

An Architecture for Recommendation based Service Mediation

Bhaskar Mehta, Claudia Niederée, Avare Stewart, Claudio Muscogiuri, Erich Neuhold

Fraunhofer IPSI, Damstadt 64293, Germany

Abstract. Resource brokering is a crucial activity in Grid infrastructures and other environments with dynamic resource selection. The mediation between resource requirements and available resources relies on adequate resource description, which becomes a special challenge when the resources are (Grid) services.

In addition to the description of service semantics, service quality is another useful selection criteria in effective service mediation. Inspired by the success of community- and usage-pattern-based filtering and recommendation systems for targeted information access, this paper discusses automatic and manual service rating as an option for judging service quality. Such ratings can contribute to targeted service selection and complement the mediation based on service semantics.

1 Introduction

The challenge of mediating between the need for a resource and an adequate resource like e.g. a Grid service is comparable to the challenge of mediating between individual information needs and relevant information objects in an information space. In both cases

- a need for a specific type of resources exists and in reply to this request one or more resources are identified that fulfill the need as well as possible.
- descriptive metadata are a crucial building block for effectively mediating between resource needs and available resources;

However, there are some important differences. The amount of resources available in the global information space is several orders of magnitudes larger than the amount of Grid resources available today. Furthermore, information objects, especially text documents can be analyzed by automatic methods to extract at least part of their semantics (meta-data) whereas services have to be annotated manually with meta-data describing their semantics. It is not possible to extract this information automatically. The users of services are not (only) humans but also applications and other services, which requires more explicit semantics and formalization in resource description.

Basic syntactic information for consistent service invocation is given by providing information like service names, addresses, and parameters, using service

description languages like e.g. Web Service Definition Language or WSDL [7]. For targeted (semi-)automatic service selection more information is required. The most important kind of service annotation for this purpose is metadata that describes service semantic i.e. the functionality provided by the service [23]. A further service selection criterion is provided by service quality that includes parameters like availability of the service, performance, etc. Services, thus, have to be described along different complementing dimensions that together give a picture of the usefulness of a service in a specific scenario for its application.

The similarity of service mediation with the mediation of information objects suggests that some of the well-established methods of information mediation can also be exploited for improving service mediation. This paper discusses metadata based on explicate user rating and on the evaluation of usage pattern, as they are traditionally used for personalization approaches, as an additional source of information for describing service quality. This may result in information like "applications, which used service X also used service Y" or "service X and Y are the most prominent printing services".

The rest of the paper is structured as follows: Section 2 summarizes state of the art and related work in the area of syntactic and semantic services description, as well as rating and usage-pattern based information filtering approaches. Section 3 discusses dimensions of service descriptions and introduces Quality of service and Usage patterns as a complementary dimension. Starting from the idea of content rating, section 4 discusses and architecture for realizing recommendation-based service mediation. The paper concludes with a summary and some ideas for future work.

2 State of the Art and Related Work

2.1 Web Services Description and Selection

The need to support the dynamic discovery and composition of services in heterogeneous environments necessitates mechanisms for registering and discovering interface definitions and endpoint implementation descriptions and for dynamically generating proxies based on (potentially multiple) bindings for specific interfaces. The language for Web Service description WSDL [7] supports this requirement by providing a standard mechanism for defining interface definitions separately from their embodiment within a particular binding.

Web service selection can be performed from two perspectives: bottom-up and top-down. Top-down selection of web services starts from the business processes. The bottom-up perspective, on the other hand, starts from the available web services, and tries to select those that fit best. In practice, both selection approaches are often combined. This phase can, for instance, be supported with the Universal Description, Discovery and Integration (UDDI)[8] standard that provides rules for building service directories and facilitates top-down querying capabilities [13]. Nevertheless, WSDL provides service descriptions that are syntactic in nature, thus fundamentally comprising a description of the service's

provided methods, like APIs, but with no semantics, besides additional full text description of non-functional service's properties such as the geographical location of the service provider, performance, its price, and so on. Therefore, most of the search engines of WSDL/UDDI repository implement keyword-based search algorithms on table-based models of services' interface descriptions, with search results of consequently low precision (i.e. services with same interface provide different behaviors) [1].

2.2 Role of Web Services in the Grid

The Open Grid Services Architecture (OGSA)[14] defines a Grid service as follows [14]: a Web service that provides a set of well-defined interfaces and that follows specific conventions. The interfaces address discovery, dynamic service creation, lifetime management, notification, and manageability; the conventions address naming and upgrading. The Web services framework and its adopted Web Service Definition Language (WSDL) offer the Grid scenario a standard mechanism for defining both services interfaces and services binding (transport protocol and data encoding format). Furthermore, the widespread adoption of Web services mechanisms means that a framework based on Web services can exploit numerous tools and extant services, such as WSDL processors that can generate language bindings for a variety of languages.

The Open Grid Services Infrastructure specification version 1.0 [14] defines a set of conventions and extensions on the use of WSDL and XML Schema [2] to enable stateful Web services. In parallel with and subsequent to this OGS work, the Web services architecture has evolved, with for example the definition of WSDL 2.0 [7] progressing and the release of new draft specifications such as WS-Addressing [3].OGSI exploit functionality provided by other specifications (in particular, WS-Addressing) and to align OGS functions with the emerging consensus on Web services architecture.

Additional efforts aiming at refactoring OGS interfaces to incorporate the new developments to produce Web Service standards have produced five specifications, which are collectively named the WS-Resource Framework (WSRF) [11]. WSRF specifications retain all of the essential functional capabilities present in OGS, while changing some of the syntax (e.g., to exploit WS-Addressing) and adopting a different terminology in its presentation. In addition, the specifications partition OGS functionality into distinct functionality that allows flexible composition in a mix-and match manner. The factoring, composition capability and greater reliance on broadly accepted Web service concepts provide a simpler, more familiar and incremental path for developers wishing to exploit OGS functionality.

2.3 Semantic Web Services

Semantic Web technologies allow for the development of ontology systems to support the exchange of knowledge on the Web. Ontologies description languages have been developed for this purpose, like the Resource Description Framework

(RDF) [21], and its more powerful (in term of ontology modeling primitives) successor, the Web Ontology Language (OWL)[22]. A legitimate approach is to take advantage of OWL capabilities to leverage the notion of web services by enriching their signature with semantic information. The OWL Service Coalition has been recently developing a Semantic Markup language for Web Services (OWL-S)[10], with the declared goal of developing ontologies of services and resources on the Web. OWL-S approach is to put a semantic layer on top of the syntactic layer already provided by WSDL: service resources are described by Services Profiles written in ad-hoc OWL ontologies, that use OWL classes as abstract types of WSDL messages, and refer to WSDL binding to specify the actual service interface [9]. To specify semantic of services, a Service Profile can link to a Service (process) Model profile to specify inputs, outputs, post and pre-conditions.

As a complement to the semantic expressed in the Service Model, a Service Profile can link to custom ontologies in order to specify additional properties of the service resource being advertised, like service taxonomies, quality ratings and quality of service; the OWL-S specification does not dictate specific ontologies for these features, but rather it provides a semantic model to integrate them. In practice, custom ontologies can be developed to formalize dimensions of service description like quality ratings and quality of service; the design of such dimensions will be effecting, and is effected by the design of the service brokering algorithm. The approach of this paper is to infer some of those dimensions from the lessons learned in the recommender systems and collaborative filtering areas.

2.4 Recommender systems/Collaborative Filtering

The similarity of service mediation with the mediation of information objects suggests that some of the well-established methods for information mediation can also be exploited for improving service mediation. In this section we consider the approaches used in Recommender Systems for information mediation and filtering and from these approaches identify those that can be exploited for improving service mediation in grid environments.

Recommender System can be defined as system that learns about a person's needs and, based on this knowledge, has the effect of guiding the user in a personalized way to useful objects in a large space of possible options. Several classes of recommender systems can be distinguished [6]. The first, content-based recommender systems (or item-to-item correlation), (e.g. [19], [24]), make recommendations for users based on a profile built by analyzing the similarity between the content of information objects the user has seen and rated in the past and new information object. Second, in the demographic-based systems (see e.g. [18], [20]), a user is categorized into demographic classes based on personal attributes and a recommendation is made based on belonging to similar a class. Third, collaborative recommenders (or people-to-people correlation) use the similarity between users, as determined by their similar rating of information object, as

the basis of a recommendation. (see e.g. [4],[17]. Fourth, knowledge-based recommenders suggest items based on: functional knowledge of users needs; how items can satisfy those needs; and the use of the so called similarity metrics - which describes, for a given criteria such as price or location, what accounts for similar when two items are deemed substitutable [6] [26]. Finally, the utility-based approach (e.g. [6] [16]) determines a match between a user's need and a set of available options by using a utility function. The utility function is created by allowing users to specify constraints (e.g. warranty, delivery time, service contract) on features of the items under consideration. Such systems support recommendations by using constraint satisfaction techniques to produce a rank ordered list of items which satisfy the imposed constraints.

Knowledge about users needs in recommender systems are typically represented in profiles of the user. The data in such profiles are either collected explicitly or implicitly. Using explicit collection methods, the user directly provides the system with information about his preferences by selecting relevant topics, e.g. from a topic list, or by rating content (see e.g. [15], [12]). In implicit methods, the users preferences are inferred by observing their usage data or behavior and interaction with the system [25]. Example usage data include: page access, length of time viewing, site files, access logs, registration or remote agent observations, access sequences, transactional information, the contents of hotlists, or navigation history.

Taking into account the established techniques existing in recommender systems, we consider those that can support our goals and play a beneficial role in the identification of quality grid services are collaborative-based, utility and knowledge based recommender systems. The benefit of incorporating utility based recommenders is that quality measures such as speed, reliability, product availability, can be considered making it possible for example to trade off price against delivery schedule for users or applications with varying service needs. The benefit of incorporating collaborative recommender systems into the grid is that quality measures as experienced by other users can be taken into account. In knowledge-based systems, the functional knowledge can be justified by the existence of an ontological description of the resources in the grid domain [5] or additionally by system usage model based on agent observations such as: temporal usage, frequency of use, patterns of use, situation-to-situation-correlation between type of application and selected resources. Knowledge based systems can lend themselves to supporting recommendations in a grid by supporting the decision making about the substitutability between items in a set which have all been deemed semantically relevant for a given category and capable of satisfying a user's needs.

3 Quality and Usage Pattern as Dimensions for Service Description

In a Grid infrastructure, different nodes may provide competing services. Moreover, the resources may change over time. Semantics and syntax are not suffi-

cient to choose the best available resource in such an environment. The quality of resources as well as best practices of service combination reflected by usage pattern are also important factors for making adequate service choices. For example, knowing how and which process a service is going to implement when invoked, and that the process model advertised by the service offer is matching the model of the service need does not guarantee how well the service will perform. For a more complete description of services the associated metadata, thus, should cover QoS information as well as usage pattern in addition to the syntactic and the semantic dimension (see figure 3).

In order to take into account the two additional service description dimensions (QoS and usage pattern) and the combination of the dimensions, we need to consider the following three questions for each of the four dimensions.

1. What metadata is needed?
2. How can this metadata be collected in a systematic way?
3. How can this metadata can be used for service selection and mediation?

In Sections 3.2 and 3.3, we shall discuss the first two questions along the dimensions of Quality of service and Usage Patterns. In Section 4, we discuss the third question and outline an architecture for service selection and recommendation.

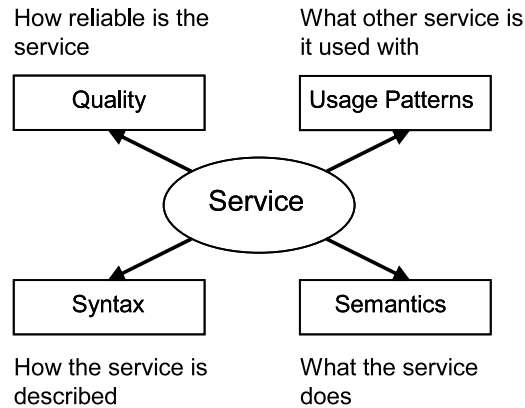


Fig. 1. Dimensions of Service Description

3.1 The Quality Of Service Dimension

Besides the functionality (semantics of a service the quality of a service is considered another important factor for service selection (see related work section). Quality statements for services are especially challenging, since service quality

may vary over time. The following list represents examples of parameters for describing service quality and are thus (partly) answering the first question:

1. *Operation Time* - This refers to the typical operation times for the service, which need not be 24x7. Services which are available for a longer time are likely to be more reliable and have more up time.
2. *Availability* - This can reflect how busy the service is, or how much usage can the service handle, or how much up time and downtime this service has. If there is a long queue or waiting time for a service on an average, other similar services could be preferred in comparison to this one.
3. *Performance* - If two nodes in a grid offer the same service, the node which is faster should be preferred. This parameter depends on how powerful a service provider is. This gives us an idea about if the service is likely to be faster in comparison to another competing node. A challenge to be solved is how performance can be described in a comparable way.
4. *Efficiency* - In a grid based environment, the efficiency depends not only on the workload of the service, but also on the network traffic, as properties like response time are unpredictable. Use of common benchmarks may increase the comparability of data.

Further metadata can be used to describe service quality. In addition to the above generic parameters for describing QoS, one might also use parameters that are characteristic for describing the quality of specific classes of services.

The next question is how to get this data. As in the case of recommendation systems, there are two basic ways of collecting the required metadata about services: explicit and implicit collection. Explicit collection contains:

Information from the service provider: Metadata about service quality can be created by the service provider as part of publishing his service.

User rating: The user is likely to be in a position to give feedback after a using the service by rating it. One possible way of collecting data from the usage will be to contact the user after the application has consumed a service for a fixed number of times (say 5). The user is then required to rate the service e.g. on a numeric scale of 1-5 for a selection of service quality parameters.

Implicit collection of QoS data can be done in different ways and depends on the quality of service parameter to be considered. Examples are:

Operation time: Since every service provider is required to register its services with the service registry, the time since which a service has been made available is known. The service can be pinged by the service registry after a fixed time repeatedly to find if the service is still alive.

Availability: This QoS parameter can also be judged by pinging the service periodically. Availability is measured as a percentage of number of successful pings against total number of pings. It is assumed that a service does not respond to a ping when it is overloaded.

Performance and efficiency: These two parameters are more difficult to assess. They would require systematic testing using e.g. agreed upon benchmarks for an entire service category.

Another indirect source on learning about service quality is analysing the frequency of use of a specific service. This aspect is covered in the next section in the context of the Usage Pattern dimension.

3.2 The Usage Pattern Dimension

For the purposes of making recommendations, **Usage Patterns** can greatly assist a new user/application in finding out which services have been used by applications with similar requirements. This can be considered as taking into account some form of best practice in combining service groups. This idea is inspired by the approach used in collaborative filtering: The assumption there is that people will tend to agree with the evaluation of a certain information item, if they have displayed similarity in the pattern of past judgements of information items. A recommendation system based on collaborative filtering makes use of the opinions of people who have already seen a piece of information to give recommendation to people who have not yet seen it.

In the grid, applications are the typical users of a data. Assuming that an application needs to authenticate itself before using a service registry usage data can be collected about an application, recording which service have been used by an application. If there is no manual rating of service available usage of a service, repetitive usage of a service can be interpreted as a sign of a positive rating of a specific service. This is comparable to tracking user behavior in a personalization system and interpret it for finding out the user's preferences (see also related work). Ratings gained by interpretation of usage pattern as well as by explicit ratings can be used to compute pairwise correlation coefficients among existing users/applications. The correlation coefficient is a measure for how similar two applications are. The system can make predictions or recommendations for services to be used next based on the degree of correlation.

Moreover the evaluation of usage patterns may also be used to determine information of the following kinds, that can contribute to improved service selection, mediation, and recommendation:

- Services frequently used together (Co-Usage)
- Most frequently used service in a category

In order to get this data, the transactions between applications and the service registry will have to be logged. Assuming every application can be uniquely identified, interactions like the services requested by an application can be tracked (this aspect is discussed in more detail in the next section). *Co-Usage* can be measured using data mining techniques on the available data. Association rules mined from the data can be used to recommend services which are normally used together with services already consumed by the user.

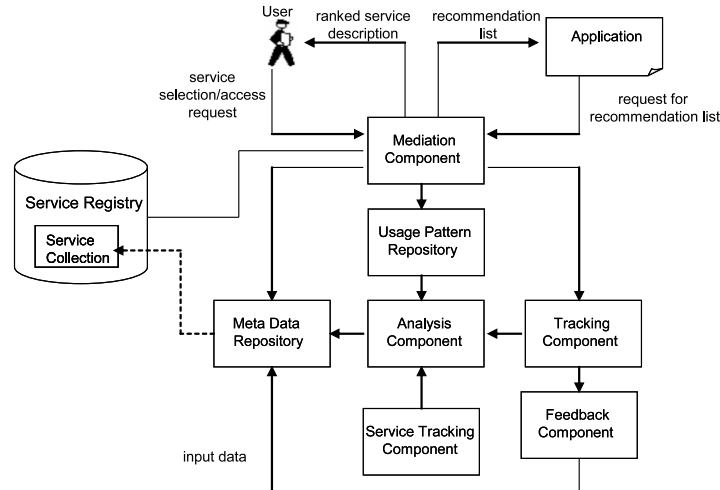


Fig. 2. Architecture for Service Mediation

4 Architecture for Service Mediation

The architecture for service recommendation and selection (see figure 2) proposed in this paper is built on the top of a preexisting *service registry* like a UDDI registry, which houses the service descriptions. As the current service registry technologies support syntactic description only, there is only basic metadata managed within the registry. All the additional metadata is managed in a *meta-data repository* containing semantic information about service functionality, as well as information about quality of service and usage patterns-based metadata.

In order to fill the metadata repository with metadata for QoS and Usage patterns, data has to be collected and analyzed. For the Quality of service metadata, there is a *Service tracking component* which monitors the services registered in the registry periodically in order to collect and validate data on QoS. As discussed in the previous section individual QoS parameters require different methods to be validated. The results of monitoring the services are used to update the respective QoS metadata within the metadata repository, like e.g. data about the availability of a service or about its operation time.

In this architecture, applications and users with service requests do not directly interact with the service registry, but with the mediation component. This enables the *tracking component*, which is a further source for service description metadata, to log which services are being used by which applications.

The *analysis component* has manifold purposes: it analyzes the data collected by the tracking component periodically. The analysis results are stored in the Usage Pattern repository as metadata for individual applications and they are also used to update metadata for individual services that are based on usage pattern (e.g. frequency of usage). Furthermore, the analysis components mines

this data to find association rules and co usage patterns applying data mining methods. The analysis results are again stored in the usage pattern repository for later use in service recommendation. Moreover, it collects data from the service tracking component and put in metadata for QoS parameters like availability and operation time.

In order to collect user ratings, there is a *feedback component* which contacts the user after an application running on the behalf of the user has consumed a service for a specific number of times (e.g. 5). User ratings are collected and combined by the Analysis component to add User rating metadata to the meta data repository.

Relying on this architecture the service selection and mediation process consists of the following steps:

1. An application or user contacts the mediation component. It identifies itself and presents its request for a service for fulfilling a task. The request describes the service along (part of the) dimensions discussed in section 3. In addition the request may also contain an application profile (comparable to a user profile in personalization approaches) that describes preferences of the application/user with respect to the service. The application may e.g. put more focus on reliable than on very fast services. Such profiles can be used to fine tune the mediation process.
2. The mediation component uses a match making algorithm and the data available in the metadata repository to create a ranked list of service candidates. For this purpose metadata on service semantics are combined with metadata on QoS and usage pattern according to the strategy of the match-making. Given the current trend in research towards semantic enrichment, it is foreseen that the description of service semantics will be in the focus of the match making algorithm. However, QoS and the Usage Patterns will play an important role in service ranking, like e.g.
 - (a) Services with higher availability and operation time can be ranked higher.
 - (b) The service with the best user ratings in its service category can be ranked higher.
3. For the top ranked services the mediation component contacts the service registry in order to fetch the syntactic descriptions of the respective services, that are required for service activation.
4. The ranked service list together with the syntactic information will be returned to the requesting application/user.

The recommendation process works as follows:

1. An application or user contacts the mediation component. It identifies itself and presents its request for service recommendation.
2. The mediation component gets the usage pattern of this application from the Usage pattern repository. The application's/user's preferences are inferred by observing their usage data or behavior and interaction with the system. Example usage data include: service access, , access logs, registration or

remote agent observations, access sequences, transactional information, or navigation history.

3. The usage pattern component finds applications with a similar usage pattern to the application making the request and then returns a recommendation list containing the services used by such applications. This computation takes ideas from the collaborative recommender systems as discussed in Section 2.4. The benefit of incorporating collaborative recommender systems into the grid is that quality measures as experienced by other users can be taken into account. In knowledge-based systems, the functional knowledge can be justified by the existence of an ontological description of the resources in the grid domain [5] or additionally by system usage model based on agent observations such as: temporal usage, frequency of use, patterns of use, situation-to-situation-correlation between type of application and selected resources. Knowledge based systems can lend themselves to supporting recommendations in a grid by supporting the decision making about the substitutability between items in a set which have all been deemed semantically relevant for a given category and capable of satisfying a user's/application's needs.

5 Conclusions and Future Work

In this paper we presented additional dimensions of service description that complement the syntactic and semantic metadata dimension and contribute to improved service mediation. As a starting point and in order to embed our ideas in the broader context we discussed the different dimensions of service descriptions for service selection and mediation. The additional dimensions, usage pattern and quality of service, are used for service mediation and service recommendation and are based on usage-pattern like co-usage of services, service rating and quality of service parameters. For both dimensions we discussed what meta data is required to capture the respective dimension and how this meta data can be collected for specific services. Furthermore, we discussed some use cases in the context of service mediation and service recommendation that exploit a combination of metadata from different service description dimensions showcasing possible applications of our approach. Finally, taking a traditional (UDDI-based) service registry as a starting point, we proposed a first component architecture for a system supporting service mediation and recommendation exploiting the four service description dimensions.

As in the case of semantic service description, this work is still in an early state and there is still a lot of research and development work to be done in this area, like e.g.:

- Methods used in data mining and recommender systems have to be validated and adapted for application in the service mediation context.
- A prototypical implementation of the proposed service mediation architecture will help to further develop the approach and will result in insights with respect to aspects like e.g. user acceptance of the service rating, i.e. is the

user ready to invest time in service rating in order to gain improved service mediation.

- Evaluation of the proposed methods will be required to validate the approach and, especially, to fine tune the combination of the metadata from the different dimension for successful service mediation.
- Furthermore, it is an important issue to examine in more detail the dependencies between the different dimensions of service description.

References

1. A. Bernstein and M. Klein. High Precision Service Retrieval. In *Proceedings of the First International Semantic Web Conference (ISWC'02), Sardinia, Italy (to appear)*. Springer-Verlag, 2002.
2. Paul V. Biron and Ashok Malhotra. XML Schema Part 2: Datatypes - W3C Recommendation, 02 May 2001, <http://www.w3.org/TR/xmlschema-2/>, May 2001.
3. A. Bosworth, D. Box, E. Christensen, F. Curbera, D. Ferguson, J. Frey, C. Kaler, D. Langworthy, F. Leymann, B. Lovering, S. Lucco, S. Millet, N. Mukhi, M. Nottingham, D. Orchard, J. Shewchuk, T. Storey, and S. Weerawarana. Web Services Addressing. <ftp://www6.software.ibm.com/software/developer/library/ws-add200403.pdf>, March 2004.
4. V. Bouthors and O Dedieu. Pharos, a Collaborative Infrastructure for Web Knowledge Sharing. In *Serge Abiteboul and Anne-Marie Vercoustre, editors, Research and Advanced Technology for Digital Libraries, Proceedings of the Third European Conference, ECDL99, Paris, France: LNCS, 1999*.
5. J. Brooke, K. Fellows, K. Garwood, and C. Goble. Semantic Matching of Grid Resource Descriptions. *Proceedings of The Second European Across Grids Conference, Nicosia, Cyprus, 2004*.
6. R. Burke. Hybrid Recommender Systems: Survey and Experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370, ?? 2002.
7. E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana. Web Services Description Language (WSDL) 1.1 - W3C Note 15 March 2001. <http://www.w3.org/TR/wsdl>, March 2001.
8. Erik Christensen, Francisco Curbera, Greg Meredith, and Sanjiva Weerawarana. UDDI Version 2.0 Data Structure Reference - UDDI Open Draft Specification 8 June 2001. <http://www.uddi.org/pubs/DataStructure-V2.00-Open-20010608.pdf>, June 2001.
9. The OWL Services Coalition. OWL-S: Semantic Markup for Web Services. <http://www.daml.org/services/owl-s/1.0/owl-s.html>.
10. The OWL Services Coalition. OWL-based Web Service Ontology OWL-S 1.1 DRAFT. <http://www.daml.org/services/owl-s/1.0/>, January 2004.
11. Karl Czajkowski, Donald F Ferguson, Ian Foster, Jeffrey Frey, Steve Graham, Sedukhin, David Snelling, Steve Tuecke, and William Vambenepe. The WS-Resource Framework. <http://www-106.ibm.com/developerworks/library/ws-resource/ws-wsrf.pdf>, March 2004.
12. J. Davies, R. Weeks, and M. Revett. Information Agents for the World Wide Web. *BT Technology Journal*, 14(4):105–114, 1996.
13. Aldo de Moor and Willem-Jan van den Heuvel. Web Service Selection in Virtual Communities. February 2004.

14. I. Foster, C. Kesselman, J. Nick, and S. Tuecke. The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. <http://www.globus.org/research/papers.html>, 2002.
15. V. Goldmann, A. Langer, and Rosenschein. S. Musag: An Agent that Learns what you Mean. In *Proceedings of the First International Conference on Practical Applications of Intelligent Agents and Multi-Agents Technology (PAAM96)*, pages 311–329, 1996.
16. H. Guttman, A. Moukas, and P. Maes. Agent-mediated Electronic Commerce: A Survey. *Knowledge Engineering Review*, 13(2):147–159, 1998.
17. J.A. Konstan, B.N. Miller, D. Herlocker Maltz, J.L. Gordon, and Reidl L.R. Applying Collaborative Filtering to Usenet News. *Communications of the ACM*, 40(4):77–87, 1997.
18. B. Krulwich. Lifestyle Finder: Intelligent User Profiling Using Large-Scale Demographic Data. *Artificial Intelligence Magazine*, 2(18):37–45, 1997.
19. K. Lang. NewsWeedweeder: Learning to Filter News. In *Proc. ICML 95*,, pages 331–336, 1995.
20. Pazzani M. A framework for Collaborative, Content-based and Demographic Filtering. *Artificial Intelligence Review*, 5-6(13):393–408, 1999.
21. F. Manola and E. Miller. RDF Primer W3C Recommendation 10 February 2004. <http://www.w3.org/TR/rdf-primer/>.
22. D. McGuinness and F. Dvan Harmelen. OWL Web Ontology Language Overview, W3C Recommendation 10 February 2004. <http://www.w3.org/TR/2004/REC-owl-features-20040210/>, February 2004.
23. S. A. McIlraith, T. C. Son, and H. Zeng. Semantic Web Services. *IEEE Intelligent Systems*, 16(2):46 – 53, March/April 2001.
24. R.J. Mooney and L. Roy. Content-Based Book Recommending Using Learning for Text Categorization. *Proceedings of the Fifth ACM Conference on Digital Libraries San Antonio, TX, June (2000)*, pages 195–204, June 2000.
25. A. Pretschner and S. Gauch. Personalization on the Web Technical Report, ITTC-FY2000-TR-13591-01. Technical report, The University of Kansas, Lawrence, KS, December 1999.
26. B. Towle and C.N. Quinn. Knowledge Based Recommender Systems Using Explicit User Models. *Knowledge-Based Electronic Markets Workshop at AAAI 2000, Austin, TX.*, 2000.