Selection of the best composite Web service based on quality of service

Serge Haddad[1], Lynda Mokdad[2], and Samir Youcef[2]

[1] Laboratoire Spécification et Vérification, École Normale Supérieure de Cachan
`haddad@lsv.ens-cachan.fr`
[2] LACL, Université Paris-Est, Créteil
`lynda.mokdad@univ-paris12.fr`

Abstract. The paper proposes a general framework to composite Web services selection based on multicriteria evaluation. The proposed framework extends the Web services architecture by adding, in the registry, a new Multicriteria Evaluation Component (MEC) devoted to multicriteria evaluation. This additional component takes as input a set of composite Web services and a set of evaluation criteria and generates a set of recommended composite Web services. In addition to the description of the conceptual architecture of the formwork, the paper also proposes solutions to construct and evaluate composite web services. In order to show the feasibility of the proposed architecture, we have developed a prototype based on the open source jUDDI registry.

1 Introduction

Individual Web services are conceptually limited to relatively simple functionalities modeled through a collection of simple operations. However, for certain types of applications, it is necessary to combine a set of individual Web services to obtain more complex ones, called *composite* or *aggregated* Web services. One important issue within Web service composition is related to the selection of the most appropriate one among the different possible compositions. One possible solution is to use quality of service (QoS) to evaluate, compare and select the most appropriate composition(s). The QoS is defined as a combination of the different attributes of the Web services such as availability, response time, throughput, etc. The QoS is an important element of Web services and other modern technologies. Currently, most of works use successive evaluation of different, non functional, aspects in order to attribute a general "level of quality" to different composite Web services and to select the "best" one from these services. In these works, the evaluation of composite Web services is based either on a single evaluation criterion or, at best, on a weighted sum of several quantitative evaluation criteria. Both evaluation schemas are not appropriate in practice since: (i) a single criterion does not permit to encompass all the facets of the problem, (ii) weighted sum-like aggregation rules may lead to the compensation problem since worst evaluations can be compensated by higher evaluations, and

(iii) several QoS evaluation criteria are naturally qualitative ones but weighted sum-like aggregation rules cannot deal with this type of evaluation criteria.

The goal of this research is to propose a general framework to composite Web services selection based on multicriteria evaluation. The proposed framework extends the Web services architecture by adding, in the registry, a new Multicriteria Evaluation Component (MEC) devoted to multicriteria evaluation. This additional component takes as input a set of composite Web services and a set of evaluation criteria. The output is a set of recommended composite Web services. The paper also proposes a solution to generate the different potential compositions which will be the main input for the MEC. Further, the paper shows how composite Web services can be evaluated.

The paper is organized as follows. Section 2 presents some related work. Section 3 details the architecture of the proposed framework. Section 4, presents the implementation of the proposed architecture. Section 5 present the Multicriteria Evaluation Component (MEC). Section 6 shows how the set of potential composite Web services is constructed. Section 7 discusses the problem of composite Web service evaluation. Section 8 concludes the paper.

2 Related work

As underlined in the introduction, to choose among the different possible compositions, most of previous works use either a single QoS evaluation criterion or a weighted-sum of serval quantitative QoS evaluation criteria. The following are some examples. The author in [9] considers two evaluation criteria (time and cost) and assigns to each one a weight between 0 and 1. The single combined score is computed as a weighted average of the scores of all attributes. The best composition of Web services can then be decided on the basis of the optimum combined score. One important limitation of this proposal is the compensation problem mentioned earlier.

In [14], the service definition models the concept of "placeholder activity" to cater for dynamic composition of Web services. A placeholder activity is an abstract activity replaced on the fly with an effective activity. The author in [15] deals with dynamic service selection based on user requirement expressed in terms of a query language. In [13], the author considers the problem of dynamically selecting several alternative tasks within workflow using QoS evaluation. In [12], the service selection is performed locally based on a selection policy involving the parameters of the request, the characteristic of the services, the history of past executions and the status of the ongoing executions. One important shortcoming of [14][15][13][12] is the use of local selection strategy. In other terms, services are considered as independent. Within this strategy, there is no guarantee that the selected Web service is the best one.

To avoid the problem of sequential selection, Zeng et *al.* [16] propose the use of linear programming techniques to compute the "optimal" execution plans for composite Web service. However, the multi-attribute decision making approach used by the authors has the same limitation as weighted-sum aggregation rules, i.e., the compensation problem.

Maximilien and Singh [17] propose an ontology-based framework for dynamic Web service selection. However, they consider only a single criterion, which is not enough to take into account all the facets of the problem.

Menascé and Dubey [10] extends the work of Menascé et al. [11] on QoS brokering for service-oriented architectures (SOA) by designing, implementing, and experimentally evaluating a service selection QoS broker that maximizes a utility function for service consumers. These functions allow stakeholders to ascribe a value to the usefulness of a system as a function of several QoS criteria such as response time, throughput, and availability. This framework is very demanding in terms of preference information from the consumers. Indeed, consumer should provide to a QoS broker their utility functions and their cost constraints on the requested services. However, the most limitation of this work is the use of weighted-sum like optimization criterion, leading to compensation problem as mentioned earlier. One important finding of this paper is the use, by the QoS broker, of analytic queuing models to predict the QoS values of the various services that could be selected under varying workload conditions.

More recently, [1] use genetic algorithm for Web service selection with global QoS constraints. The authors integrate two policies (an enhanced initial policy and an evolution policy), which permits to overcome several shortcomings of genetic algorithm. The simulation on Web service selection shown an improved convergence and stability of genetic algorithm.

3 Extended Web service architecture

The Web service architecture is defined by 3WC in order to determinate a common set of concepts and relationships that allow different implementations working together [21]. The Web service architecture consists of three entities, the service provider, the service registry and the service consumer. The service provider creates or simply offers the Web service. The service provider needs to describe the Web service in a standard format, which is often XML, and publish it in a central service registry. The service registry contains additional information about the service provider, such as address and contact of the providing company, and technical details about the service. The service consumer retrieves the information from the registry and uses the service description obtained to bind to and invoke the Web service.

The proposed framework, in this paper, extends the Web services architecture by adding, in the registry, a new Multicriteria Evaluation Component (MEC) devoted to multicriteria evaluation. The general schema of the extended architecture is given in Figure 1. According to the requirement of the consumer, the registry opts either for conventional evaluation or for multicriteria evaluation. By default, the registry uses conventional evaluation; multicriteria evaluation is used only if the consumer explicitly specifies this to the registry manager. This ensures the flexibility of the proposed architecture.

However, the application of a multicriteria method needs the definition of a set of preference parameters. The definition of these parameters needs an important cognitive effort from the consumer. To reduce this effort, MEC uses specific Web service called W-IRIS which is a Web version of IRIS (Interactive Robustness analysis and Parameters Inference for multicriteria Sorting Problems) [23] system permitting to infer the different preference parameters.

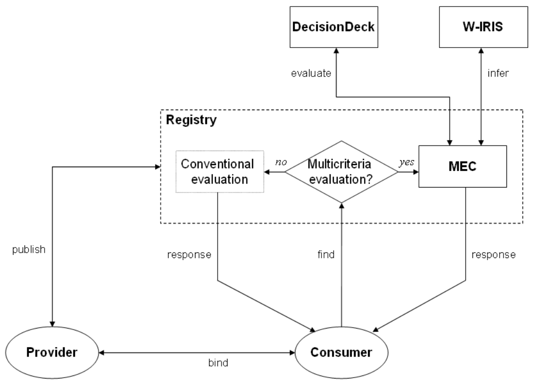

Fig. 1. Extended architecture of Web services

As we can see in Figure 1, the three basic operations of the Web service architecture denoted by publish, bind and find still exist. Two additional operations, denoted by keywords infer and evaluate are included in the extended architecture. The first permits to handle data exchange between MEC and W-IRIS. The latter permits to handle data exchange between MEC and DecisionDeck platform.

To achieve the interaction among the entities of the extended Web service model, we need to extend some SOAP protocoles and add new ones. More specifically, we need to extend protocols of consumer request to registry and registry response to consumer; and add the ones relative to MEC request to W-IRIS and W-IRIS response to MEC. A detailed description of the proposed architecture is given in Figure 2.

W-IRIS permits to infer the different preference parameters needed to apply multicriteria evaluation using ELECTRE TRI method. The inference procedure included in W-IRIS needs the resolution of different mathematical programs. For this purpose, W-IRIS includes the solver GLPK, which is an open-source and free package (see [24]).

The current version of MEC supports the advanced multicriteria method ELECTRE TRI (see [5]) and several elementary methods (weighted sum, conjunctive and disjunctive rules and the majority rule). Additional methods will be included in the future via the DecisionDeck platform. The DecisionDeck platform is issued from D2-Decision Deck project that has started in 2003 under the

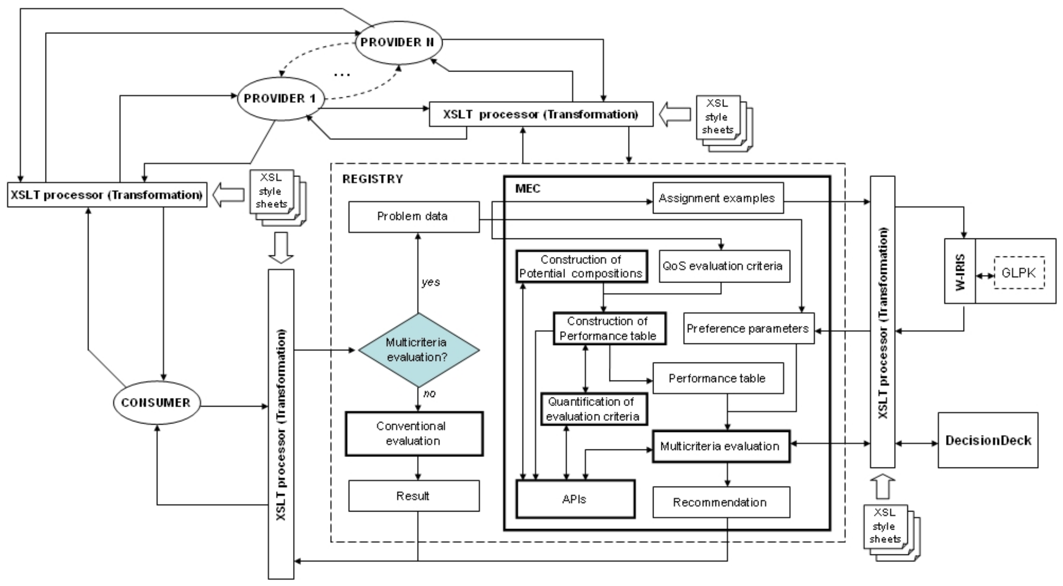

Fig. 2. Dynamics of the system

name EVAL, an acronym which refers to an ongoing research project funded by the Government of the Walloon region (Belgium). The aim is to develop a Web-based platform to assist decision makers in evaluating alternatives in a multicriteria and multi-experts context.

In the following, we present the jUDDI extensions and its implementation. More precisely, we detail the required extension/addition to support data exchange between the different entities of the proposed architecture.

4 jUDDI extension and implementation

jUDDI information model is composed of data structure instances expressed in XML schema. They are stored in jUDDI registries. A service is discovered by sending requests based on service information. The four core data elements within the jUDDI data model are described bellow (more information can be found in http://juddi.og): i) businessEntity: contains informations business, such as name and contact (each entity may provide various businessEntity); ii) businessService: contains informations about published services; iii) bindingTemplate: represents a service implementation and provides the information needed to bind with the service; and iv) tModel: is used to establish the existence of a variety of concepts and to point to their technical definitions.

In addition to the existence entities, we defined the following elements: i) qosInscription: contains customers who wish to take into account the QoS in their search of services in the extended registry; ii) qosParameters: contains the different parameters, for each customer registered to this option, needed to use multicriteria methods. Note that these parameters can be provided directly by the user or deduced using W-IRIS service (see section 6); iii) qosDescription: contains the QoS values for each service provider. The provider requests service publication and providers the QoS values. These last are checked and validated by the registrymanager. Note that QoS values can be update by the registry manager, and if a value is not provided, thus it is valued at worst. The qos-Description table refers to the bindingTemplate table that stores Web services instances. It also refers to the tModel.

The registry is implemented using Apache jUDDI Version 0.4rc4 which is an open source UDDI implementation compliant with Version 2.0 specification. MySQL Version 5.0.16 was used to implement the jUDDI databases. UDDI4J (version 2.0.4) is an open source Java class library that provides an APIs (Application Programming Interfaces) to interact with a jUDDI. They are grouped in three APIs categories: i) Iniquity APIs set, ii) Publication APIs set and iii) Security APIs set. The extended registry includes extensions to the UDDI4J Inquiry and Publication APIs set in order to manipulate the QoS related data. The extended registry is done thought registry manager, who implements the QoS management operations (see section 3). Experiments results, by simulation, are effectuated and show the compatibility withe the basic UDDI and both types of UDDI registries and can coexist in the same environment.

The W-IRIS is a special kind of Web service used by MEC to infer the preference parameters to use with ELECTRE TRI method. This method is used by MEC to assign composite Web services into different categories. It applies when “type of result” in the SOAP message sent by the consumer to registry is “sorting” (see Figure 3). The XML schema of the “infer” SOAP message sent by the MEC to W-IRIS and the same information are included in the “sorting_data” element (see Figure 3).

In the most general case, the inputs of W-IRIS are: (i) the number of categories, (ii) a set of profile limits, and (iii) a set of assignment examples. All of these data are extracted from the SOAP message sent by the consumer to registry detailed in the previous subsection (see Figure 3). As underlined earlier, the number of categories is an optional parameter and when it is omitted, three categories are automatically used.

In the case where the profile limits are not provided by the consumer, they will be automatically constructed by MEC. To this purpose, the measurement scale of each QoS evaluation criterion included in the “find” SOAP message sent by the consumer to the registry is subdivided into three equal intervals. Then, profile limits are defined by joining the limits of these intervals on the different evaluation criteria.

The set of assignment examples are defined as follows. First, MEC generates a set of fictive compositions. Each fictive composition k_f is associated with a vector of m elements $(g_1(k_f), g_2(k_f), \cdots, g_m(k_f))$, where m is the number of QoS evaluation criteria. Evaluations $g_j(k_f)$ $(j = 1, \cdots, m)$ are defined such that k_f may be assigned to two succussive categories. For better explanation, consider

```
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema">
<xsd:complexType element name="infer">
 <xsd:sequence>
  <xsd:element name="sorting_data">
 </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="sorting_data">
   <xsd:element name="categories_number" type="xsd:positiveInteger">
   <xsd:element name="profiles">
   <xsd:element name="assignment_examples">
</xsd:complexType>
    ....
```

Fig. 3. XML schema of MEC request to W-IRIS

two categories C_i and C_j and let b_h be the profile limit between C_i and C_j with evaluation vector $(g_1(b_h), g_2(b_h), \cdots, g_m(b_h))$. Then, a fictive composition k_f is defined such that its performances on a subset of QoS evaluation permits to assign it to C_i and the rest permits to assign it to C_j.

XML schema of W-IRIS "inference_ouptut" SOAP message to MEC is given in Figures 3. It is a collection of preference parameters and the corresponding values. These parameters will be used by MEC to apply ELECTRE TRI.

```
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema">
<xsd:complexType element name="inference_output">
 <xsd:sequence>
  <xsd:group ref="preference_parametersGroup">
 </xsd:sequence>
</xsd:complexType>
<xsd:group name="preference_parametersGroup">
 <xsd:sequence>
   <xsd:element name="preference_parameter" type="preference_parameterType" minOccurs="1">
 <xsd:sequence>
</xsd:group>
<xsd:complexType name="preference_parameterType">
 <xsd:sequence>
  <xsd:element name="name"  type="token" #REQUIRED>
  <xsd:element name="value"  type="anyType"  #REQUIRED>
 </xsd:sequence>
</xsd:complexType>
```

Fig. 4. XML schema of W-IRIS response to MEC

5 Multicriteria evaluation component

The general schema of multicriteria evaluation component (MEC) is depicted in Figure 1. Basically, it takes as input a set of composite Web services and a set of QoS evaluation criteria and generates a set of recommended compositions.

The final choice should be performed by the consumer, based on the MEC recommendation. In the rest of the paper, $K = \{k_1, k_2, \cdots, k_n\}$ denotes a set of n potential composite Web services and $I = \{1, 2, \cdots, n\}$ denotes the indices of these services. The solution proposed to construct set K will be detailed in Section 6.

The set of QoS evaluation criteria to be used is extracted from the "qosDescription" data of the extended registry (see section 4). The set of evaluation criteria will be denoted by $F = \{g_1, g_2, ..., g_m\}$ in the rest of the paper. Theoretically, there is no limit to the number of criteria. We observe, however, that a large set increases the cognitive effort required from the consumer and a few ones do not permit to encompass all the facets of the selection problem.

The quantification of evaluation criteria permits to transform qualitative evaluation criteria into quantitative ones by assigning values to the qualitative data. This is useful for mostly of multicriteria methods based on weighted-sum like aggregation decision rules. The most used quantification method is the scaling one. The quantification process consists in the definition of a measurement scale as the one mentioned earlier and then to associate to each level of the scale a numerical value.

Once potential composite Web services are constructed and evaluation criteria are identified, the next step consists in the evaluation of all these composite Web services against all the evaluation criteria in F. The evaluation of a composite Web Service $k_i \in K$ in respect to criterion $g_j \in F$ is denoted $g_j(k_i)$. The matrix $[g_j(k_i)]$, $\forall i \in I$, $\forall j \in F$ is called the performance table. The computing of $g_j(k_i)$, $\forall i \in I$, $\forall j \in F$, will be dealt with in Section 7.

Most of multicriteria methods require the definition of a set of preference parameters. Two cases hold here: either the preference parameters are provided explicitly by the consumer and extracted from the "find" SOAP message to the registry; or inferred by W-IRIS based on the assignment examples equally extracted from the "find" SOAP message sent by the consumer.

The input for multicriteria evaluation step are the performance table and the preference parameters. The objective of multicriteria evaluation is to evaluate and compare the different compositions in K.

As signaled above the advanced multicriteria method ELECTRE TRI and four elementary methods (weighted sum, conjunctive and disjunctive rules, and the majority rule) are incorporated in the framework.

As underlined above, three types of recommendations are possible within the proposed framework. Based on the specifications of the consumer, one of the following results is provided to it: i) one or a restricted set of composite Web services; ii) a ranging of composite Web services from best to worst with eventually equal positions and iii) a classification of composite Web services into different pre-defined categories.

These three types of result correspond in fact to the three ways usually used to formalize multicriteria problems as identified by [6]: *choice*, *ranking* and *sorting*.

6 Constructing potential composite Web services

Definition 1 *A **Web service** S_i is a tuple (F_i, Q_i, H_i), where: i) F_i is a description of the service's functionality; ii) Q_i is a specification of its QoS evaluation criteria and iii) H_i is its cost specification.*

We assume that each Web service S_i has a unique functionality F_i. In turn, the same functionality may be provided by different providers. Let P_i be the collection of providers supporting functionality F_i of Web service S_i: $P_i = \{s_i^1, s_i^2, \cdots, s_i^{n_i}\}$ where n_i is the number of providers in P_i. A composite Web service is defined as follows.

Definition 2 *Let $S_1, S_2, \cdots, S_n$ be a set of n individual Web services such that $S_i = (F_i, Q_i, H_i)$ $(i = 1, \cdots, n)$. Let P_i be the collection of Web services supporting functionality F_i. Let $G = (X, V)$ be the composition graph associated with $S_1, S_2, \cdots, S_n$. A **composite Web service** k is an instance $\{s_1, s_2, \cdots, s_n\}$ of G defined such that $s_1 \in P_1$, $s_2 \in P_2$, $\cdots$, $s_n \in P_n$.*

To construct the set of potential compositions, we have incorporated two algorithms in the MEC. The first one, called `CompositionGraph` and is not given here which permits the construction of the composition graph.

The second algorithm, given hereafter, is `CompositionsConstruction` that generates the potential compostions. This algorithm proceeds as follows. First a tree T is constructed using `Construct_Tree`. The inputs for this procedure is the set of nodes X and the set of providers for each node in X: $P = \{P_1, P_2, \cdots, P_n\}$. The tree T is constructed as follows. The nodes of the ith level are the providers in P_i. For each node in level i, we associate the providers in set P_{i+1} as sons. The same reasoning is used for $i = 1$ to $n - 1$. The nodes of the $n - 1$th level is associated with the providers in P_n. Finally, a root r is added to T as the parent of nodes in the first level (representing the providers in P_1). Then, the set of nodes for each composition is obtained as an elementary path in T.

```
Algorithm CompositionsConstruction

INPUT: $G = (X, V)$: composition graph
       $P = \{P_1, P_2, \cdots, P_n\}$: providers
OUTPUT: $K$: potential compositions

$T \leftarrow$ Construct_Tree($X$,$P$)
$t \leftarrow 1$

WHILE $t <= \prod_{i=1}^{n} |P_i|$
    $X_t \leftarrow$ ElementaryPath($T$)
    $//X_t = \{s_1^{\cdot}, s_2^{\cdot}, \cdots, s_n^{\cdot}\}$
    FOR each $(S_h, S_k) \in V$
        $V_t \leftarrow (s_h^{\cdot}, s_k^{\cdot})$
    END_FOR
$k_t \leftarrow G_t = (X_t, V_t)$
$K \leftarrow K \cup k_t$
$t \leftarrow t + 1$
END_WHILE
```

The complexity of algorithm considered algorithm is $O(r_1 \times (r_2 + r_3))$ where $r_1 = |V|$ is the cardinality of V, $r_2 = \prod_{i=1}^{n} |P_i|$ is the number of compositions and r_3 is the complexity of `ElementaryPath`.

7 Evaluation of compositions

As defined earlier, a potential composition is an instance of the composition graph $G = (X, V)$. Each composition can be seen as collection of individual Web services. The evaluation provided by the UDDI registry are relative to these individual Web services. However, to evaluate and compare the different potential compositions, it is required to define a set of rules to combine the partial evaluations (i.e. in respect to individual Web services) to obtain partial evaluations that apply to the whole composition.

To compute the partial evaluations $g_j(k_i)$ $(j = 1, \cdots, m)$ of the different compositions k_i $(i = 1, \cdots, n)$, we need to define a set of m aggregation operators $\Phi_1, \Phi_2, \cdots, \Phi_m$, one for each evaluation criterion. The partial evaluation of a composition k_i on criterion g_j, $g_j(k_i)$, is computed as follows. It consists in applying a bottom-top scan on graph $G_i = (X_i, V_i)$ and to apply the aggregation operator Φ_j on each node. Algorithm `PartialEvaluation` below implements this idea. It runs on $O(r^2)$ where $r = |X|$ is the number of nodes in the composition graph. The valuation, in respect to criterion g_j, of a node $x \in X_i$, denoted $v_j(x)$, is computed as follows: $v_j(x) = \Phi_j[g_j(x), \Omega(\Gamma^+(x))]$.

Recall that $\Gamma^+(x)$ is the set of successors of node x. The operator Ω involves nodes on the same level and may be any aggregation operator such as `sum`, `product`, `max`, `min`, `average`, etc. The operator Φ_j implies nodes on different levels and vary according to the BPEL constructors associated with node x. It may be the `sum`, `product`, `max`, `min`, or `average`.

```
Algorithm PartialEvaluation

INPUT: $k_i = G_i(X_i, V_i)$: composition
       $\Phi_j$: aggregation operators
OUTPUT: $g_j(k_i)$: partial evaluation of $k_i$ on $g_j$

$L_r \leftarrow \{s \in X_i : \Gamma^+(s) = \emptyset\}$
$Z \leftarrow \emptyset$

WHILE $Z \neq X_i$
    FOR each $x \in L_r$
        $v_j(x) \leftarrow \Phi_j[g_j(x), \Omega(\Gamma^+(x))]$
        $Z \leftarrow Z \bigcup \{x\}$
    END_FOR
    $L_r \leftarrow \{s \in X_i : v_j(w)$ is computed $\forall w \in \Gamma^+(s)\}$
END_WHILE

$g_j(k_i) \leftarrow v_j(s)$ where $s$ is the root of $G_i$
```

In the following, we provide the proposed formula for computing $v_j(x)$ $(j = 1, \cdots, 4)$ for response time, availability, cost and security evaluation criteria, denoted g_1, g_2, g_3 and g_4, respectively. Evaluation criteria g_1 and g_3 are to be minimized while criteria g_2 and g_4 are to be maximized. The three first criteria are cardinal. The latter is an ordinal one.

First, we mention that the following formula apply for non-leaf nodes, i.e., $x \in X_i$ such that $\Gamma^+(x) \neq \emptyset$. For leaf nodes, i.e. $x \in X_i$ such that $\Gamma^+(x) = \emptyset$, the partial evaluation on a criterion g_j is simply $v_j(x) = g_j(x)$.

Response time (g_1) The response time of a non-leaf node x is computed as follows: $v_1(x) = g_1(x) + \max\{v_1(y) : y \in \Gamma^+(x)\}$ *or* $v_1(x) = g_1(x) \quad + \sum_{y\in\Gamma^+(x)} \pi(x,y)\, v_1(y)$

The first part of $v_1(x)$ applies for the <`flow`> or the sequential BPEL constructors. The second part applies when the constructor <`switch`> is used.

Availability (g_2) For the availability, two formulae may be applied for respectively the <`flow`> or the sequential constructors and the <`switch`> constructor: $v_2(x) = g_2(x)\ \prod_{y\in\Gamma^+(x)} v_2(y)$ *or* $v_2(x) = g_2(x)\ \sum_{y\in\Gamma^+(x)} \pi_(x,y)\ v_2(y)$

Cost (g_3) For cost criterion, two formula may be used for respectively the <`flow`> or the sequential constructors and the <`switch`> constructor:
$v_3(x) = g_3(x) + \sum_{y\in\Gamma^+(x)} v_3(y)$ *or* $v_3(x) = g_3(x) + \sum_{y\in\Gamma^+(x)} \pi(x,y)\ v_3(y)$

Security (g_4) Finally, for security criterion, we have:
$v_4(x) = \min\{g_4(x), \min_{y\in\Gamma^+(x)}\{v_4(y)\}\}$

8 Conclusion

We have proposed a framework for composite Web services selection based on multicriteria evaluation. The framework extends the Web services architecture by adding a new Multicriteria Evaluation Component (MEC) devoted to multicriteria evaluation. This additional component takes as input a set of composite Web services and a set of evaluation criteria. The output is a set of recommended composite Web services. We also proposed solutions to construct and evaluate the different potential compositions. To show the feasibility of our proposal, we have developed a prototype based on the jUDDI registry.

There are several directions for future research. A first point to investigate is related to the extension of the framework to support dynamic composition. The basic change concerns essentially the construction of the potential compositions and their evaluations.

References

1. Y. Ma and C. Zhang. Quick convergence of genetic algorithm for QoS-driven Web service selection. Computer Networks. 2008. 52.
2. V. Mousseau and L. Dias. Valued outranking relations in ELECTRE providing manageable disaggregation procedures. European Journal of Operation Research. 2004. 156. 467-482
3. S. Dustdar and W. Schreiner. A survey on web services composition. International Journal and Grid Services. 2005. 1. 1. 1-30.
4. S. Dustdar and W. Schreiner. A survey on web services composition. International Journal and Grid Services. 2005. volume = 1. 1. pp. 1-30.
5. J. Figueira and S. Greco and M. Ehrgott. Multiple Criteria Decision Analysis: State of the Art Surveys. SVer. 2005. NY.
6. B. Roy. Multicriteria methodology for decision aiding. Kluwer Academic Publishers. 1996. Dordrecht.

7. D.C. Schmidt and D.L. Levine and S. Mungee. The design and performance of the TOA real-time object request broker. Computer Communications. 1998. 21.4. pp. 294-324.
8. E.M. Maximilien and M.P. Singh. Conceptual model of Web services reputation. ACM SIGMOD Record. 2002. 31. 4. pp. 36-41.
9. D.A. Menascé. Composing Web servies: A QoS view. IEEE Internet Computing. 2004. December 2004.
10. D.A. Menascé and V. Dubey. Utility-based QoS brokering in service oriented architectures. 2007.
11. D.A. Menascé and H. Ruan and H. Gomma. QoS management in service oriented architectures. Performance Evaluation Journal. 2007. 64. 7-8. pp. 646-663.
12. B. Benatallah and M. Dumas and Q.Z. Sheng and A. Ngu. Declarative composition and peer-to-peer provisioning of dynamic Web services. Proc. of ICDE'02, IEEE Computer Society. 2002. 297-308. San Jose.
13. J. Klingemann. Controlled flexibility in workflow management. Proc. of the 12th International Conference on Advanced Information Systems (CAiSE). 2000. pp. 126-141. Stockholm, Sweden.
14. D. Georgakopoulos and H. Schuster and A. Cichocki and D. Baker. Managing process and service fusion in virtual enterprises. Information System. 1999. 24. 6. pp. 429-456.
15. F. Casati and S. Ilnicki and L.-J. Jin and V. Krishnamoorthy and M.-C. Shan. eFlow: a platform for developing and managing composite e-services. HP Laboratoris. 2000. Technical Report. HPL-2000-36. Palo Alto.
16. L. Zeng and B. Bentallah and M. Dumas and J. Kalagnanam and Q.Z. Sheng. Quality driven web service composition. Proc. of the 12th international conference on World Wide Web. 2003. Budapest Hungray. May 20-24. pp. 411-421. ACM Press, New York, NY, USA.
17. E.M. Maximilien and M.P. Singh. A Framework and ontology for dynamic Web services selection. IEEE Internet Computing. 2004. 8. 5. pp. 84-93.
18. J. Klingemann and J. Wasch and K. Aberer. Deriving service models in cross-organizational workflows. Proc. Ninth International Workshop on Research Issues in Data Engineering: Virtual Enterprise (RIDE-VE'99). 1999. pp. 100-107. Sydney, Australia.
19. S. Youcef and U. Bhatti and L. Mokdad and V. Monfort. Simulation-based response-time analysis of composite Web services. Proc. 10th IEEE international Multitopic conference (INMIC'06). 2006. pp. 349-354.
20. J. Cardoso. Quality of service and semantic composition of workflows. University of Georgia. 2002.
21. M. Champion and E. Newcomer and D. Orchard. Web service architecture. 2000. W3C Draft.
22. S. Dustdar and W. Schreiner. A survey on Web services composition. International Journal of Web and Grid Services. 2005. 1. 1. pp. 1-30.
23. L.C. Dias and V. Mousseau. 2003. IRIS: A DSS for multiple criteria sorting problems. MCDA. 12. pp. 285-298.
24. A. Makhorin. GNU linear programming kit reference manual version 4.8. Department for Applied Informatics, Moscow Aviation Institute, Moscow, Russia. 2005. 84 pages.
25. D.A. Menascé. 2002. QoS Issues in Web Services. IEEE Internet Computing. 6. 6. pp. 72-75.

26. Danilo Ardagna and Barbara Pernici. Dynamic web service composition with QoS constraints. International Journal of Business Process Integration and Management 2006 - Vol. 1, No.4 pp. 233 - 243