Moreover policies can be associated with conditions to state that their applicability depends on the requester’s profile or context; e.g., a condition may grant a discount on shipment services to registered customers or for multiple shipment requests.

Support for static and dynamic properties description. Static NFPs specify properties of services by fixed values that can be included in service descriptions at publishing time. Dynamic NFPs describe properties of services that are strictly related to the context of execution and cannot be stated at publishing time. Moreover, dynamic NFPs require service calls to fix the actual values. For example, the actual price for shipment depends on the shipped object’s weight, therefore a call is necessary to fix it.

3 The Policy Centered Meta-Model

In this section the Policy Centered Meta-model (PCM) is described by a BNF grammar¹ that provides a conceptual syntax whose semantics is defined by an upper-level ontology. The ontology has been formalized in the most popular semantic languages in the Web Service domain, namely OWL-DL and WSML. Each formalization provides a logical syntax that can be exploited to write actual descriptions of requested and offered NFPs.

3.1 PCM conceptual syntax

Figure 1 is a graphical representation of the ontology that illustrates the top-level concepts and relationships of the proposed meta-model. Policy is the main concept of the meta-model. It is characterized by the following elements, Id: the URI that makes the policy available over the Internet; name: a given name that improves human readability, it can be used internally to identify a policy; serviceReference: the reference to the functional description of services the policy is attached to, e.g., the URI of a WSDL description; policyCondition: a logical element that defines the conditions for the validity of the policy; PolicyNfp: one or more elements that specify the properties and their values.

The grammar to define instances of this concept is:

```
Policy := 'policy' Id
PolicyHeader := 'PolicyHeader' PolicyNfp *
Id := URI
PolicyHeader := 'name' String // label for human readability
'serviceReference' Id */
'policyCondition' PolicyCondition
'description' string // textual description of conditions
PolicyCondition := Id ['definedBy'
LogicalExpression]*
LogicalExpression := // axiom definition in the form of the selected semantic language
```

The serviceReference allows providers to link a policy to one or more WS, independently from the chosen WS description language. The aim of policy condition is to define the conditions for client profile to select the policy (e.g., the senior policy is for clients older than 60). A policy condition is an instance of the PolicyCondition concept that can be further specified by LogicalExpressions that are axioms in the semantic language used to express the offer.

The core of the meta-model is the definition of the non-functional properties (PolicyNfp) by means of the following elements. Id: the URI that makes the non-functional property available over the Internet; NfpClass: the ontology concepts of which this is an instance; NfpExpression: the expression that characterizes the properties and their values.

The corresponding grammar is reported in the next listing. Each NfpExpression is specified by a ConstraintOperator and by a set of attributes that depend on the con-
Consequently, starting from a classification of ConstraintOperator, a classification of PolicyExpression and PolicyNfp has been defined. Figure 2 shows the logical relationships between the classifications. PCM distinguishes between qualitative and quantitative properties. The former are properties defined through qualitative expressions that refer to objects (their values are instances of given domain ontologies); the latter use quantitative expressions that assume numeric values, whose measurement units is specified by a ‘unit’ term.

```plaintext
PolicyNfp := 'nfp' | \Id NfpClass\ NfpExpression
NfpClass := 'memberOf' | \ Id \ // specifies the nfp class the property is member of
NfpExpression := SingleValueExpression | RangeExpression | SetExpression | UserDefinedExpression
SingleValueExpression := BinaryOperator numericValue 'unit' unitValue
RangeExpression := TernaryOperator '(' numericValue , numericValue ')' 'unit' unitValue
SetExpression := SetOperator Id∗ UserDefinedExpression := UserDefinedOperator Id∗ BinaryOperator := 'greaterEqual' | 'lessEqual' | 'equal' | 'atLeast' | 'atMost'
TernaryOperator := 'interval'
SetOperator := 'all' | 'exist'
UserDefinedOperator := // user defined
```

The model introduces a set of logical operators (e.g., all, exist) of class SetOperator. However, the set of qualitative operators can be extended with user-defined operators by domain specific functions. As an example, a semanticDistance operator along with a defining function can be introduced to support ranking operations in a domain ontology.

As for qualitative expressions, we have defined a set of operators that support the most common clauses for numeric values (e.g., inequalities and ranges). Beside the standard operators 'equal' and 'interval' (that fixes minimum and maximum values in range), new operators have been introduced to increase expressiveness of inequalities. These operators are: (i) 'greaterEqual' ('≥'): the highest possible value is better; (ii) 'atLeast' ('≥'): the lowest possible value is better; (iii) 'lessEqual' ('≤'): the lowest possible value is better; (iv) 'atMost' ('≤'): the highest possible value is better. Note that the referred property is an implicit parameter for binary and ternary operators.

The counterpart of offered NFP is required NFP. The aim of a request is to state what values are acceptable for a certain property, and possibly to express the relevance of each required property. Therefore, Request is a subclass of PolicyNfp augmented with the definition of relevance values. RequestedPolicy is a subclass of Policy, whose associated PolicyNfp are Requests. The BNF is:

```plaintext
RequestedPolicy := 'requestedPolicy' | Id PolicyHeader Request∗
Request := PolicyNfp 'relevance' RelevanceValue
ReferenceValue := Float // in range [0, 1]
```

### 3.2 The PCM ontological semantics

Although the meta-model has been designed to be independent from any specific language, the ontology defining its semantics has been formalized in the OWL-DL and WSML languages. OWL is in fact the language adopted by the semantic Web Service model OWL-S and is recalled as mark up language in light semantic annotation languages such as SAWSDL [11]. WSML is a family of languages used for the Web Service Modeling Ontology WSMO. Besides the differences between WSML and OWL-DL, the two formalization of PCM are largely equivalent but for small details.

For reasons of space, in this section only the definition of main concepts is given. Interested readers can retrieve the complete formalizations at the Website: http://www.siti.disco.unimib.it/research/ontologies/.

Our primary formalization is in WSML because of the WSMO meta-model that natively includes the distinction between service offer and service request. The latter is modeled by the concept of goal. Goal descriptions have the...
same schema of WS descriptions; that is, they are described mainly by capabilities (preconditions, effects, assumptions, postconditions), and annotations.

WSML-Flight is a language based on frame-logic and Logic Programming (LP); syntax of LP rules in WSML-Flight is a Object Oriented (or ontological) notational variant of normal logic programs (Prolog-like Horn Logic) [6]. WSML-Flight defines a conceptual syntax over the Frame Logic based core to provide a more user friendly description language. The syntax used for the meta-model formalization will be explained together with the axiom description.

The top concepts are formalized by the following axioms:

```
concept Policy
  hasServiceReference ofType URI
  hasNfp ofType (1 ->) PolicyNfp
  hasConditionDescription ofType (0 1) string
  hasCondition ofType (0 1) PolicyCondition
concept PolicyNfp
  hasExpression ofType (1 1) PolicyExpression
concept Request subConceptOf PolicyNfp
  hasRelevance ofType (1 1) float
```

where the literals hasX are used to give an intuitive name to relationships associated with a concept, the clause (n m) states the cardinality of a relationship.

A set of axioms define the dependencies between the three classifications of ConstraintOperator, NfpExpression and PolicyNfp represented in Figure 2 (they are omitted here for reasons of space). An example of operator definition is given by SingleValueExpression that declares a binary operator, a value and a measurement unit:

```
concept SingleValueExpression subConceptOf QuantitativeExpression
  hasOperator ofType (1 1) BinaryOperator
  hasParameter ofType (1 1) float
  hasUnit ofType (1 1) Unit
```

Qualitative expressions require the definition of Logical Expressions that are defined by a logical operator and a set of values:

```
concept LogicalExpression subConceptOf QualitativeExpression
  hasOperator ofType (1 1) SetOperator
  hasParameters ofType (1 +) <anyType>
```

The operators can be predefined (e.g., exist) or user defined to support extensions by inclusion of domain ontologies. In the later case, the definition becomes:

```
concept UserDefinedExpression subConceptOf QualitativeExpression
  hasOperator ofType (1 1) UserDefinedOperator
  hasParameters ofType (1 +) <anyType>
```

3.3 Policy examples

The next listing provides an example of policy in the context of logistics operators. The term 'instance' is used in WSML to introduce a description of an instance of the ontology\(^2\). Namespaces such as 'pcm#' and 'nfpo#' refer respectively to the PCM ontology and a domain NFP ontology extending the PCM (imported in the descriptions). The last ontology defines a class representing the payment methods available in the domain, and SetPaymentMethod as a qualitative NFP whose parameters are of class PaymentMethod. Therefore in the offer description one can refer directly to the PCM meta-model (e.g. offeredHoursToDelivery in the example) or to domain NFP ontologies defined according to the PCM. Moreover, multiple inheritance can be exploited to use other independent NFP ontologies together with the PCM, e.g. by defining a non functional property ReliabilityOffer which is both subclass of PolicyNfp and of DomainOntology#Reliability, where DomainOntology is any domain ontology describing the concept Reliability. In this way the specifications are inherited from PCM, and new descriptive attributes are inherited from the imported domain ontology.

```
instance goldPolicy memberOf pcm#Policy
  pcm#hasServiceReference hasValue
    'http://www.siti.disco.unimib.it/research/ontologies/WSSouthernItalyOrdinaryTransport.wsm1'
  pcm#hasNfp hasValue offeredPaymentMethod
  pcm#hasNfp hasValue offeredHoursToDelivery
  pcm#hasNfp hasValue offeredPaymentMethodExpression
  pcm#hasOperator hasValue pcm#all
  pcm#hasParameters hasValue nfpo#carrierPaid
  pcm#hasMaxParameter hasValue 48
  pcm#hasUnit hasValue nfpo#hour
```

In this model offered and requested non-functional properties are coupled by inheritance to deliver a stratified model. The next listing provides an example of requested policy in the context of logistic operator.

```
instance LORequest1 memberOf pcm#RequestedPolicy
  pcm#hasOperator hasValue requestedPaymentMethod
  pcm#hasNfp hasValue requestedPaymentMethodExpression
  pcm#hasMaxParameter hasValue 24
  pcm#hasMinParameter hasValue 48
```

\(^2\)Descriptions are flat since nested descriptions in ontology specification are not supported by current editing and processing tools for WSML.
4 Support for NFP-based WS Selection

The main purpose of PCM is to support service selection, which intends to address matching between service requests and offers. The selection process is composed of three phases:

- **Matching phase.** For each requested NFP, identify the set of offered NFP to be evaluated.
- **Local evaluation phase.** For each identified request/offer pair, evaluate how the offered property satisfies the requested one. Results are values in range [0, 1].
- **Global evaluation phase.** For each policy, evaluate the results of the previous phase to compute a global satisfaction degree for the candidate services. Results are values in range [0, n].

The meta-model has been formalized by an ontology to exploit semantic-based techniques to support the matching phase. When using WSML descriptions, the WSMO’s mediators-centered approach and related tools can be exploited. For example, semantic matching rules can be defined in WSML by defining ontology-based logic programs. The following listing is an example of a matching rule for service price; the rule specifies that a request and an offer match if they belong to specific classes of PolicyNfp.

```
axiom BasePriceMatching definedBy
  matchCouple(?request, ?nfp, baseprice) :-
  ( ?request memberOf nfpo#BasePriceRequest ) and
  ( ?nfp memberOf nfpo#BasePrice ) or
  ( ?nfp memberOf nfpo#ServicePrice )
```

By means of similar logic programming rules it is possible to extract information from policies to support the local and global evaluation phases. Reasoners can be used to evaluate the rules and compute the matches. With respect to the rule-based matching techniques, this approach is modular in the sense that new rules can be added to widen or narrow the number of matching policies considered (e.g. by introducing new subsumption statements).

Table 1 reports an example of matching results obtained with the KAON2 reasoner [9] (values are displayed by the generic attribute hasParameter which is set as ‘null’ when not applicable).

On such results, algorithmic techniques can be exploited for the evaluation of local satisfaction degrees [5]. This solution has been introduced to overcome the limits of semantic-based techniques to manage non-boolean expressions: for OWL-based languages, standard Description logics reasoners do not support any arithmetic reasoning; for WSML, the IRIS reasoner is expected to support some basic built-in functions but its development is still at an early stage; KAON2, the DL reasoner we used for the matching phase, provides no support for functions (because it is based on datalog).

The meta-model supports a full constraint-based perspective on satisfaction degree evaluation: the assumption is that the satisfaction degrees can be determined uniquely on the basis of the values and the constraint operator used in request and offer descriptions. This means that property declarations are not based on predefined assumptions, such as the assumption that prices have a downward direction (that is, lower prices are always preferred).

This approach makes clear boundaries to the evaluation space (the number of possible local evaluation functions depends only on the number of operators used).

Once local satisfaction degrees have been evaluated for NFP/request pairs, a global degree of satisfaction of the available policies with respect to the requested policy is computed. Basic global degree formulas are weighted sums where weights are the relevance that requesters have assigned to each NFP. The global degree of satisfaction can be therefore used to rank policies, and therefore services. Notice that given a set of results for the matching phase, the meta-model discussed in this paper allows for adopting different strategies for the definition of local and global satisfaction evaluation functions. Interested readers can refer to [5] for details about the definition of (i) a set of local evaluation functions and (ii) a ranking function based on a global evaluation strategy.

5 Related Works

Current standards for semantic descriptions of services (e.g., WSMO [6] and OWL-S [15]) only marginally cover the specification of NFPs. They basically adopt attribute-value descriptions as discussed in Section 2. Toma et al.

3http://iris-reasoner.org/download
[18] propose to overcome the WSMO limitation by specifying NFP through axioms. Other papers [12, 7] propose to fill the OWL-S gap by complementing it with models for Quality of Service (QoS), i.e., the subset of NFPs related to technological characteristics of a Web services. Kritikos and Plexousakis [12] propose a rich and extensible model of QoS to support matching for Web service selection. The model is designed into six facets that describe the characteristics of a QoS (e.g., metric, unit, value type). Giallonardo and Zimeo [7] define a model that allow service providers to advertise on the QoS offered, and service consumers to specify QoS requirements associated to an OWL-S profile.

Other papers (e.g., [13, 1, 19]) propose models for QoS descriptions that are not related to a particular standard. Maximilien and Singh [13] propose a model that facilitates providers in expressing policies and consumers in expressing preferences on QoS. The model is created in order to support QoS matching and matching degree evaluation. Afandi et al. [1] provide a model to extend service description and advertisement mechanisms to gather QoS information. The model provides support for Web service discovery and QoS monitoring. Tsesmetzis et al. [19] introduce a model to establish a set of rules that are used to represent QoS characteristics of Web services along with the relationships among them.

PCM addresses the following aspects for the description of NFPs: (i) the explicit distinction between requested and offered NFPs; (ii) the extensive use of constraint operators; (iii) the definition of relevance values for requested NFPs; (iv) the description of policies; (v) the definition of policy conditions. Moreover, PCM has been designed to provide a good tradeoff between expressiveness (i.e., the ability to describe the several facets related to NFPs) and complexity (i.e., the number of concepts introduced by the model). Observe that complexity influences the depth and length of a NFP description.

Table 2 summarizes the proposals in the literature and PCM in regards to the described aspects. Expressiveness and complexity of the approach proposed in [18] can hardly be evaluated since NFP are expressed by arbitrary NFP ontologies (e.g. the PCM itself); the use of axioms provides for flexibility at the cost of no support in writing and interpreting specifications.

Relevance and policy conditions are distinctive characteristics of PCM. The definition of combined offers can also be considered a distinguishing characteristic of PCM, since only Giallonardo and Zimeo and Kritikos and Plexousakis provide limited support by supporting association of more QoS offers with an OWL-S service profile. The models proposed by Afandi et al. and Tsesmetzis et al. present a good tradeoff between expressiveness and complexity but are limited w.r.t. other features in comparison. The model proposed by Maximilien and Singh provides a good balance between expressiveness and complexity but a limited set of constraint operators are supported and the explicit distinction between NFP offers and requests is not addressed. The proposal by Kritikos and Plexousakis outperforms PCM in expressiveness, but with the drawback of higher complexity due to the number of facets (with about 100 high-level concepts) that need to be described. Moreover, the specification of new constraint operators is supported but limited to numeric and string value types.

6 Conclusions and Future Works

This paper presented the Policy Centered Meta-model that has been designed to complement existing service descriptions by adding non-functional properties. The goal is to support service selection according to business requirements. Two aspects have been addressed: expressiveness, by supporting rich, flexible and extendable descriptions, and effectiveness, by combining semantic techniques with algorithmic ones. Effectiveness deals also with efficiency: the use of reasoners is limited to match requests and offers that are not agreed on in advance. This limits the impact of reasoners performance which is usually low. For the semantic aspects, tools are currently based on WSML to support descriptions and matching. On the contrary, evaluation of property expressions is performed by algorithmic tools. A PCM-based discovery engine and tools to support and facilitate the policy writing by means of wizards and macros are under development as part of research projects (see acknowledgments).

Current activities are targeted to test the expressiveness of the PCM and the efficiency of the policy discovery process. In order to compare the expressiveness of PCM
w.r.t. other models, a new scenario for the Semantic Web Service Challenge has been proposed [3]. Further development of the PCM deals with the definition and management of dynamic properties. Moreover, we plan to extend the current model to support service negotiation.

Acknowledgment

The work presented in this paper has been partially supported by the European IST project n. 27347 SEEMP - Single European Employment Market-Place, and the FIRB project NeP4B - Networked Peers for Business.

References


