Closing the Service Discovery Gap by Collaborative Tagging and Clustering Techniques

Alberto Fernandez[3], Conor Hayes[1], Nikos Loutas[1, 2], Vassilios Peristeras[1], Axel Polleres[1], Konstantinos Tarabanis[2]

[1]National University of Ireland, Galway
Digital Enterprise Research Institute
{firstname.lastname}@deri.org

[2]University of Macedonia, Thessaloniki, Greece
kat@uom.gr

[3]CETINIA, Universidad Rey Juan Carlos, Móstoles, Spain
alberto.fernandez@urjc.es

Abstract. Whereas the number of services that are provided online is growing rapidly, current service discovery approaches seem to have problems fulfilling their objectives. Existing approaches are hampered by the complexity of underlying semantic service models and by the fact that they try to impose a technical vocabulary to users. This leads to what we call the service discovery gap. In this paper we envision an approach that allows users to query or browse services using free text tags, thus providing an interface in terms of the users' vocabulary instead of the service's vocabulary. Unlike simple keyword search, we envision tag clouds associated with services themselves as semantic descriptions carrying collaborative knowledge about the service that can be clustered hierarchically, forming lightweight "ontologies". Besides tag-based discovery only describing the service on a global view, we envision refined tags and refined search/discovery in terms of the concepts that are common to all current semantic service description models, i.e. input, output, and operation. We argue that Service matching can be achieved, by applying tag-cloud-based service similarity on the one hand and by clustering services using case based indexing and retrieval techniques on the other hand.

Keywords: service discovery, tag, clustering

1 Introduction

The Service Oriented Architecture (SOA) paradigm [34, 35] has recently become the prevalent way of building enterprise Information Systems (IS). The main idea behind SOA is the ability to use, reuse and share services from different sources. Nowadays, Web Services (WS) are the prevailing paradigm for implementing SOA, supported by significant industry investment [20].

IBM's initial reference architecture for SOA [12] identified three basic players: service providers, service requestors and service brokers. This reference architecture supports four key functionalities - discovering, composing, publishing and invoking a service. A common use case is where the service provider publishes services using the service broker, while the service requestor uses the service broker to discover services.

Complementing the SOA/WS paradigm, the Semantic Web model [4] offers a means of enhancing the brokerage model using machine-readable annotations, which could be used by agents for automated discovery and composition, even at run-time. This led to the definition of various Semantic Web Service (SWS) models. The first SWS model proposed was DAML-S [1] followed by OWL-S [31], WSMO [40], SWSF [50], WSDL-S [1] and finally the SAWSDL [26] standard recommendation. Common to these service models is the separation of aspects to describe a service in terms of *inputs*, *outputs*, and *operations* (plus often real-world *preconditions* and service execution *effects*). To describe these aspects, SWS models rely on the existence of respective domain ontologies which can be referenced in actual service descriptions.

However, SWS efforts have struggled to achieve adoption [52], due to the complexity in providing meaningful service descriptions and the lack of pervasive domain ontologies for service descriptions. Despite the fact that all OWL-S, WSMO, and SWSF were submitted to W3C, these comprehensive frameworks have not become standards [51]. Instead, the more lightweight WSDL-S framework has made a significant contribution to the recently published SAWSDL standard by W3C and has led to other lightweight "versions" of SWS description frameworks such as SA-REST [45], or WSMO-Lite [27].

Another important obstacle to the success of SWS technologies is that all existing frameworks have the assumption that the service description (semantic or not) is a publishing task that will be handled by the service providers. As such, current frameworks do not capture information about the way, the reason or the context in which services are used and do not take into account how actual users perceive services.

In the present paper, we aim to promote a paradigm where services are semantically annotated both by service providers, who may use formal SWS descriptions as well as free text tags to describe their services, and by users, who will generally use free text tags to annotate services. Our approach is influenced by the way that content is annotated in Web 2.0 platforms.

The Web 2.0 paradigm can be viewed as a minimalistic, bottom-up approach to semantic annotation, emphasising on the ease-of-use and on the need for participation over formal correctness [33,32]. In the rapidly growing Web 2.0 realm, formal description frameworks to describe services or mash-ups, or repositories for storing service descriptions have not yet emerged. Services are generated and published in a completely decentralized and uncontrolled way. Web 2.0 service discovery is enabled by lightweight annotation (tagging) services provided by third parties, such as del.icio.us or digg.com, where the annotation is completely decoupled from the actual service provision.

In turn, attempts to reconnect the "anarchic" annotation/resource description practice of Web 2.0 to the Semantic Web world are already underway with efforts such as the Meaning of a Tag (MOAT) project [37], that help in assigning machine-readable meaning (namely URIs, possibly defined in an ontology) to tags.

This paper is a first attempt to mix existing SWS description and discovery models with the Web 2.0 tag- and user-centric collaborative solutions to resource discovery.

Our approach suggests that Web 2.0 descriptions provide surface level or user-centric descriptions of services, enabling users to draw upon the perceptions of other users to narrow the discovery search space. More formal service descriptions from the matching subset of services can then be presented to the users, enabling them to identify the best-matching service to their query context. As such, service descriptions should ideally cater for both formal semantic descriptions from one of the existing frameworks as well as tag-clouds produced by collaborative annotation.

Our proposal of a mixed service discovery model consists of two main ideas. Firstly, we encourage users to provide tags upon service usage, forming tag-clouds per service. These tag-clouds can be matched using standard similarity measures against user requests. As an ignition step and refinement we propose clustering techniques in order to first generate initial tag-clouds by clustering provider descriptions based on both formal descriptions and provider-based tag sets. Secondly, we hierarchically cluster existing service tag-clouds, in order to achieve lightweight, browsable service ontologies, represented by discriminating tags per cluster.

The first step of our approach addresses the cold start problem, i.e. even without user-provided tags, we generate synthetic tag-clouds from provider descriptions. We expect precision to gradually increase with uptake of the system and the addition of user tags. We believe that this approach could be a viable alternative for facilitating the discovery of services.

The remainder of this paper is organized as follows: Section 2 discusses previous work on service discovery and matchmaking, and introduces the notion of the service discovery gap. Section 3 discusses our approach in detail. Finally, Section 4 concludes the paper and presents our future directions.

2 The Service Discovery Gap

The area of service discovery and matchmaking has been a very active research area in recent years. Nonetheless a vast majority of approaches has focused on partially complex description frameworks defining matchmaking algorithms that rely on exact logical reasoning capabilities, see [13, 36, 28, 48] for approaches for OWL-S and its predecessor DAML-S, or [21, 46] for WSMO. In acknowledgement of the infeasibility of exact logical matchmaking by sheer complexity of the involved reasoning, Stollberg et al. [46] propose a stepwise refinement of the service discovery process. There keyword-based matching serves as pre-filtering, followed by abstract matching of high-level service capabilities, and exact logical matching being the last step. D'Amato et al. [8] propose a service retrieval method based on a conceptual clustering approach, where services are specified as description logic concepts.

Catering for the infeasibility of exact logical matchmaking in general, Klusch et. al [25] present a hybrid matchmaker for OWL-S that complements logic based reasoning with approximate matching techniques from Information Retrieval. The work is inspired by earlier work for similarity-based matchmaking among software agents in LARKS [47]. In our own previous work, we have further discussed similarity-based matchmaking for semantic service descriptions [9].

Bernstein et al. suggested precise logical matching in order to increase precision of keyword only based models [5] matching descriptions of a process ontology. Since then Bernstein and colleagues have presented several non-logical approaches to enhance precise logical matchmaking for service discovery including similarity measures from IR, machine-learning, data-mining [23, 24, 22]. Still, while not expecting user requests being specified in terms of complex semantic service descriptions, these works rely on a query language with a relatively high learning curve using a SPARQL-based language for describing user requests and search terms. Somewhat orthogonal, [42] promotes the idea for using SPARQL as a "description language" i.e., an expression language for OWL-S process result's pre- and post-conditions and effects.

In summary, we have observed that, within SWS discovery, exact logical matchmaking is being superseded by similarity-based matchmaking, but still at a level of logical descriptions. Some approaches propose multi-step filtering/or selection, where simple keyword matching or IR methods may precede logical matching. However, although [21] already envisions "*way for requesters to easily locate pre-defined goals e.g. keyword matching*" concrete methods to obtain the relevant keyword set for matching services and associating them with more formal descriptions of a service are rarely found.

For real users, who provide their request in the form of free text, service discovery is hampered by what has been called the *vocabulary problem* [11]: the user requires a service but is unsure of what terms he needs to find it. This problem has previously been described in HCI [11], IR [7] and case-based reasoning [3]. Indeed, it regularly appears in retrieval systems where humans are required to guess an underlying system vocabulary, indexing or reasoning that may be non-intuitive or non coincident with human understanding. In the domain of multimedia retrieval, for example, the gap between the computational representation and the human interpretation of an image is referred to as the 'Semantic Gap' [15]. In the context of this paper, we may term this problem the *Service Discovery Gap* - caused by the breakdown between user vocabulary and expectations and service description. For example, the user's mental model may stress the usefulness of the service outputs in terms of the inputs to a common task, whereas the similarity function may primarily use the service input and operation features. This mismatch means that the user may not have a suitable vocabulary to formulate queries to retrieve relevant services.

This gap has many more faces than only the discrepancy between technical descriptions of services and the vocabulary that might be used for keyword search by potential requesters.

Language Gap: there is a lack of common semantic descriptions of available services by their providers. Where available at all, service descriptions may use descriptions in varying formats, levels of granularity and underlying (logical) languages. Despite this most service frameworks contain the same service aspects contained in WSDL: *Inputs, Outputs, Operations,* (and more rarely *Preconditions* and *Effects,* which are for instance missing in SA-WSDL).

Provider/User Gap: This gap occurs where providers have different intentions for the use of their service than the users who consume the service.

3 Web 2.0 Approaches to Service Discovery

To address this problem we look at how user-defined tags might help. Tags are short informal descriptions, often one or two words long, used by Web users to describe online resources. There are no techniques for specifying “meaning” or inferring or describing relationships between tags. Tag-clouds refer to aggregated tag information, in which a taxonomy or “tagsonomy” emerges through repeated collective usage of the same tags. Part A of Figure 1 illustrates a tag-cloud in the blog domain.
The advantage of using tags in the context of service discovery is that they supply a user-defined vocabulary based on a consensus of how the service is perceived or used in the world. For the needs of our research, we assume that each service that is made available via the Web has its own tag-cloud.

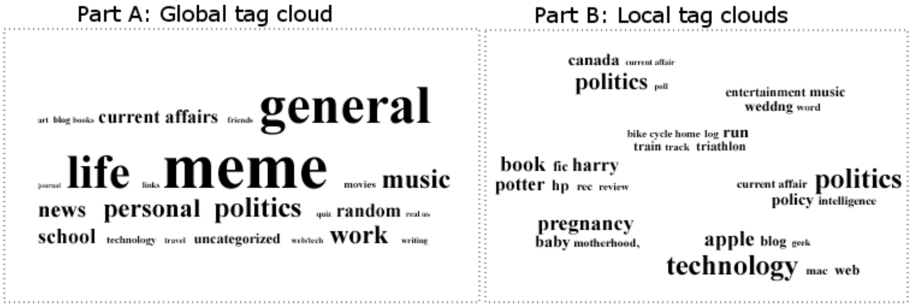

Figure 1: A global tag cloud and several local ‘synthetic’ tag clouds, which were produced by clustering the associated blog content [17].

3.1 MAC/FAC Similarity Matching

In this section we introduce several approaches to service discovery using Web 2.0, formal service descriptions and similarity matching. The conceptual framework for the similarity matching that we draw upon is from case-based reasoning (CBR), a well-established domain in which similarity-based retrieval is a fundamental part. The key observation we take from this domain is that similarity-based retrieval can be viewed as a process of at least two steps, rather than a single-shot retrieval. An influential theory of similarity-based retrieval in this domain is the *Many Are Called/Few Are Chosen* (MAC/FAC) model of Gentner, Forbus and Law [10].

The first or MAC stage uses a process called *surface similarity*, a relatively inexpensive matching function that uses simple surface level features to return a subset of items from the search space. The second stage or FAC stage involves a deeper, more powerful and (more expensive) matching process carried out on this subset using richer, structured feature information. The MAC/FAC procedure has been used for many years in several forms in case-based reasoning. A typical example is a MAC stage where incremental query-expansion is performed on a database followed by a FAC stage where similarity matching is performed on the resultset [44]. An alternative approach, closer to the approach we propose, involves a MAC stage using collaborative filtering followed by FAC stage using similarity-based matching

[19]. The essential observation is that retrieval can take place incrementally, in at least two steps, where the initial steps involve the retrieval of a subset of item by means of an inexpensive function on surface-level features followed by a refinement of this subset using more sophisticated techniques on structural features.

Approach 1: Matching Tag clouds

This model allows us to incorporate tag information and formal service descriptions into a user-centric service discovery model. If we take a step back and re-think which "semantic descriptions" would best suit typical free text service requests, the obvious solution are simply *tag-clouds* to describe and classify services. By a tag-cloud-based service description *C* here we mean a set of pairs $<t_i,n_i>$, where t_i is a free text tag and n_i is the frequency of tag t_i in the tag cloud *C*. Tag clouds are used in faceted browsing and often displayed graphically there, emphasizing the weight n_i by the font size of a tag (see Figure 1). A service tag-cloud is obtained by aggregating the free text tag annotations provided by the users of that service, being the frequency n_i the number of users that included the tag t_i in their annotation of that service.

Such tag-clouds immediately solve the language gap, since there is no more formal language involved which needs mediation. Matching user requests (i.e. a tag set) to tag-clouds is obvious: Tag-clouds have a natural correspondence to the typical vector space model used in standard document classification and information retrieval, and indexing methods. Assuming we had tag-clouds describing each service in place, which properly describe the meaning of a service, matching itself would be an almost straightforward task. The user specified keywords would just be used as a "filter" to mask each service tag cloud, dropping all tags that are not of interest and the weighted sum of this masked tag cloud would denote the degree of match.

To compare tag-clouds, weights are usually normalised. The *normalised tag frequency* r_i of tag t_i in *C*, where *k* is the number of tags in *C*, is defined as:

$$r_i = \frac{n_i}{\sum_k n_k}$$

Let *T* be a normalised tag-cloud and *Q* a user specified tag set, then the similarity between *T* and *Q* is defined as:

$$sim(T,Q) = \sum_{t_i \in T} \delta(tp_i, Q)$$

where tp_i stands for the normalised tag pair (<tag, weight>), and δ is defined as:

$$\delta(< t_1, r_1 >, Q) = \begin{cases} r_1 & if\ t_1 \in Q \\ 0 & otherwise \end{cases}$$

This naïve approach of tag-cloud matching could obviously be refined by lessons learned from the traditional semantic service matchmaking realm. Typical proposals to service matching use asymmetric measures in the degree of match between request and offer, which takes into account the subsumption relation between them (e.g. *plug-in* vs. *subsumes* in Paolucci's [36] approach). For example, suppose two service descriptions s_1 = *buy book*, and s_2 = *buy fiction book*, the degree of match is usually different if the s_1 is the user request and s_2 is the provider, or vice versa. This asymmetry is lost in the tag cloud (there is no subsumption relation between concepts), but might be emulated by using the subset operator between tag sets (e.g. *{buy, book}* $\subseteq$ *{buy, fiction, book}*). However, asymmetry might not make sense when the intended use of the similarity measure is clustering (as detailed below), since this process is done without a reference request. Thus, these approaches cannot be directly applied in a clustering context.

The tag-cloud description metaphor may be further refined by dividing each service description according to the aspects common to current SWS description frameworks. For example, instead of a single tag cloud, we could envision separate tag-cloud descriptions per service for *inputs*, *outputs* and *operations*. Requests could be grouped likewise, e.g. providing separate search fields for these different aspects in a tag-based service discovery engine. In that case, service similarity must combine the similarity value for each of these fields. Different options can be considered here. If we consider those fields as a conjunctive set (i.e. all are expected to be matched) then a triangular norm (e.g. the *minimum*) can be used. A more general approach is a weighted sum of each similarity, where the weighting parameters can be established a priori (e.g. equally distributed: 1/3) or defined by the user.

Approach 2: Browsing a Tag-Cloud Concept Hierarchy

Approach 1 provides a means for querying service descriptions using simple natural language query terms. Although a tag-cloud describes the most frequently used terms by other users of a service, a new user may still have trouble formulating a query that would match it, particularly if the tag-cloud is sparse. An alternative approach is to provide a visual browsing mechanism where the various concepts represented in the service space are described using representative tags. To achieve this, we propose a type of ontology that is automatically built by matching similar tag-clouds in order to allow users to browse service descriptions at different levels of granularity. To realise this, we draw upon three techniques. The first is in the area of hierarchical clustering [53], the second is in the area of tag analysis in blogs [17]. The third is in centroid-based classification [14].

In the context of this work, hierarchical clustering allows us to produce a browseable interface of service descriptions at different levels of granularity using tag data. Furthermore, a clustering approach provides the means for implementing recommendation mechanisms: when a user finds a service, other services that belong to the same cluster can be recommended to her.

Hierarchical clustering can be further subdivided into two approaches: agglomerative or bottom-up approaches where data objects are initially assigned to their own clusters and then pairs of similar clusters are repeatedly merged until a whole tree is formed; and partitional or top-down approaches where the entire corpus is initially divided into two clusters and these clusters are repeatedly sub-divided until a whole tree is formed. Conventional wisdom has it that agglomerative approaches, while computationally more expensive, tend to outperform partitional approaches in clustering accuracy. Recent analysis on large, high dimensional data sets has suggested that this is not necessarily the case [53]. Furthermore, a hybrid approach called *constrained agglomerative clustering*, where an initial partitional approach provides constraints for the subsequent agglomerative process, has demonstrated improved clustering performance with small increase in computational cost over partitional approaches [53].

Thus, in approach 2 we use constrained agglomerative clustering to cluster our service descriptions into a concept tree. Each service is represented in terms of its tag-cloud and similarity between services is calculated based on similarity between tag-clouds. Note that semantic concept similarity techniques ([38, 29, 39]) cannot be applied in this context, since they assume that concepts are defined and related to each other in some ontology. In general clustering rests upon a fundamental hypothesis in information retrieval: Van Rijsbergen's cluster hypothesis, which proposes that similar documents are likely to be more relevant to an information requirement than less similar documents [49]. In the context of this paper, we consider a tag-cloud to be a document and an information requirement to be service discovery requirement. We believe that this is a reasonable assumption as tag-cloud data can be pre-processed, stemmed and weighted as input data for clustering in exactly the same way as text document data.

Using the vector-space model, each tag-cloud is represented as a vector in term space and each term in the vector is weighted according to the standard *tf-idf* weighting scheme [41]. In the vector-space model, similarity is calculated using the cosine measure. To prevent large clouds having undue influence during similarity matching, each vector is normalised so that it is of unit length on the hypersphere. The corpus of tag-cloud vectors can then be used as input on the clustering algorithm.

For a detailed account of the constrained agglomerative clustering, we refer the reader to the work of Zhao et al. [53]. The output of the algorithm is a dendrogram that can be browsed from its root nodes (containing all tag cloud documents) to its leaves, where each leaf represents a single tag-cloud (and its associated service). At each concept node, the *extensional* description of the concept is represented in terms of the services associated with the tag-clouds in that node. The *intensional* and thus browsable, description of the concept at each node is easily extracted from the cluster *centroid* at the node.

The cluster centroid at each node is produced by firstly producing a *composite vector* of the tag-cloud vectors contained in the cluster at the node and then normalising each term of the composite vector by the number of tag clouds (at the node). For a node *p*, containing a set *N* of tag cloud vectors, the centroid vector C_p is defined by

$$C_p = \sum_{n \in N} n \Big/ |N|$$

The cluster centroid is a vector that contains a weighted representation of the tags most representative of the concept in cluster. A synthetic tag-cloud can be extracted from the centroid using a threshold to filter lowly weighted terms and using the tag term weights as an input to any tag cloud presentation algorithm.

The overall output is a browseable dendogram, where each node is represented by a synthetic tag cloud representing service description at different levels of specificity (see Figure 2).

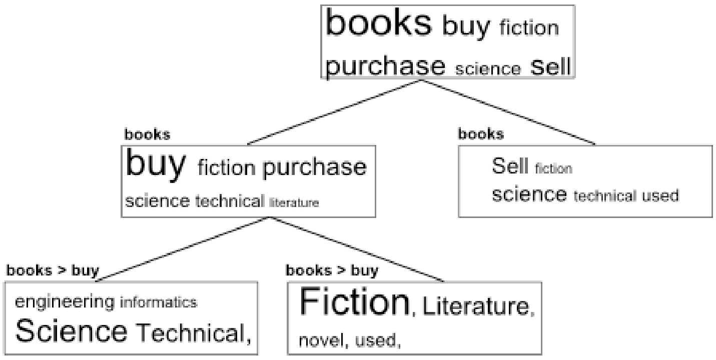

Figure 2: An illustration of a dendrogram induced from tag cloud data. At each node a threshold controls how many tags are displayed. For each sub-node, the single mostly highly weighted tag in each parent node is displayed as a part of a path summary. E.g. books>buy

The Cold Start Problem

A fundamental problem with all online services that rely upon collecting social data is the *cold-start problem* [30, 43] which refers to the difficulty in offering a service when there is yet no user data and the difficulty in collecting user data when there is no service. Although, the problem has generally been defined in terms of collaborative recommendation systems, it is equally relevant to services that rely upon user submitted tag data, such as the one that we have just defined.

A typical solution to the cold-start problem in collaborative recommendation is to deploy what is termed a *content-based* service along side the collaborative service [43, 6]. The key idea is that the content-based service can be deployed where there is insufficient social data to make a socially-based recommendation. The terms ‘content’ loosely refers to any non-socially derived descriptive data that can be used for retrieval purposes, typically using IR inspired matching algorithms.

In the context of this work, we plan to leverage a ‘content’ based approach to the ‘cold start’ problem by clustering the semantic descriptions that the service providers add to their services. These may both consist of well-structured formal descriptions that follow one of the SWS frameworks discussed earlier, i.e. OWL-S, WSMO, SAWSDL, textual descriptions, or tag sets provided directly by the provider. Note that, in the case of SWS descriptions, similarity-based matchmaking techniques for semantic service descriptions (as described in section 2) must be used.

Our technique draws upon the work of Hayes et al. which uses content clustering and tags to produce interpretable tag-based summaries of data in the blog domain [16] (See Figure 1). The essential observation of this work is that where tag data is sparse, the underlying content data can be clustered, producing synthetic tag-clouds as by-process. These tag clouds are shown to be strong indicators of the cluster semantics and coherence. Currently we are collecting service provider descriptions, which will act as input for a cold-start clustering process. The cluster concepts will be represented by synthetic tag clouds extracted from the available service descriptions, providers' tags, as well as tag sets extracted from service descriptions by traditional information extraction techniques.

The cold start approach raises the question as to when we know when there is enough social data or what to do when there is social data for some services and not for others. Although we do not pretend to have an answer at this point, we do acknowledge that a substantial amount of research has been directed to the question of interleaving outputs from different models in the field of recommender systems [6]. Furthermore, the success of a system must be measured in terms of use satisfaction and, in this regard, we plan to exploit our previous experience in evaluating online whether one recommender algorithm improves upon another [18]. Another question is how often the clustering process needs to be carried out, given that tag-clouds evolve over time. We propose that experimental analysis will provide the answer to this question and direct the reader to our initial work on this subject in the area of clustering blog data [17].

4 Conclusions and Future Work

In this paper we started by describing the current state in service provision and by outlining the problems that are raised during service discovery by introducing the service discovery gap. Afterwards, we made the assumption that in our world of services, services are semantically described both by service providers and by users. The semantic descriptions of the services providers follow some SWS framework, while those of the users are expressed via tag clouds.

Our solution allows users, at a first step, to query or browse services using free text tags, thus providing an interface in terms of the users' vocabulary instead of the service's vocabulary. Then, in a second step, the users can query deeper in this narrowed result set by using concepts that are common between different service models, i.e. input, output, and operation. A description of the second step is beyond the scope of this paper. However, we have recently developed a common mapping language between service descriptions that allows their input, output and operation tasks to be compared. After step one, where the user has found a candidate set of services using the tag cloud method, he/she can make an informed choice of services available without having to learn the syntax of each service type. Further populating this set of common concepts is one of the most important future steps in our research agenda.

Two approaches to service discovery have been suggested: a similarity-based approach for querying tag clouds and a browsing approach using hierarchical clustering. In addition, we proposed a content-based approach to solve the cold-start problem.

As part of our future work, we plan to implement and evaluate these different approaches using standard evaluation techniques from machine learning and information retrieval, as well as on-line approaches to measuring user satisfaction [18].

Acknowledgments

This research has been partially supported by the European Commission under grant FP-6-IST-034718 (inContext), by the Spanish Ministry of Education and Science through projects URJC-CM-2006-CET-0300, and CONSOLIDER CSD2007-0022, INGENIO 2010 (Agreement Technologies), as well as by Science Foundation Ireland under the Líon project (SFI/02/CE1/I131).

References

1. Akkiraju, R. et. al.: Web Service Semantics - WSDL-S, W3C Member Submission, http://www.w3.org/Submission/WSDL-S/
2. Ankolekar, A. et. al: DAML-S: Semantic Markup For Web Services. Proceedings of the International Semantic Web Workshop (2001).
3. Avesani, P., Hayes, C., Cova, M. (2005). Language Games: Solving the Vocabulary Problem in Multi-Case-Base Reasoning in proceedings of the 6th International Conference on Case-Based Reasoning, ICCBR 05. Hector Munoz-Avila, Francisco Ricci (eds), Springer
4. Berners-Lee, T., Hendler, J., Lassila, O.L.: The semantic web. Scientific American, vol. 284(5), pp. 34–43 (2001)
5. Bernstein, A., Klein, M.: Towards high-precision service retrieval. IEEE Internet Computing, vol. 8(1): 30-36 (2004)
6. Burke, R. 2002. Hybrid Recommender Systems: Survey and Experiments. User Modeling and User-Adapted Interaction 12, 4 (Nov. 2002), 331-370.
7. Chen, H., Ng, T. D., Martinez, J., and Schatz, B. R. 1997. A concept space approach to addressing the vocabulary problem in scientific information retrieval: an experiment on the worm community system. J. Am. Soc. Inf. Sci. 48, 1 (Jan. 1997), 17-31.
8. d'Amato, C., Staab, S., Fanizzi, N., and F. Esposito. *Efficient discovery of services specified in description logics languages*. In Proceedings of the ISWC-2007 Workshop on Service Matchmaking and Resource Retrieval in the Semantic Web (SMR2), 2007.
9. Fernandez, A., Polleres, A., and Ossowski, S. Towards Fine-grained Service Matchmaking by Using Concept Similarity, In Proceedings of the ISWC-2007 Workshop on Service Matchmaking and Resource Retrieval in the Semantic Web (SMR2), 31-45. 2007.
10. Forbus, K. D., Gentner D., and Law, K. 1994. MAC/FAC: A Model of Similarity-based Retrieval. Cognitive Science 19, 141-205.
11. Furnas, G. W., Landauer, T. K., Gomez, L. M., and Dumais, S. T. 1987. The vocabulary problem in human-system communication. Commun. ACM 30, 11 (Nov. 1987), 964-971.
12. Gisolfi, D., IBM: *An introduction to dynamic e-business*. http://www.ibm.com/developerworks/webservices/library/ws-arc1/ (2001).
13. Grimm, S., Motik, B., Preist, C.: *Variance in e-business service discovery*. In Proceedings of the ISWC Workshop on Semantic Web Services. (2004).

14. Han, E. and Karypis, G. 2000. Centroid-Based Document Classification: Analysis and Experimental Results. In Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery (September 13 - 16, 2000). D. A. Zighed, H. J. Komorowski, and J. M. Zytkow, Eds. Lecture Notes In Computer Science, vol. 1910. Springer-Verlag, London, 424-431.
15. Hare, J. S., Lewis, P. H., Enser, P. G. B. and Sandom, C. J. (2006) Mind the Gap: Another look at the problem of the semantic gap in image retrieval. In: Multimedia Content Analysis, Management and Retrieval 2006, 17-19 January, San Jose, California, USA.
16. Hayes, C., Avesani, P. Using Tags and Clustering to Identify Topic-Relevant Blogs. In Proceedings of the 1st International Conference on Weblogs and Social Media (ICWSM 07): 67—75 Boulder, Colorada, U.S.A, 2007
17. Hayes, C., Avesani, P., and Bojars, U. An Analysis of Bloggers, Topics and Tags for a Blog Recommender System. In Bettina Berendt, Andreas Hotho, Dunja Mladenic, Giovanni Semeraro (ed.), LNAI: Advances in Web Mining Springer, 2007.
18. Hayes, C., Massa, P., Avesani, P., Cunningham, P. (2002) An On-line Evaluation Framework For Recommender Systems in Proceedings of the Workshop on Recommendation and Personalization in eCommerce at AH2002, University of Malaga Press, 2002. F. Ricci and B. Smyth (editors). ISBN 699-8196-7
19. Hayes, C., Cunningham, P. (2003) Context Boosting Collaborative Recommendations in Applications and Innovations in Intelligent Systems XI, proceedings of the Twenty-Third Annual International Conference of the British Computer Society's Specialist Group on Artificial Intelligence (SGAI), Cambridge. Springer Verlag.
20. InfoWorld: http://www.infoworld.com/article/08/02/25/news-soa-adoption_1.html (2008)
21. Keller, U., Lara, R., Lausen, H., Polleres, A., Fensel, D.: Automatic Location of Services. Proceedings of the 2nd ESWC 3532 (2005)
22. Kiefer, C., and Bernstein, A. The Creation and Evaluation of iSPARQL Strategies for Matchmaking. 5th ESWC, 2008.
23. Kiefer, C., Bernstein, A. and Locher, A. Adding Data Mining Support to SPARQL via Statistical Relational Learning Methods. 5th ESWC, 2008.
24. Kiefer, C., Bernstein, A., Lee, H., Klein, M., and Stocker, M. Semantic Process Retrieval with iSPARQL. 4th ESWC, 2007.
25. Klusch, M., Fries, B., Sycara, K.: Automated semantic web service discovery with owls-mx. Proceedings of the AAMAS, pp. 915-922 (2006)
26. Kopecky, J., Vitvar, T., Bournez, C., Farrell, J.: SAWSDL: Semantic Annotations for WSDL and XML Schema, IEEE Internet Computing, vol. 11 (6) (2007)
27. Kopecky, J., Vitvar, Zaremba, M., Fensel, D.: WSMO-Lite: Lightweight Semantic Descriptions for Services on the Web Full, Proceedings of 5th ECOWS, pp. 77-86 (2007)
28. Li, L. and Horrocks, I. A software framework for matchmaking based on semantic web technology. Int. J. of Electronic Commerce, 8(4):39–60, 2004.
29. Li, Y., Bandar, Z., and McLean, D. An approach for measuring semantic similarity between words using multiple information sources. IEEE Trans.

Knowl. Data Eng., 15(4):871–882, 2003.
30. Maltz, D. and Ehrlich, K. 1995. Pointing the way: active collaborative filtering. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (Denver, Colorado, United States, May 07 - 11, 1995). I. R. Katz, R. Mack, L. Marks, M. B. Rosson, and J. Nielsen, Eds. Conference on Human Factors in Computing Systems. ACM Press/Addison-Wesley Publishing Co., New York, NY, 202-209.
31. Martin, D. et. al.: OWL-S: Semantic Markup for Web Services, W3C Member Submission, http://www.w3.org/Submission/OWL-S/ (2004)
32. Murugesan, S.: Understanding Web 2.0. IT Professional, vol. 9(4), pp. 34-41 (2007)
33. O'Reilly, T.: What Is Web 2.0, Design Patterns and Business Models for the Next Generation of Software, http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html (2005)
34. OASIS Reference Model for Service Oriented Architecture, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=soa-rm
35. OPEN Group, http://www.opengroup.org/projects/soa/doc.tpl?gdid=10632
36. Paolucci, M., Kawamura, T., Payne, T., and Sycara, K. Semantic Matching of Web Service Capabilities. In ISWC, pages 333–347. Springer Verlag, 2002.
37. Passant, A. MOAT Project: moat-project.org/
38. Rada, R., Mili, H., Bicknell, E. and Blettner, M. Development and application of a metric on semantic nets. IEEE Transactions on Systems, Man and Cybernetics,, 19(1):17–30, 1989.
39. Resnik, P. Using information content to evaluate semantic similarity in a taxonomy. In IJCAI, pages 448–453, 1995.
40. Roman, D., Lausen, H., Keller, U. (eds.), Web Service Modeling Ontology (WSMO), Technical report, WSMO Final Draft, http://www.wsmo.org/TR/d2/v1.2/ (2005)
41. Salton, G. 1989. Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer. Addison-Wesley.
42. Sbodio, M. L., Moulin. C.: SPARQL as an expression language for OWL-S. Proceedings of OWL-S: Experiences and Directions, 4th ESWC (2007)
43. Schein, A. I., Popescul, A., Ungar, L. H., and Pennock, D. M. 2002. Methods and metrics for cold-start recommendations. In Proceedings of the 25th Annual international ACM SIGIR Conference on Research and Development in information Retrieval (Tampere, Finland, August 11 - 15, 2002). SIGIR '02. ACM, New York, NY, 253-260
44. Schumacher, J., Bergmann, R. (2000) An effective approach for similarity-based retrieval on top of relational databases, in: Fifth European Workshop on Case-Based Reasoning. Springer-Verlag.
45. Sheth, A., Gomadam, K., Lathem, J.: SA-REST: Semantically Interoperable and Easier-to-Use Services and Mashups, IEEE Internet Computing, vol. 11 (6): pp. 91-94 (2007)
46. Stollberg, M., Keller, U., Lausen, H., Heymans. S.: Two phase web service discovery based on rich functional descriptions. Proceedings of 4th ESWC (2007)
47. Sycara, K., Klusch, M., Widoff, S., Lu, J., Larks, K.: Dynamic matchmaking

among heterogeneous software agents in cyberspace. Autonomous Agents and Multi-Agent Systems, 5(2), Kluwer (2002)
48. Sycara, K., Paolucci, M., Anolekar, A., Srinivasan, N.: Automated discovery, interaction and composition of semantic web services. Web Semantics, 1(1), Elsevier (2003).
49. Van Rijsbergen, C. J. Information Retrieval. Butterworths, second edition, 1979.
50. W3C: SWSF Submission, www.w3.org/Submission/SWSF/
51. W3C: Team Comment on Web Service Modeling Ontology (WSMO) Submission, http://www.w3.org/Submission/2005/06/Comment (2005)
52. Xuan S.: Semantic Web Services: An Unfulfilled Promise. IEEE IT Professional, vol. 9(4), pp. 42 – 45 (2007)
53. Zhao, Y., Karypis, G., and Fayyad, U. 2005. Hierarchical Clustering Algorithms for Document Datasets. Data Min. Knowl. Discov. 10, 2 (Mar. 2005), 141-168.