Non-functional Property based service selection: A survey and classification of approaches

Conference or Workshop Item

Non-functional property based service selection: A survey and classification of approaches

Hong Qing Yu and Stephan Reiff-Marganiec

University of Leicester, Computer Science Department, Leicester, UK
{hqy1,srm13}@mcs.le.ac.uk

Abstract. In recent years there has been much effort dedicated to developing approaches for service selection based on non-functional properties. It is clear that much progress has been made, and by considering the individual approaches there is some overlap in functionality, but obviously also some divergence. In this paper we contribute a classification of approaches, that is, we define a number of criteria which allow to differentiate approaches. We use this classification to provide a comparison of existing approaches and in that sense provide a survey of the state of the art of the field. Finally we make some suggestions as to where the research in this area might be heading and which new challenges need to be addressed.

Key words: Selection Approaches, Classification, Survey

1 Introduction

Service-oriented computing (and its predominant incarnation as web services) is reaching a certain maturity, which is reflected in the number of services that are becoming available. Established technologies for service oriented computing allow providers to describe and deploy services and clients to bind to and invoke these over the internet. Much of the matchmaking is still a manual task, selecting the appropriate service often a question of retrieving functional descriptions from service repositories and then ensuring that the described and required interfaces match a technical level. However, with the rapidly growing number of available services, clients are presented with a choice of functionally similar (or even identical) services. This choice allows for clients to select services that match other criteria, often referred to as non-functional attributes.

This line of work opens up two fundamental questions: how can these extra attributes be described and how can we select the most approproiate service. The selection question should of course address both the selection of isolated services as well as the selection of services within the context of other services (as for example in a business process).

The authors are of course not the only researchers to make this observation and in the past few years research results and approaches have emerged that address description of non-functional attributes and also approaches for selecting services based on non-functional attributes, amongst these there are suggestions from us and some work has been published in the proceedings of the first NFPSLA-SOC which was held in 2007. What has not been done, and this is where our contributions lie, is (1) a survey of the existing solutions and (2) a classification of approaches.

In this paper we establish a number of criteria that allow to compare approaches and allow for selecting the right approach for specific scenarios. The criteria are capturing what we believe to be the essential criteria that non-functional property aware service selection approaches should fulfil. Many of these criteria have indeed been identified by considering real case studies as part of the inContext project, where selecting services based on dynamic user and service context data is a key focus and selecting the most appropriate service is heavily based on non-functional attributes. We further establish a classification of approaches and provide a survey of the state of the art of service selection based on non-functional properties. While the classification and also the criteria might not be complete, they form the first step into a larger summary of the concerns of the field of non-functional property based service selection and provide for the first time structure to the approaches and should stimulate discussion for extensions towards a complete requirements specification and categorisation of approaches.

Overview. The remainder of this paper is structured as follows: section 2 introduces the criteria and classification, section 3 considers a classification of approaches with section 4 populating the classification while simultaneously surveying existing approaches. Section 5 provides a comparsion of the considered approaches by mapping these against the criteria. Finally we conclude with a summary and an outlook of aspects that form worthwhile directions for the field.

2 Requirements for Web Service selection approaches

Service selection is a very complex and challenging task, especially if it takes a variety of different non-functional properties into account. The fundamental issues of service selection are (1) specifying requestor's service requirements, (2) evaluation of the service offerings, and (3) aggregating the evaluation results into a comparable unit. Of course the requestor's requirements and the service offerings have both functional and complex non-functional aspects, which need to be expressed and for evaluation matched against each other (which is often only possible in some partial way).

Over the past few years a number of approaches addressing some or all of these issues in some way have emerged, concentrating on specific issues that were close to the researchers heart. In that way, many of these approaches when presented in the literature are motivated by a specific issue and then focus on solving that. In this section we present a number of criteria that we believe should be considered when comparing service selection mechanisms – note that we concentrate on approches that consider non-functional properties.

Model for non-functional properties: Service requestors need to objectively distinguish services based on their non-functional criteria to make the most appropriate choice amongst a number of services with equal or similar functionality. In the light of that a model for non-functional properties is required, that can be used in service descriptions as well as service requests. Due to the verstility of non-functional properties (and the fact that new ones might be required at any time) it is unlikely that a complete standard set can be identified. Furthermore, criteria should differ depending on the domain. For example, the printing service domain should consider print speed, color-options, location, quality and price properties. In contrast, financial service should consider security, privacy and correctness properties. Therefore it is desirable that the model for expressing non-functional properties is extensible to allow for new properties to be added in a simple fashion. Furthermore the ranking methods build on top of the models must be generic enough to be able to work with additional properties (possibly using additional information provided through the model).

Hierarchical Properties: While specific non-functional properties at a lower level are mostly what we are interested in, it is meaningful to place properties into a hierachical structure. This allows for example to group properties by domains and by broader aspects such as performance or safety. Speed and quality properties are performance aspects while security and privacy are both safety aspects. If such structure exists then users should be able to express preferences at a higher level, while service provides will express their offerings in fine detail. This will impose additional criteria on the ranking mechanism, but also provides a benefit when considering aggregation of results into a final ranking score.

User preferences: service requestors usually have varying preferences for the non-functional criteria depending on the situation they find themselves in, and of course different requestors will have different preferences. A good mechanisms should not only allow to express values for each property, but preferably also represent the relations among the preferences. For example, a user may consider the security property as more important than privacy when requesting a financial service. Hence, the selection approach needs to provide for mechansims for users to specify their preferences, that is which of the non-functional properties they feel more strongly about and also relations between these properties.

Evaluation of properties: As we discussed under "model" above, it is difficult to predict how many non-functional properties will be available, and additionaly the type of these properties. For example, the evaluation function to compute the speed criteria will be very different from the function to calculate the location criteria. It is very difficult to define a universal evaluation function for all kinds of non-functional properties. Hence, the evaluation framework must on one hand adapt to varying numbers of criteria, but also automatically identify the measurement methods that should be used to evaluate each criteria.

Dynamic aggregation: When all desired non-functional criteria have been evaluated, the next important step is to aggregate individual scores to gain a final score for the service. In this step a suitable aggregation method needs to be selected. Intuitively, arithmetic or geometric means based on weighted sums or products might appear to be an efficient and understandable choice. Unfortunately, they are not the best choice for complex situations with tens or even hundreds of criteria. For example, the service requestor considers both speed and price to be mandatory criteria, the aggregation results should reflect the preference to score 0 for not satisfying any one of these criteria. This aggregation feature can not be gained through summation, a product would solve this issue. However, in reality there are more complicated issues than just mandatory and optional criteria, and with large numbers it is easy that extremely high values meassured for criteria with a low weight can overshadow values of other factors with higher weights (and hence higher importance). Therefore, it is important that the aggregation functions are chosen dynamically to best match the aggregation problem.

Automation: Service selection can at one extreme of course be performed by a human: look up suitable services in a registry and make decisions as to which one to choose (as a matter of fact, this is currently often common practice). However, the ultimate goal of service selection research, and especially service selection based on non-functional criteria, is to provide fully automatic processes. A service designer would still specify data for the service when making it available, and a user would still be able to specify requirements, but the selection would be performed without human intervention. Such automatic selection methods are essential when for example considering context aware service provisioning (where requirements are automatically generated, and change rapidly) or selection of services within workflow contexts (essentially allowing for the execution of "abstract" processes where specific service endpoints are not predefined. One of the aspects that needs automation is the selection of evaluation functions for specific criteria, another is the selection of the agregation function.

Scalability and accuracy: Scalability here means that the approach can consider large numbers of properties, but also that many ranking processes are taking place simultaenously. Of course there is also a question as to how accurate the result is. While one would aim for perfect accuracy (that is one has proveably chosen the best service), it is often sufficient to choose a good enough service if the decision can be made quickly.

In general these criteria are influencing each other. For example scalability and automation have an inherent dependency: an automatic approach might be applied to larger problems (so it could be more scalable), while on the other hand the complexity on the approach might have an influence on how much can be automated. Another such example is the relation between basic models and the hierarchial ones: together they will be very useful, but detailed non-hierarchial models in the above sense can be suitable on their own for approaches that consider properties but not their structural dependency while hierarchical structures omitting detailed models also serve a purpose when considering the structural dependency of the properties under consideration.

We will return to these criteria later, in section 5, when we identify which of the surveyed approaches fulfils the respective criteria.

3 Categorisation of Approaches

When considering existing approaches we find that they fall into one of six categories, that essentially represent three pairs which we will refer to as dimensions next. One interesting aspect of the dimensions is that they are orthogonal, and it would be worth studying how approaches that combine a number of the categories on orthogonal axes perform – we return to this in section 6.

Considering why we have porposed this categorization amongst a pletora of possibilities is a valid question. First of all, there is no existing classification, and hence approaches in the literature are not really classified at all which makes their comparsion difficult. In the light of this, we considered that any classification would be a first step in the right direction. However, the classes are not this arbitrary, as they have a link back to our requirements as we will indicate in each dimension.

The first dimension distinguishes approaches where selection information based on policies vs approaches that rely on reports from other users and previous experiences with services, that is the reputation that a service has gained. Of course the latter could be based on the standing of the service provider, rather than just the reported quality of the service offered. The policy based approaches rely on policy languages that traditionally only allow to express a small number of non-functional properties, however that is more a limitation of current approaches than a general shortcoming. Approaches in this category are particularly well suited to deliver solutions to the aggregation of scores and evaluation of criteria.

The second dimension of differentiation is presented by how the service description including the non-functional properties is captured. There are a number of approaches based on seman[1]tic web technology, with WSDL-S and OWL-S being used to describe the non-functional properties. Other researchers decided to build extensions on UDDI repositories to allow expression of non-functional properties. Both are viable approaches, the former might prove more flexible and expandable in the long run, the latter are certainly more immediately applicable as they are based on widely deployed repository technologies. Also, for the former it is essential that service descriptions based on semantic web standards are available, while for the latter a service profile without any extra information would allow for the service to be selected if no better, closer match is available. These approaches propose models for description of non-functional properties, their interdependencies and hierachies of properties.

The third dimension distinguishes approaches based on how they allow to capture user preferences. One set of approaches uses graphical specification of

[1]

user preferences in terms of network graphs, but there would be more cope for other graphical techniques to describe preferences. Other approaches are textual, heavily relying on ontologies, to express the users preferences. This dimension is modtly concerned with capturing user requirements.

In addition to the above scalability and automation concerns are cross cutting, that is they apply to all approaches in all dimensions.

We consider the dimensions and the approaches that fall into them next, providing a survey of the state of the art in the field.

4 Survey of service selection approaches

4.1 Policy vs reputation

Policy based service selection approaches allow to specify the non-functional requirements by coding these in a QoS policy model or policy language. [11] and [9] are typical examples of policy based selection approaches.

The QoS policy model in [11] is designed as a textual document. It offers two types of non-functional properties: generic and domain specific ones. The domain specific properties are extensions of the generic ones for different kinds of services. The content of the policy model represents the service requestor's non-functional constraints and preferences. It also defines two universal evaluation functions (one is used for the case where a lower value benefits the requestor and the other one is for the opposite case) to evaluate the service's non-functional properties against the policy model. The relations between the non-functional criteria are expressed in a matrix, which is also used for their aggregation.

One of the most crucial disadvantages of this approach is the human involvement. Firstly, it is difficult to formalize all the non-functional criteria in order to allow computation of the overall score. Secondly, all the non-functional properties have to be presented as numbers or be converted into that format. Thirdly, while it captures how to express requirements, there is no mention of where and how the properties' value is stored and expressed. Fourthly, the matrix aggregation function is difficult to understand for users and does not show the relation of user's preferences at all. Finally, the final ranking scores do not reflect the satisfaction level that can be expected from the service as the overall range of values is not specified.

A similar approach is introduce in [9], the improvement is that it formalizes the the non-functional properties into a conditional policy language. The other differencs is that the service properties are dynamically detected by hardware sensors monitoring whether the selected service breaks the requestor's requirements. However, this feature limits the number of properties for practical reasons (it is only possible to monitor a small range of properties), and of course is of limited value to decisions on service selection before execution as the monitoring is performed during execution. This technique could however be very useful combined with a selection strategy that uses service execution history.

In contrast to the policy based approaches, there are a number of approaches based on trust and reputation presented in [20] and [8].

The web service selection criteria presented in [20] is statically defined for all kinds of services and each criterion is linked to a trust and reputation typology. The values of the criteria for different services are collected through the typology based feedback from communities or agencies. The selection processes are different based on the classification of trust and reputation systems which might be centralized or decentralized. For example, a centralized reputation system may use a PageRank [15] selection function. A similar idea is proposed in [8], but using IRS-III [6] selection methodology based on ontology mapping technology to calculate the ranking scores. All of these approaches focus on evaluating the selection criteria based on trust – that is whoever provides the values for the services is a trusted party. However, there has not been any uptake of these approaches for real world problems because of the complexity and time consuming manner for establishing the trust/reputation community – a system similar to certficate agencies might be required. Furthermore, the proposals do not present service evaluation and aggregation functions or consider the requestor's preferences.

While the approaches in this category introduce two interesting aspects: namely capturing user requirements by policies and also relying on observations of the services before making decisions there are some common shortcomings:

- They do not define a model of expressing service properties – they assume that the values of service properties are simply available somewhere.
- They do not consider the evaluation functions to different criteria – they either define a unified function for all kinds of non-functional properties (which is not practical in general as the properties vary widely).
- They do not consider aggregation functions in detail, simply assuming that this is not an issue (but we have mentioned earlier that this is a complex matter in itself if large numbers of criteria are considered).

4.2 UDDI-extensions vs Semantic Web Services

In order to address the problem of modelling and using service's properties, some research projects have investigated the extensions to UDDI and Semantic Web service technologies.

[17] and [3] proposed two similar types of UDDI extensions for service selection. [17] adds an extra component in the service oriented architecture called *Quality broker* which sits between the service requestor and UDDI repository. The Quality broker will randomly invoke the services which are registered in the UDDI repository through the WSDL endpoint and is in this way monitoring the performance (response time and throughput), safety (availability and reliability) and cost. In this approach all kinds of service selection problem will only consider these three non-functional properties and hence do not allow for additional properties, for example domain specific ones. Also, the approach uses a simple three value approach for representing the match to the required properties: gold, silver and bronze. Two utility functions representing the gains for service requester and provider and are composed by linear programming following equations 1 to

3. In the functions, x_i is the monitored value of the i-th criterion and v_i is a universal evaluation function, which means all x_i will be calculated by the same function and are assumed to be numerical values.

Moreover, the w_i^r and w_i^p are possibly different as they represent importance considerations from requester and provider. Consequently, the $LinearFunction$ (equation 3) matches the result between requirements and offers and it is not obvious why the service with the highest score is the best one to be selected: it represents the best overall score, but is almost certainly not requestor optimal.

$$RequesterUtility = \sum_{i=1}^{n} w_i^r v_i^r(x_i), 0 \leq RequesterUtility \leq 1. \tag{1}$$

$$ProviderUtility = \sum_{i=1}^{n} w_i^p v_i^p(x_i), 0 \leq ProviderUtility \leq 1. \tag{2}$$

$$LinearFunction = RequesterUtility + ProviderUtility. \tag{3}$$

A similar system has been proposed in [3]. The differences are (1) the *Quality broker* is a database which can be queried by giving the names of desired functional and non-functional properties, (2) the aggregation function is the simple sum function summing all desired properties, and (3) the criterion evaluation function is a universal function designed to measure a weighted distance of each value from the maximum value for the criterion.

The two basic disadvantages of the UDDI based approaches are:

– the service data and the quality information are separated. Therefore, the service provider needs to register details of their service in more than one place or the quality broker has to dynamically monitor the registered services which on one hand limits the number of criteria as monitoring is expensive and impractical as the broker would somehow need to be aware of any new service added.
– there is no extensible service quality model, meaning that the approaches are restricted for selections based on the few predefined, generic criteria.

Understanding these disadvantages, some work has been conducted to define the nonfunctional models for web services using Semantic Web Service (SWS) technology. [19] introduced a WSMO (Web service ModelingOntology)[2] based approach. The non-functional properties are organized as QoS ontology and vocabulary in WSMO. However, its evaluation functions do not make use of the full power of the SWS technology because all the values have to be numerical number without considering the the semantics of the vocabulary at all. [12] enhances on this by introducing a DAML-S based service selection approach. In [12], the matching algorithm uses the semantics of the vocabulary by introducing concepts of $Exact, Plugin, Subsumption, Container, PartOf, Misjoint$ matching.

Oldham et al. [14] present work that also clearly fits into the area of Semantic Web based approaches where they consider partner selection. Their work marries the semantic web approach with service level agreements.

[18] introduces a more generic model for capturing non-functional requirements as needed together with a type based evaluation framework which is detailed in [21]. This is not based on SWS, but rather presents an extension to UDDI.

4.3 Graphic preference modelling VS ontology based preference modelling approaches

While the previously discussed approaches in some sense address the providers perspective, little attention is paid to the requestor. However, as important as expressing non-functional properties is for the provider side it is also important to consider how user's can best express their needs.

To that end, a graphical preference modelling and service selection approach has been discussed in [16], where the preferences are modeled as TCP network graph [5] or UCP network graph [4]. These network graphs cannot only present simple importance relations among different non-functional properties but also model the dependency relations between them. For example, "color=true" might be the most important criteria for selecting a printing service outweighing price and quality. However, if there is color printing service available, then price is more important than quality, while otherwise quality is more important. While this might sound trivial, it is a situation that one would naturally consider in many decisions, and hence being able to capture it in a formal way for service selection is a very big achievement

However, the graph based modelling approach has a big disadvantage in that graphs become very complex and difficult to understand for an average user. Moreover, the selection algorithm that is presented is based on simple textual matching without making use of a model for non-functional properties and hence is less extensible and cannot deal with hierarchically structured properties.

[13] and [10] present two approaches which use ontology modelling techniques to model both the requestor's requirements and service properties. In [13], the selection algorithm is quite simple to only select the service which fully match the requestor's requirements by only using the *Exact* match concept. In contrast to it, [10] uses a price based evaluation method. The method is proven to be a NP-complete.

The ontology modelling approaches only solve the first part of capturing the requestor's preference by formally specifying the considered selection criteria with semantic vocabulary and a classification structure. Unlike graph modeling approach, they do not model the dependency relations between these criteria.

More recently, [18] and [21] proposed a context-aware service selection approach. The requestor's context is modeled using OWL/RDF which is transformed to be part of the constraint values of the category based and domain specific non-functional service selection criteria. While the initial setup of the weights for the preferences is left to users, the system can modify the values automatically to deal with emergent behaviour (e.g. an emergency status). Additionally, the weight values are associated with semantics so that a user knows what they are setting the weight to (see Table 1).

Table 1. Weight semantics

Weight	Important Level
$\mid W_i \mid = 1$	mandatory preference
$0.9 \leq \mid W_i \mid < 1$	highest importance
$0.8 \leq \mid W_i \mid < 0.9$	higher importance
$0.7 \leq \mid W_i \mid < 0.8$	priori importance
$0.6 \leq \mid W_i \mid < 0.7$	normal importance
$0.5 \leq \mid W_i \mid < 0.6$	lower importance
$0.4 \leq \mid W_i \mid < 0.5$	high desire
$0.3 \leq \mid W_i \mid < 0.4$	higher desire
$0.2 \leq \mid W_i \mid < 0.3$	low desire
$0.1 \leq \mid W_i \mid < 0.2$	lower desire

The logic preferences (that is the relation between preferences) is evaluated using an adopted LSP (Logic Scoring Preferences) method – the original LSP [7] was developed for making decisions in multi-criteria problems such as selecting hardware, but was not meant to be used in automated settings. The final aggregation method is presented in equation 4, where $W_i' = \frac{W_i}{\sum W}$ and $W_i' \neq 1$ or -1.

$$E = (\sum_{i=1}^{n} \mid W_i' \mid E_i^r)^{\frac{1}{r}} \tag{4}$$

The basic idea here is to use the power of r to reflect the dependency relations among the preferences expressed by defined weight values from the requestor and the context. The aggregation function adapts dynamically to changing requirements. Moreover, this work also introduced an extensible type-based evaluation mechanism to allow evaluation of more natural values of non-functional properties, such as set overlaps, numbers, location and boolean types. The registered service's non-functional properties are also modeled and stored as a category based OWL/RDF [1] ontology model. As with other ontology based approaches, the data provided by the service provider is trusted. Furthermore, the aggregating results of equation 4 do not to reflect the satisfaction level of the requirement but only present the comparison relations between the competitive services.

5 Approaches comparison

Having presented a number of methods for service selection, we will now provide an overall comparison between these by considering whether they match the requirements which have been introduced in the Section 2. We will use two types of comparsion values: *yes/no* to show whether the approach achieved the requirement, and *low, average, high* to indicate at what level the approach reaches the requirement. Table 2 shows the results.

Table 2. Comparison results

	NFP model	User preferences	Evaluation of properties	Hierarchical properties	Dynamic aggregation	Automation	Scalability
[11]	yes	average	no	yes	low	low	average
[9]	no	average	no	no	low	low	low
[20]	no	low	no	yes	average	low	high
[8]	yes	low	no	yes	low	low	high
[17]	no	average	no	no	average	average	low
[3]	unknown	average	no	yes	low	average	high
[19]	yes	low	no	yes	average	average	high
[12]	yes	low	yes	yes	high	average	average
[16]	yes	high	no	no	high	low	low
[13]	yes	average	no	yes	low	high	high
[10]	yes	average	no	yes	average	high	low
[18]	yes	high	yes	yes	high	high	average

The table shows that most of the approaches are lacking flexible methods for evaluating properties. Moreover, the level of expressing meaningful preferences is still low. However, the use of semantic web/ontology technologies has a huge advantage for addressing preference modelling and services non-functional properties. By analyzing the current service selection approaches, we found that most of them are designed by focusing on one or few aspects of the overall service selection problems. Our own approach [18] manages to appear very generic and comes out well against the seven requirements. This is probably a result of a number of factors: to some extent that we have tried to address all requirements as they all occured in the inContext project, to some extent that we could refer to the stength and weaknesses of many of the other approaches. However, that is not to say that the approach is perfect: further expermintal evaluation would be beneficial and there are shortcomings on capturing user requirements.

6 Conclusion and Further Work

In this paper we have outlined a series of seven requirements that we feel, based on requirements of several case studies inside a large project and thorough consideration of the topic, are essential requirements for a general service selection approach that allows users to select services based on both functional and non-functional properties. We present a categorisation of approaches.

We have reviewed a number of approaches in the context of the categorization and have presented a summary of how they perform against the requirements that we identified.

It can be concluded that most approaches contribute specific aspects to the overall picture of service selection, which requires methods for expressing user requirements, expressing service offerings and also the actual service selection method. Approaches tend to concentrate on specific of these areas and employ a vaiety of techniques to do that.

Rather than pointing out future work that we intend to do, we feel that it is more approprtiate to make some suggestions for future developments in the area of selection approaches.

One interesting aspect of the work is that, as we said, it addresses very specific issues and tries to make contributions in these. There does not seem to be any work trying to address the overall spectrum of service selection and full set of requirements. While this might be natural, as work tends to address aspects that are closer to the researchers heart or employ methods that are pioneered by a specific research group, it would be worthwhile studying how approaches that combine the best features of existing approaches compare.

Some more concrete aspects that need addressing are powerful mechanisms to capture user requirements, that are both user friendly but also expressive enough to capture large numbers of preferences and the logical relations between preferences. One aspect that falls into this arena is the capturing of weights. Also, while there is work that needs to address the needs of users who want to specify the data, there also needs to be capability to automatically capture this, partly to reduce the burden on the user, but partly to react to changes in circumstances.

Another inetersting aspect is generic models for non-functional properties, some of which have been presented, which are combined with powerful evaluation frameworks and aggregation functions that can copy with a wide variety of types of properties and changing aggregation requirements.

Finally, a discussion meriting consideration is on trust. For each of the presented dimensions one can ask how trust is established – for example can we trust because of signed agreements, because of experience or because a certification authorities assures the providers claims.

Acknowledgments. This work is partially supported by the EU inContext (Interaction and Context Based Technologies for Collaborative Teams) project: IST-2006-034718. We also like to thank the reviewers for the valuable comments.

References

1. OWL Web Ontology Language. http://www.w3.org/TR/owl-ref/.
2. Web service modeling ontology (WSMO). http://www.w3.org/Submission/WSMO/.
3. Eyhab Al-Masri and Qusay H. Mahmoud. Discovering the best web service. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 1257–1258. ACM, 2007.
4. C. Boutilier, F. Bacchus, and R.I. Brafman. Ucp-networks: A directed graphical representation of conditional utilities. In Jack S. Breese and Daphne Koller, editors, *UAI*, pages 56–64. Morgan Kaufmann, 2001.

5. R.I. Brafman, C. Domshlak, and S.E. Shimony. On graphical modeling of preference and importance. *J. Artif. Intell. Res. (JAIR)*, 25:389–424, 2006.
6. J. Domingue, L. Cabral, S. Galizia, V. Tanasescu, A. Gugliotta, B. Norton, and C. Pedrinaci. IRS-III: A broker-based approach to semantic web services. 2008.
7. J.J. Dujmovic and H. Larsen. Properties and modeling of partial conjunction/disjunction. In B. De Baets at al, editor, *Proceedings of Eurofuse Workshop on Data and Knowledge Engineering published as Current Issues in Data and Knowledge Engineering*, pages 215–224. Exit Publishers, 2004.
8. S. Galizia, A. Gugliotta, and J. Domingue. A trust based methodology for web service selection. In *Proceedings of International Conference on Semantic Computing*, pages 193–200, 2007.
9. H. Janicke and M. Solanki. Policy-driven service discovery. In *Proceedings of the 2nd European Young Researchers Workshop on Service Oriented Computing*, pages 56–62, 2007.
10. S. Lamparter, A. Ankolekar, R. Studer, and S. Grimm. Preference-based selection of highly configurable web services. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 1013–1022. ACM, 2007.
11. Y. Liu, A.H. Ngu, and L.Z. Zeng. Qos computation and policing in dynamic web service selection. In *WWW Alt. '04: Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, pages 66–73. ACM, 2004.
12. U.S. Manikrao and T.V.Prabhakar. Dynamic selection of web services with recommendation system. In *NWESP '05: Proceedings of the International Conference on Next Generation Web Services Practices*, page 117. IEEE Computer Society, 2005.
13. E.M. Maximilien and M.P. Singh. A framework and ontology for dynamic web services selection. *IEEE Internet Computing*, 8(5):84–93, 2004.
14. N. Oldham, K. Verma, A. Sheth, and F. Hakimpour. Semantic ws-agreement partner selection. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 697–706, New York, NY, USA, 2006. ACM.
15. L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998. http://citeseer.ist.psu.edu/page98pagerank.html.
16. C. Schropfer, M. Binshtok, S.E. Shimony, A. Dayan, R. Brafman, P. Offermann, and O. Holschke. Introducing preferences over nfps into service selection in soa. In *Proceedings of Non Functional Properties and Service Level Agreements in Service Oriented Computing Workshop 2007*. Springer, 2007.
17. Y.J. Seo, H.Y. Jeong, and Y.J. Song. A study on web services selection method based on the negotiation through quality broker: A maut-based approach. In *Book Embedded Software and Systems*, pages 65–73, 2005.
18. S.Reiff-Marganiec, H.Q. Yu, and M. Tilly. Service selection based on non-functional properties. In *Proceedings of Non Functional Properties and Service Level Agreements in Service Oriented Computing Workshop 2007*. Springer, 2007.
19. X. Wang, T. Vitvar, M. Kerrigan, and I. Toma. A qos-aware selection model for semantic web services. In *Service-Oriented Computing ICSOC 2006*, pages 390–401, 2006.
20. Yao Wang and Julita Vassileva. Vassileva: Toward trust and reputation based web service selection: A survey, [online], http://bistrica.usask.ca/madmuc/papers/yao-julita-ws-mas-survey.pdf, 2007.
21. H.Q. Yu and S. Reiff-Marganiec. A method for automated web service selection. In *Proceedings of IEEE Congress on Services - Part I*.