A modified Logic Scoring Preference method for dynamic Web services evaluation and selection

Hong Qing Yu¹ and Hernán Molina²
¹ Department of Computer Science, University of Leicester, UK
  hqy1@mcs.le.ac.uk
² Department of Informatics, School of Engineering, University of La Pampa, Argentina
  hmolina@ing.unlpam.edu.ar

Abstract. The Logic Scoring Preference (LSP) method extends existing scoring techniques and provides a means for the development of complex criterion functions using continuous preference logic. It has been successfully used to evaluate hardware system, software system and websites. However, the current LSP method is not sufficient to help evaluating and selecting web services in the context of dynamic service composition. This area depends not only on static service functional and non-functional requirements (which are traditionally considered in LSP applications), but also on run-time context such as business policies and pre/succeeding service instances in workflow patterns. Thus, we propose a modified LSP method to target the problem of dynamic web service evaluation and selection.

1 Introduction

Web services technology is widely used by industry today. With increasing number of web services available, there is in general more than one service to fulfill a given task. Consequently, the biggest challenge is to choose the most appropriate service which satisfies not only functional requirements but also non-functional, context related requirements, such as policies, the business’ target market, QoS (Quality of Service) and existing system workflow patterns.

In this paper, we propose a modified Logic Scoring Preference (LSP) [1] method to achieve dynamic web service evaluation and selection. The remaining content is structured as follows. We conclude the introduction with a motivating example. The simple introduction of LSP method is described in section 2. Our modified LSP method is introduced in section 3. A worked evaluation example and related work comparing are shown in section 4. Section 5 concludes the paper and highlights further work.

1.1 Motivating Example

Considering a scenario (shown in Fig. 1), where a business organization needs a payment service to complete an online product selling business process, some context aspects should be considered. Firstly, the market aspect, the target customers might be
at home or traveling. In the case that they are traveling, the customers could make use of several devices, such as a laptop, PDA, landline, desktop or mobile phone. Secondly, the workflow aspect requires that the protocols of the payment service are suitable for integrating to existing online selling service. Thirdly, the QoS aspects, such as security must be high; performance rate should be reasonable and privacy should be respected. Fourthly, the policies include that customers are supposed to understand one language out of English, Spanish and French. Moreover, the service provider must be located in Europe and a lower transaction fee is better. Finally, being able to accept more types of bank card such as Visa, MasterCard, Solo and Switch is preferable. There are four services available which can functionally fulfill the payment task shown in Fig. 1. Now the question is which one is the most suitable for use.

![Fig. 1: Motivating Example](image)

2 The LSP method

LSP is a quantitative method based on scoring techniques and a continuous preference logic [2]. Basically, the method allows establishing an evaluation criterion by specifying the expected properties of a system. To each one of these properties a criterion function is assigned. These functions transform specific domain values to a normalized scale indicating the degree of satisfaction of the corresponding preference. Then, all preference values can be properly grouped using a stepwise aggregation structure to yield a global preference. This can be achieved by means of a preference aggregation function, called generalized conjunction/disjunction or andor, combining weighted power means to obtain the global preference $e_0$ as in:

$$e_0 = \left( W_1 e_1^r + ... + W_k e_k^r \right)^{1/r}, W_1 + ... + W_k = 1$$

(1)

Where the power $r$ can be suitably selected to obtain desired logical properties (see [3, 2] for further details). The LSP decision method has been applied in the evaluation and selection of hardware and software systems [3, 1], as well as in the evaluation and comparison of web applications [4]. The strong drawback of these applications is that
they require the participation of human expert interactions, which is not suitable for working in a dynamic environment as we discussed before.

3 The modified LSP method (MLSPM)

3.1 Context model

The first step of our method is to model the dynamic context environment and gather useful information such as preference elements and their values from a well defined model (Context model). Defining a context model actually is an analysis process that needs to understand the purposes, scales, background information and the activity rules, etc. It is very difficult to build a unified model which includes all aspects and their attributes. In this work, we use a context model which is just an expression model for showing how our dynamic method works for our web service evaluation and selection purpose (see Fig. A.1 in appendix A).

In this paper, we are not going to illustrate the detailed techniques of context reasoning and mining addressed in the “inContex” project [5]. However, we assume the model is implemented by XML technology and a well defined mediator can be used to resolve mismatches between different terminologies (data level).

3.2 Specification of Type-based Unified evaluation methods

We can design different evaluation metrics based on the given context model types. For example device elements may be described as value=“computer, PDA”, type=“string set overlap”. In our current work, we have identified four different evaluation functions to capture the variety of types. The four functions are: “exact match” (equation 3), “set overlap” (equation 4), “level match” (equation 5) and “specific value” (equation 6).

Typical usage is linked to the data type of the context aspect: if the context aspect can be expressed by a Boolean, then exact match would be used; considering sets of information, set overlap is useful; level match is useful for ordered discrete values (such as low, medium and high); and finally specific value allows for complex functions that calculate a numerical value (e.g. for distance functions). Because of the link to the data type, it can be automatically determined which function should be used. In addition, the weight $\omega < 0$ expresses the lower value is desired or $\omega \geq 0$ means higher value is respect. Thus the global preferences evaluation function is changed from equation (1) to (2).

\[
e_{0} = \left( |\sum_{i=1}^{n} \omega_{i} | E_{1}^{i} + |\sum_{i=1}^{n} \omega_{i} | E_{2}^{i} + \ldots + |\sum_{i=1}^{n} \omega_{i} | E_{n}^{i} \right)^{1/n} \text{ with } 0 \leq E \leq 1, \sum_{i=1}^{n} |\omega_{i}| = 1
\]

The respective formulas to compute values for these functions (E1 to En) are as follows:

\[
E = \begin{cases} 
1 & \text{if criterion is met} \\
0 & \text{otherwise} 
\end{cases}
\]

\[
E = \left( e_{1} + e_{2} + \ldots + e_{n} \right) / n \text{ with } e_{i} \text{ being a score for each element of the set}
\]
where \( i \) is the number of levels and \( i_0 \) is the current level match \( E = \frac{i}{i_0} \)

\[
E = \begin{cases} 
1 - \frac{v_{\min} - v}{v_{\min} - v_{\max}} & \text{iff } \omega \geq 0 \\
\frac{v_{\min} - v}{v_{\min} - v_{\max}} & \text{otherwise}
\end{cases}
\]

\( v_{\min} \) being the minimum value for all services, \( v_{\max} \) the maximum value and \( v \) the value for the current service in (6). For the motivating example, we match evaluation functions to each preference as in Table A.1 (see Appendix A).

### 3.3 Global aggregation

We need a global preference aggregation function \( L(g_1(a_1),...,g_n(a_n)) \) to calculate all aspects of preferences. The function itself must reflect specific requirements and logic conditions, such as simultaneity, replaceability and others [2]. The function \( g \) is one of the individual evaluation methods which were discussed in the previous section. We use the conjunctive partial absorption function as global aggregation structure (see Fig. 2) [3]. We separate preferences into a critical group (\( EP^C \)) and a desired group (\( EP^D \)). The critical group presents all mandatory requirements; the desired group takes all the other preferences.

**Fig. 2:** The structure of the conjunctive partial absorption aggregation function

Behaviors of the conjunctive partial absorption function are such that the global preference value (denoted by \( GP \)) will be 0\(^1\) when any of the critical preferences are not satisfied, in which case the service will be discarded. On the other hand, a web service that satisfies all critical preferences will be valuated to a non-zero value, from which the degree of satisfaction of the desired preferences can raise or reduce the final global preference. The first function (denoted by a circle named DAC) is dynamically selected by an automated calculate method (ACM) to reflect dynamic preferences. The second function (denoted by a circle named CA) is always there to achieve the desired absorption behaviors.

\(^1\)In \( f(x) = x^r \), When \( r < 0 \) and \( x = 0 \), then an error occurs, we use 0 to express it.
We design an automated calculate method (ACM) to find a logic correctness GCD (Generalized Conjunction/disjunction) function based on Continuous Logic [6] for the desired preferences without considering critical preferences. Thus, Fig. 2 shows that all weights of desired preference sum up to 1. The logic meaning can be reflected by meaningful weights. We consider the value of the weight \( \omega_i \) belongs to a set \( A, A \in (0,1) \). Then we have an ordered set \( W = (\omega_1, \ldots, \omega_n) \), and \( \omega_1 \geq \ldots \geq \omega_n \). Based on the meaning of or-ness in OWA decision making method [7], we can get the following function:

\[
\lambda_{\text{orness}} = \frac{1}{n-1} \sum_{i=1}^{n-1} (n-i)\omega_i, \omega_i \text{ is the } i\text{th place in set } V
\] (7)

Where \( V \) is reordered set of \( W \), and the reorder algorithm firstly is to find the same weights \( \omega_{1+1}, \ldots, \omega_{1+i} \) value to \( \omega_1 \). Put \( \omega_{1+1}, \ldots, \omega_{1+i} \) to the tail of the set. Secondly, taking \( \omega_{2+1}, \ldots, \omega_{2+i} \) which have the same value of \( \omega_2 \) in the new set in front of the \( \omega_{n-1}, \ldots, \omega_n \). The rest of the refreshed set repeats second step until the last element that has not been reordered before. For example, if \( W = \{0.2, 0.2, 0.15, 0.15, 0.1, 0.1, 0.2\} \), then, \( V = \{0.2, 0.15, 0.1, 0.15, 0.1, 0.1, 0.2\} \) the value \( \lambda \) presents the degree of the “or-ness” by equation (7) calculated. The mapping table between the value \( \lambda \) and the GCD operators is given in Table A.2 of Appendix A.

4 Worked example and related work

Applying our method to the motivating example, we can consider that the devices preference (0.25), performance rate (0.5), privacy (0.01), cost (-0.04) and bank cards (0.2) requirements are desired preferences. Communication protocol (0.1), security (0.2), location (0.1) and language (0.1) are critical preferences. First we calculate the “or-ness” position based on equation (7).

\[
\lambda = \frac{1}{4} (0.5 \times 4 + 0.25 \times 3 + 0.20 \times 2 + 0.04) = 0.7985
\]

Comparing to the appendix A, the result shows D-+ is a suitable GCD function to be selected (power \( r = 3 \)). According to equation function (2), we got following values for the four competitive services:

\[
e_1 = 0.201 (S1), e_2 = 0.126 (S2), e_3 = 0.181 (S3) \text{ and } e_4 = 0.276 (S4)
\]

The next stage is to use the conjunctive partial absorption aggregation function to determine which service is best to use. We take all critical evaluation values and output values from last step of each service into the CA operator (power \( r = 0.72 \)), and then we get the final evaluation values by using function (2) again:

\[
e_1 = 0.333, e_2 = 0, e_3 = 0 \text{ and } e_4 = 0.493 \text{ the returned ranked list is } \{S4,S1\}.
\]

Comparing to related work [8] and [9], our method has 5 outstanding advantages. (1) MLSPM combines evaluation and selection activities together contrast to [8] and [9] which only address selection issues. (2) MLSPM can deal with all types of preferences such as String and Boolean. However, the other works only focused on
numbering type. (3) Our method is more dynamic and easier. (4) The context model we defined covers more evaluation aspects, such as policy rules, workflow requirements beyond QoS. (5) The key feature of LSP method is considering the logic relations between preferences rather than just simply using weight mechanism.

5 Conclusion and future work

In this paper we outlined a modified LSP method that allows us to dynamically evaluate and select the most suitable web services considering large number of available services, changing contexts and run-time service requests in a timely and efficient manner. In addition, we gave an example scenario in which our LSP method was well applied.

Future work will cover the definition of the meanings of weights used in this paper from the perspective of user preferences, context mining and reasoning techniques since their outcomes will be the inputs for web service evaluation and selection. Additionally, implementation issues of the modified LSP method, as well as related mechanisms will be addressed.

6 Acknowledgment

We thank Dr. Stephan Reiff-Marganiec and Dr. Luis Olsina for their comments and suggestions. This work is also supported by inContext (Interaction and Context Based Technologies for Collaborative Teams) project: IST IST-2006-034718 and WEE-NET (Web Engineering Network of Excellence) project: ALFA II-0359-FA.

References:

Appendix A

![Fig. A.1 context model](image)

<table>
<thead>
<tr>
<th>Desired preferences</th>
<th>Evaluation methods</th>
<th>Critical preferences</th>
<th>Evaluation methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance rate</td>
<td>(6)</td>
<td>Protocol</td>
<td>(3)</td>
</tr>
<tr>
<td>Devices</td>
<td>(4)</td>
<td>Security</td>
<td>(3)</td>
</tr>
<tr>
<td>Privacy</td>
<td>(5)</td>
<td>Location</td>
<td>(3)</td>
</tr>
<tr>
<td>Cost</td>
<td>(6)</td>
<td>Language</td>
<td>(4)</td>
</tr>
<tr>
<td>Bank cards</td>
<td>(4)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table A.1:** Assigned evaluation methods to example

<table>
<thead>
<tr>
<th>Value of $\gamma$</th>
<th>GCD Operator symbols</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\gamma &lt; 0.3333$</td>
<td>GEO</td>
<td>Geometric mean</td>
</tr>
<tr>
<td>0.3333 ≤ $\gamma$</td>
<td>C-</td>
<td>Weak QC</td>
</tr>
<tr>
<td>0.3750 ≤ $\gamma$</td>
<td>C--</td>
<td>Weak QC (-)</td>
</tr>
<tr>
<td>0.4375 ≤ $\gamma$</td>
<td>A</td>
<td>Arithmetic mean</td>
</tr>
<tr>
<td>0.5000 ≤ $\gamma$</td>
<td>D--</td>
<td>Weak QD (-)</td>
</tr>
<tr>
<td>0.5625 ≤ $\gamma$</td>
<td>SQU</td>
<td>Square mean</td>
</tr>
<tr>
<td>0.6232 ≤ $\gamma$</td>
<td>D-</td>
<td>Weak QD</td>
</tr>
<tr>
<td>0.6250 ≤ $\gamma$</td>
<td>D-+</td>
<td>Weak QD (+)</td>
</tr>
</tbody>
</table>

**Table A.2:** The range of selection GCD operators [3]