

LASS - License Aware Service Selection: Methodology and Framework

G.R. Gangadharan¹, Marco Comerio², Hong-Linh Truong³,
Vincenzo D'Andrea⁴, Flavio De Paoli², Schahram Dustdar³

¹ Telematica Institute, Enschede, The Netherlands
gr@telin.nl

² University of Milano - Bicocca, Milano, Italy
{comerio, depaoli}@disco.unimib.it

³ Vienna University of Technology, Vienna, Austria
{truong, dustdar}@infosys.tuwien.ac.at

⁴ University of Trento, Trento, Italy
dandrea@disi.unitn.it

Abstract. A service provider defines individual services with corresponding service licenses which consumers should follow. Often, service consumers are interested in selecting a service based on certain licensing clauses. For a set of requested licensing clauses by a consumer, there can be several licenses that differ in the set of offered license specifications. Thus, a license aware service selection process includes the discovery of a set of services that meets certain functional parameters and, in addition, the process evaluates these services in order to identify the ones that fulfill a set of license specifications as requested by a consumer. In this paper, we present a methodology and framework for service selection process, based on matching the offered licensing specifications by providers with the requested licensing specifications by consumers⁵.

1 Introduction

Selecting different services to fulfill a consumer's need is a fundamental issue that has attracted much research efforts. Most existing works concentrate on developing selection techniques based on functional properties (FPs) and non-functional properties (NFPs). However, the selection of a service usage is also dependent on other clauses, such as scope of rights and warranties, that are important in deciding whether a service should be used. One of the relevant issues from this perspective is the role of service licensing in service selection. A service license includes all transactions between the licensor and the licensee, in which the licensor agrees to grant the licensee the right to use and access the service under predefined terms and conditions. Various aspects of service licensing are described in [1].

⁵ This work is partially supported by the IST COMPAS project, funded by the European Commission, FP7-ICT-2007-1 contract number 215175.

We propose to extend the traditional FPs and NFPs based service selection process with an additional selection based on licenses. In our previous work on license-aware selection [2], we discussed steps in license-aware service selection and introduced the ranking of services by applying degree and distance indicators to scope of rights and financial terms in service licenses. This paper proposes an extension of the work discussed in [2] by introducing an approach for enhancing service selection with a novel process based on matching the offered licensing specifications with the requested licensing specifications that takes into account not only *Scope of Rights* and *Financial Terms* but also *Warranty*, *Indemnity*, *Limitation of Liability* and *Evolution* clauses. We also introduce in detail the steps in license-aware service selection, including selection algorithm, and present a framework realizing the service selection process.

2 License Aware Service Selection (LASS) Methodology

The LASS methodology includes the phases of *Matching Evaluation* and *Filtering*. The *Matching Evaluation* phase computes two different indicators (*Degree* and *Distance*) between the clauses of offered license specifications and the clauses of requested license specifications. The *Degree* shows if an offer matches a request and is expressed by a value in the range [0..1], where 0 means ‘no match’ and 1 means ‘exact match’. The *Distance* indicator is used to capture additional information about the matching. In the case of ‘exact match’, it points out how much the offer dominates the request ($Distance \geq 0$) and in the case of ‘no match’, how much the offer is far from satisfying the request ($Distance \leq 0$).

Evaluating the values for license specifications involves the following kinds of data: (i) *Scope of Rights* clauses expressed as distinct values (e.g., *adaptation*, *composition*, and *derivation*); (ii) *Financial Terms* and *Warranty* clauses expressed in a range of values; (iii) *Indemnity* and *Limitation of Liability* clauses expressed as a set of qualitative values; (iv) *Evolution* clauses expressed as an integer values.

The evaluation of *Degree* and *Distance* indicators for *Scope of Rights* and *Financial Terms* is explained in [2].

Warranty clauses can be expressed by consumers using the constraint operators \geq or \leq (for example, amount ≤ 2 Euros) or an interval of values (for example, $97\% \leq$ availability rate $\leq 99\%$). The service provider usually publishes a service license with the specifications of the offered *Warranties* expressed as an interval of values. The *Degree* and *Distance* indicators are evaluated according to the constraint operators of the requested license specifications. Examples of these formulas are discussed in [3].

A service license also specifies *Indemnity* and *Limitation of liability* clauses. *Indemnity* clauses specify the provision of defense by the licensor to the licensee. An example is the *Third Party Infringements Claims* clause that represents the statement provided by the licensor to the licensee to protect against the claims of a third party if any infringements over the intellectual property rights arise. *Limitation of liability* clauses limit the liability of the licensor and the

licensee under the license agreement. An example is the *Non-Network Errors* clause specifying that the licensor will not be liable if any problem with the network occur during the service provisioning. For *Indemnity* and *Limitation of liability* clauses the *Degree* is set to 1 if the requested clause matches exactly with the offered one and it is set to 0 elsewhere. These clauses differ from the ones on *Scope of Rights* because subsumption cannot be used. For this reason, the *Distance* is always set to 0.

The last kind of data that our methodology is able to manage is *Evolution* clauses that specify the modifications on the license by future releases or versions of the service. Examples are the *Maximum Upgrades* and the *Maximum Versions* clauses that indicate the allowed number of upgrades and versions of the service before the license becomes invalid. As these clauses are expressed as a fixed integer number, the *Degree* is equal to 1 if the requested value is \leq than the offered value and it is set to 0 elsewhere. The *Distance* indicator is evaluated subtracting the requested value from the offered value.

The second phase of the LASS methodology, consists of three activities. The first activity is in charge of discarding unsuitable services and starts considering the *Degree* evaluated for each license clause. A service whose license has a clause with *Degree* equal to 0 is discarded and no longer considered in the selection process. Thus, the number of candidate services in the set of functionally matched services can be reduced. If all services are discarded (no services are able to satisfy the requested license specifications), a service with a license closer to a consumer's requested specifications will be recommended. The service recommended to a consumer is selected based on the *Distance* indicator. The consumer can accept or deny the recommended service (as not exactly satisfying the requested license specifications). The second activity of the *Filtering* phase is in charge of evaluating net indicators for each license, which is a distinguishing characteristic of our approach. The *Net Degree* provides information about how much an offered license matches a requested license. The *Net Distance* provides additional information about how the required clauses are matched. Details about how to calculate *Net Degree* and *Net Distance* are described in [2].

Finally, in the *Filtering* phase the list of services is sorted according to their *Net Degree* values. If two or more services have equal *Net Degree* value, their *Net Distance* values are considered for ranking.

Service license selection algorithm is listed in Algorithm 1. The inputs provided by a consumer are a set of requested functionalities F and the requested license clauses lc .

Let $F = \{f_1, f_2, \dots, f_n\}$ denotes a set of functional parameters. Functional parameters, specified by consumers, represent the requested operations performed by services. For each f_i , we assume that there exists a category of services, t_i , that offers the functionality specified by f_i .

Let $T = \{t_1, t_2, \dots, t_n\}$ denotes categories of services associated to F , where t_i provides the functionality required by f_i . Given a $t_i \in T$, there exists many services belonging to this service type, each offering the functionality f_i but with

possibly different implementations and associated licenses. We denote this set of services with $S(t_i) = \{S_1, S_2, \dots, S_m\}$.

Let $\Lambda(S(t_i)) = \{L(S_1), L(S_2), \dots, L(S_n)\}$ be the set of licenses in which $L(S_i)$ indicates the license associated with the service S_i . Let lc be the requested license specifications that need to be considered along the selection process. Let $\Upsilon(t_i)$ be a set of filtered services having the requested license clauses that match with the offered license specifications. Our objective is to select a service license in $\Lambda(S(t_i))$ that best matches lc .

Algorithm 1 Service License Selection

```

1: for all  $t_i \in T$  do
2:   for all  $S_j \in S(t_i)$  do
3:      $Degree(L(S_j)) \leftarrow DgEval(lc, L(S_j))$ 
4:      $Distance(L(S_j)) \leftarrow DsEval(lc, L(S_j))$ 
5:   end for
6:    $\Upsilon(t_i) \leftarrow Filter(S(t_i))$ 
7:   Recommend a service closer to  $lc$  if  $\Upsilon(t_i) = \phi$ 
8:    $\Psi(S(t_i)) \leftarrow PsiEval(\Upsilon(t_i))$ 
9:    $\Delta(S(t_i)) \leftarrow DeltaEval(\Upsilon(t_i))$ 
10:   $\Xi(t_i) \leftarrow Rank(\Psi(S(t_i)), \Delta(S(t_i)))$ 
11: end for

```

The algorithm starts (lines 3-4) with the evaluation of the clauses. *Degree* and *Distance* indicators for a requested clause specified in lc and an offered clause specified in $L(S_j)$ are evaluated as described previously. Line 6 discards services having offered license specifications that do not match any requested clauses. The services that are not discarded are saved in $\Upsilon(t_i)$. The algorithm proceeds checking if the set of filtered services is empty (line 7). The emptiness of this set indicates that $S(t_i)$ does not contain services that are able to satisfy the requested license specifications mentioned in lc . In this case, a service closer to the request is recommended. The consumer can decide to terminate the process or to accept the proposed service (changing his/her license specifications). The algorithm proceeds (lines 8-9) evaluating *Net Degree* (Ψ) and *Net Distance* (Δ) in order to link each services in $\Upsilon(t_i)$. The services in $\Upsilon(t_i)$ are ordered based on Ψ and Δ . A set of ranked services are placed in $\Xi(t_i)$ (line 10).

3 License Aware Service Selection (LASS) Framework

LASS framework supports the selection of services based on licensing specification in addition to performing service selection with functional and non-functional properties. Based on the FPs, NFPs, and licensing specifications by a service consumer, the LASS framework selects a service that best matches with the requested specifications. The LASS framework tries to find if any service advertisements given by providers match the request of consumers in service

functionality at first, then followed to match the specified NFPs. There may be always the possibility of more than one services, offering similar functionality that differ in their licenses. Figure 1 depicts the LASS framework which comprises of the following components.

- *User Interface*: supports consumers to specify FPs, NFPs, and license clauses based on which services would be selected.
- *Service Selection Request Handler*: receives FPs, NFPs, and license specifications from consumers.
- *FP/NFP Selector*: discovers a set of services satisfying the required functional and non-functional parameters. The techniques for selecting web services based on FPs and NFPs are not the focus of this paper and they are built on well-established works and considered as plug-ins of the LASS framework.
- *License Selector*: discovers a set of services based on matching the offered licensing specifications of functionally matching services against the requested licensing specifications. This component includes the algorithm introduced in this paper.
- *Service Information*: an XML-based repository where information associated with services are stored. In our framework, we utilize SEMF [4] which is able to manage different types of web service related information, including license and QoS.

For a given functionality (expressed by a set of functional parameters), NFPs, and a set of requested license specifications, the framework performs license-aware selection of services in the following two steps.

1. A set of services are retrieved that match with functional parameters and NFPs.
2. These set of services having offered license specifications are filtered in order to retrieve a set of services that satisfy the requested license specifications.

4 Related Work and Discussions

The increasing availability of services that offer similar functionalities requires to enhance the traditional functionality-based service selection process [5, 6] with an additional selection phase that identifies the services that better fulfill a set of NFPs requested by the actual user.

The selection of services based on non-functional specifications has been studied intensively by the research community. Several approaches are based on semantically rich descriptions of non-functional parameters. The approach proposed in [7] represents a solution for matching NFPs of web services represented using WS-Policy. In [8], declarative logic-based matching rules and optimization methods are applied for optimal service selection. A dynamic web service selection based on semantic interpretation of offered service capabilities and the parameters specifying the actual request is proposed in [9].

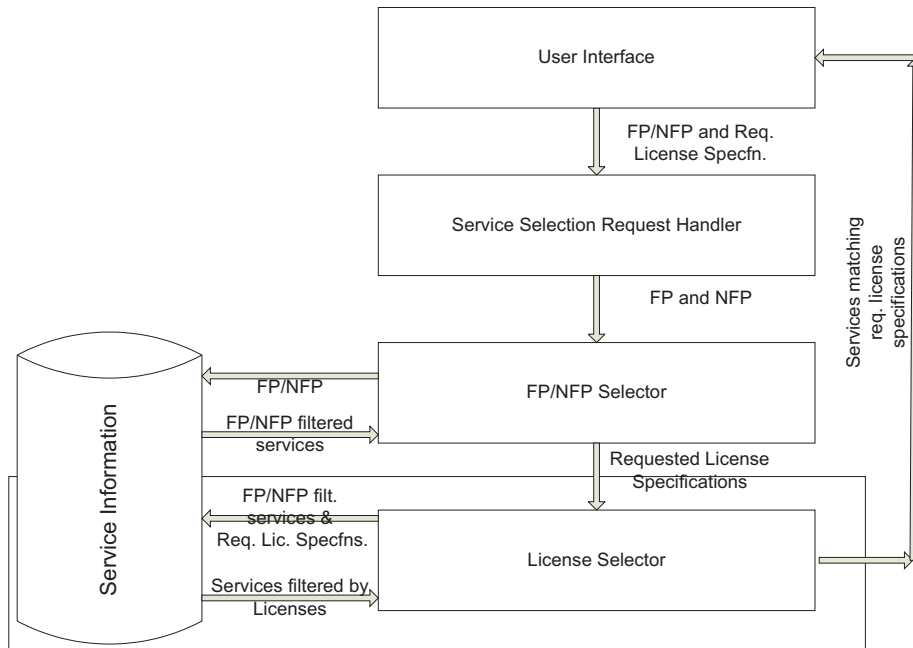


Fig. 1. License aware service selection (LASS) framework

There are also several proposals to address service selection without the use of semantics. A modified logic scoring preference method of evaluating non-functional aspects is proposed in [10]. A framework supporting brokers in selecting web services based on the required QoS for autonomic grid environments is proposed in [11].

In our earlier work [3], we have presented a semantic approach for selection of services by evaluating both qualitative and quantitative NFPs. In this paper, we have extended our approach described in [3] that can be used for selection of services whose descriptions are not semantic (as in the case of ODRL-S descriptions). Moreover, we evaluate the degree of match when service offers are expressed as interval of values, thereby overcoming a limitation of [3]. This makes the process of service selection more realistic allowing the description of NFPs that can assume any value in the given interval.

To the best of our knowledge, there exists no work on selection of services based on their license specifications. Google⁶ and Yahoo!⁷ search engines provide advanced options to retrieve contents based on requested licenses. However, these options restrict consumers with limited specifications of licenses.

⁶ http://www.google.com/advanced_search?hl=en

⁷ <http://search.yahoo.com/web/advanced?ei=UTF-8>

5 Concluding Remarks

Being a way to manage the intellectual rights between service consumers and service providers, licenses are critical to be considered in services. In this paper, we have illustrated a novel methodology for selection of services by matchmaking of license clauses requested by a consumer and offered by several providers. Our LASS methodology describes the selection of services based on all possible clauses of licenses. Following the LASS methodology, we have presented the LASS framework that integrates the process of license aware service selection with functional and non-functional properties. We are currently extending the framework to allow for specifying multiple functionalities, each functionality with differing license specifications and then finding a composite service associated with a composite service license that meets the requested license specifications.

References

1. Gangadharan, G.R., Weiss, M., D'Andrea, V., Iannella, R.: Service License Composition and Compatibility Analysis. In: Proceedings of the International Conference on Service Oriented Computing (ICSOC'07) . (2007)
2. Gangadharan, G.R., Comerio, M., Truong, H.L., D'Andrea, V., De Paoli, F., Dustdar, S.: License-aware Service Selection. In: Proc. of the IEEE Conf. on Enterprise Computing, E-Commerce and E-Services (EEE'08). (2008)
3. Comerio, M., De Paoli, F., Maurino, A., Palmonari, M.: NFP-aware Semantic Web Services Selection. In: Proceedings of the 11th IEEE International Enterprise Distributed Object Computing Conference (EDOC). (2007)
4. Treiber, M., Truong, H.L., Dustdar, S.: SEMF - Service Evolution Management Framework. In: Proc. of the 34th Euromicro Conf. on Software Engineering and Advanced Applications. (2008)
5. Brogi, A., Corfini, S.: Behaviour-aware Discovery of Web service Compositions. *International Journal of Web Services Research* 4(3) (2007)
6. Aversano, L., Canfora, G., Ciampi, A.: An Algorithm for Web service Discovery through their Composition. In: Proceedings of the IEEE International Conference on Web Services (ICWS). (2004)
7. Verma, K., Akkiraj, R., Goodwin, R.: Semantic Matching of Web Service Policies. In: Proc. of the Second Intl. Workshop on Semantic and Dynamic Web Processes. (2005)
8. Lamparter, S., Ankolekar, A., Studer, R., Grimm, S.: Preference-based Selection of Highly Configurable Web Services. In: Proc. of the 16th Intl. Conf. on World Wide Web. (2007)
9. Friesen, A., Namiri, K.: Towards Semantic Selection for B2B Integration. In: Proceedings of the 12th International Conference on Web Engineering. (2006)
10. Reiff-Marganiec, S., Yu, H.Q., Tilly, M.: Service Selection based on Non-Functional Properties. In: Proceedings of the NFPSLA-SOC Workshop (ICSOC'07). (2007)
11. Anselmi, J., Ardagna, D., Cremonesi, P.: A QoS-Based Selection Approach of Autonomic Grid Services. In: Proceedings of the 2007 workshop on SOCP. (2007)