Similarity Measurement about Ontology-based Semantic Web Services

Xia Wang
Digital Enterprise Research Institute
IDA Business Park,
Lower Dangan Galway, Ireland
xia.wang@deri.org

Yihong Ding
Dept. of Computer Science
Brigham Young University
Provo, Utah, USA
ding@cs.byu.edu

Yi Zhao
Chair of Computer Engineering
Fernuniversität
Hagen, Germany
yi.zhao@fernuni-hagen.de

Abstract

Measurement of semantic similarity between Web services is an important factor for Web service discovery, composition, and even execution. Semantic Web services (SWS) are usually specified based on ontologies. The measurement of semantic similarity between Web services thus can be reduced to computing semantic distances between ontologies. In this paper, we briefly surveyed three major existing ontology-distance-computation algorithms and enhanced them to measure the single and multiple ontolgies similarity in SWS context. Based on this survey, we summarized a new hybrid ontology-similarity-measurement methodology that measures similarity between Semantic Web Services.

1. Introduction

Service similarity is crucial to service discovery, selection, composition, and even execution. Especially semantic service discovery aims to locate the best matched service, it mostly depends on the measurement of the similarity between an user's service requirements and the profiles of published services. Currently, the semantic Ontology languages for services, such as the OWL-S[1] and the WSMO[2], are required to semantically represent service capabilities, including non-function information (including Qos of service), functional information (*IOPE* of services operations, denoting input, output, precondition and effect). The service discovery, therefore, focuses on the matchmaking of service capability [15] and QoS, while less work is done on ontology-based services selection.

Moreover, Ontology receives great attention in the progressively emerging Semantic Web [2] and Semantic Web Services [6], by formally defining the concepts and relationships in a machine understandable way and enabling knowledge sharing and reuse. As the elements of the representation of semantic services, the similarity of the ontologies used is crucial to service similarity, especially when considering the discovery and execution.

Ontology similarity which is related to Ontology mapping is a well known topic in information retrieval, database integration systems, and artificial intelligence fields. Also, there is a wealth of work on similarity measures of Ontology concepts and concept-related notions [19]. However, the measurement of similarity of ontologies and concepts itself is not easy, additionally many specific features (see section 2.2) in Semantic Web Service description environment.

After surveying the previous Ontology similarity measures and their application situations, in our SWS context two methods are combined to adapt to calculate the semantic distance of single formal Ontology concepts, e.g. *code* and *zip* in the examples of figure 1. The approaches are a fuzzy-weighted associative network (edge-based measure) and an information-theoretical approach (content-based measure).

Also regarding the compound ontology concepts in semantic service context, for example, in figure 1 the concepts *findZipCodeDistance* and *CalcDistTwoZipsKm*, are not the

[1] OWL-S, $http://www.w3.org/Submission/OWL-S/$
[2] WSMO, $http://www.wsmo.org$

formal single terms as the ones in WordNet[4], in this case the traditional method (e.g. edit distance of strings) is not useful to measure their semantic similarity. Therefore, we refine the hierarchical clustering algorithm to calculate the distance of two compound concept terms, similar to [7, 4].

In this paper we aim to solve the ontology similarity in a semantic service environment. First, we differentiate two cases of service ontology concepts: single and compound ontology concept to measure their similarity in service context. Then, a hybrid Ontology-similarity measurement is proposed by combining and refining three existing methods. Finally, we define our ontology similarity-based model as $sim_S = \Sigma sim_O \in [0, 1]$ to improve the service selection; This model fuzzily and quantitatively measures the service similarity basing on service ontology similarity.

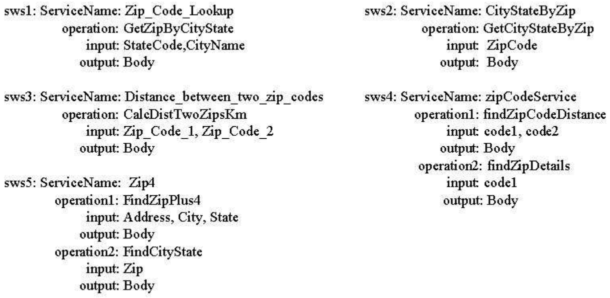
This paper is structured as follows. In Section 2 we state the occurring problems of Ontology in Semantic Web Service description context, and investigate the specific name features of ontology concept. A Ontology concept distance definition and three refined ontology similarity algorithms are discussed with examples of single formal term and compound term in Section 3. The service similarity measurement is defined in Section 4. In Section 5 and 6 we respectively discuss related works, and give conclusion and indications for future work.

2 Ontology in SWS Description Context

2.1 Problem Statements

In order to illustrate the challenge of measuring the similarity of semantic Web services, we extract a set of *zip code* related services from the *dataset* of *OWL-S* annotated Web Services of the University College Dublin[3]. In Figure 1., there are snatch description of four services, which are used for looking up a zip code or calculating the distance between two places according to the given zip codes. The information shown is retrieved from the *wsdl* documents of the respective service.

Current service matchmaking algorithms normally focus on measuring the syntactic (as service name, service text description and so on) and semantic (as service capabilities) of service. Taking *sws4* and *sws5* of Figure 1 as examples, if we assume that *zip* and *code* have the similar meaning, intuitively, by comparing service name and operation name, service *sws4* and *sws5* are regarded as similar from the signature level; and by matching their operation, as both *operation2*, they also have similar inputs and outputs, so that *sws4* and *sws5* are concluded as similar services. This means that both can provide detailed information of a city according to the given zip code.



```
sws1: ServiceName: Zip_Code_Lookup              sws2: ServiceName: CityStateByZip
         operation: GetZipByCityState                     operation: GetCityStateByZip
             input: StateCode,CityName                        input:  ZipCode
            output: Body                                     output:  Body

sws3: ServiceName: Distance_between_two_zip_codes  sws4: ServiceName: zipCodeService
         operation: CalcDistTwoZipsKm                        operation1: findZipCodeDistance
             input: Zip_Code_1, Zip_Code_2                       input: code1, code2
            output: Body                                        output: Body
                                                             operation2: findZipDetails
sws5: ServiceName:  Zip4                                         input: code1
        operation1: FindZipPlus4                                output: Body
             input: Address, City, State
            output: Body
        operation2: FindCityState
             input: Zip
            output: Body
```

Figure 1. Snatch of Semantic Web Services Description

Further, if we assume that a machine can understand some similarity between $\{zip, ZipCode, Zip\_Code\_1, code, code1\}$ and $\{CaleDisTwoZipsKm, findZipCode Distance\}$, then intuitively and naively from the above example services of Figure 1, we know that *sws1:operation1* is similar to *sws5:operation1*; *sws2:operation1* is similar to *sws3:operation2* and *sws4:operatio2*; and *sws3:operation1* is similar to *sws4:operation1*.

Obviously, similarity, whether syntactic or semantic, the matching of ontology concepts used in service description is a critical challenge. If a machine can not understand the meaning of service concepts, it also cannot infer the imply relationships, then the automatic matching and discovery of services is impossible.

2.2 Naming Conventions for Ontologies

Intuitively and in ontology-related work, when ontology terminology is mentioned, it mostly means the terms in thesauri, e.g., Wordnet. On the other hand, the ontology terms defined in applications is very different from the formal words. Generally, the ontology concept used in service semantic descriptions are most compound terms, which are named depending on service developers by their ontology knowledge, experience and wonted. The situation is made worse by the following practices (parts of examples from Fig.1.):

Abbreviations Names are not given in their correct forms, but shortened, e.g. *CalcDistTwoZipsKm*;

Associated words with capitalization or delimiters Words have the form of associations of several words parts (full word or abbreviation) with delimiters, normally a part's first letter capitalized, and sometimes also using underscore, dash or space, e.g., *LogIn, AcctName, ArrivalAirport_In*.

[4] WordNet, an online lexical reference system, $http : //wordnet.princeton.edu/$

[3] The Semantic Web Services Repository at the Smart Media Institute in University College Dublin, $http : //moguntia.ucd.ie/repository/$.

Words with suffix and prefix Examples are *hasFlavour*, *locatedIn*.

Variations or misspelling Names may be variations of word often due to grammatical flexion, e.g., *Booking*, *madeFromGrape*; And defined words are in misspelling format for machine.

Free inventions Any other cases the traditional similarity measures (based, e.g., on WordNet) are prevented to work.

Considering the above compound concept terms, the existing ontology measure algorithms can not work. Moreover, the data clustering algorithm from data mining field can be borrowed to apply to this case. This paper enhances the clustering algorithm in [4] to measure the semantic closeness of composed terms.

3 Ontology Similarity

3.1 Ontology Concept Distance

To semantically measure Ontology concept distance, we should consider both concept structure and concept content. Fortunately, both of these information are prolifically provided by service description. Here, we define the semantic distance dis of the assumed concepts C and D (which could be single formal term or compound term) as:

$$dis = w_1 * Dis_s + w_2 * Dis_i + w_3 * Dis_c, \sum_{i=1}^{3} w_i = 1 \quad (1)$$

where Dis_s is the distance basing on the structure of concept in service Ontology, the Dis_i basing on the common contents shared by concepts, and the Dis_c is only used to measure the compound concept terms by clustering concepts, basing on the concept elements co-occurrence. Formulae 1 not only considers the different concept naming features, but also make up the loss of any single approach, because the service description context is just a structure and a short piece of text, not a corpus or thesaurus.

In the following sections, we will present the detail explanation of every distance measurement.

3.2 Fuzzy-weighted Associative Network

Concepts in a hierarchical taxonomy are all related by certain relationships, based on which concepts can be represented in an associative network consisting of nodes and edges, where nodes denote concepts, edge denotes the binary relationship of the two linked concepts. Also for service description Ontology, such associative network with fuzzy-weighted value on each link can be constructed, in which the similarity of concepts can be measured by the shortest distance as [5] and [17], which is defined as Dis_s in our context.

As the detailed explanation by [5] and correspondence to OWL-Lite, we define four concept relations as ***g***eneralization (e.g., *superclass*), ***s***pecification (e.g., *subclass*), ***n***egative association (e.g., *disjoined*) and ***p***ositive association (e.g., *equivalent*).

Therefore, the distances of arbitrary two nodes in the network can be calculated based on Tables 1–3 [17]. In Table.1 s, g, p and n represent explicit relationships, that is, each two notes relationship can be evaluated basing on triangular norms. τ in Table 2. are the triangular norms (t-norms), which is defined in Table3, where α or β are fuzzy-weighted strength values of relations ($0 \leq \alpha, \ \beta \leq 1$), n is the degree of dependence ($-\infty \leq n \leq \infty$) between the relationships, details please refer to [9]. In the tables those fields are marked with X for which there is no definition. Therefore, the relationship of two arbitrary concepts can easily be inferred by traveling through the associative network.

	g	s	p	n
g	g	p	p	n
s	p	s	p	n
p	p	p	p	n
n	n	n	n	X

Table 1. Kind of paths

	g	s	p	n
g	τ_3	τ_1	τ_2	τ_2
s	τ_1	τ_3	τ_2	τ_2
p	τ_2	τ_2	τ_3	τ_3
n	τ_2	τ_2	τ_3	X

Table 2. Strength of paths

$\tau_1(\alpha, \beta) = max(0, \alpha + \beta - 1)$	$n = -1$
$\tau_2(\alpha, \beta) = \alpha\beta$	$n = 0$
$\tau_3(\alpha, \beta) = min(\alpha, \beta)$	$n = \infty$

Table 3. T-norms function

3.3 Information-theoretical Approach

The definition of similarity between the concepts C and D relates to the concepts' commonality and difference [11]:

$$sim(C, D) = \frac{I(common(C, D))}{I(description(C, D))} = \frac{log\ P(common(C, D))}{log\ P(description(C, D))} \quad (2)$$

where $common(C, D)$ is a proposition that states the commonalities between C and D, $I(common(C, D))$ is the amount of information contained in this proposition and, similarly, $I(description(C, D))$ is a proposition describing what C and D are. In our service context, we refine the similarity expression as follows to calculate the distance Dis_i:

$$Dis_i = \frac{|C \cap D|}{|C \cap D| + \gamma|D/C| + \delta|C/D|}, \qquad \gamma, \delta \in [0, 1] \quad (3)$$

where C and D are two Ontology concept classes of OWL-Lite, $|C \cap D|$ is the number of common elements of C and D, e.g., the number of shared attributes, instances and relational classes, γ and δ are weight values defining the relative importance of their non-common characteristics.

3.4 Ontology Distance for Single Formal Term

Given two single-form Ontology concept terms from two differen Web service description, as t_1 and t_2, which are respectively described by a set of other class terms as their properties, instances and relational members (e.g., *"g,s,n,p"*). There are two cases:

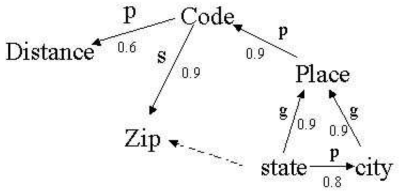


Figure 2. Example of distance in Associative Network

- Two terms t_1 and $t_2$are organized in one hierarchical structure, which is transformed to a fuzzy weighted associative network of Section 3.2.

  Reconsidering the example in Fig. 1, it assumes that in the *Zip* service application domain, terms have the relationships (which are all experimental data, not the real value) cp.Fig. 2. For example, the distance of term *State* and *Zip* is examined, the shortest path is $path = \{State, Place, Code, Zip\}$ with $State \Longrightarrow_{g,0.9} Place$, $Place \Longrightarrow_{p,0.9} Code$ and $Code \Longrightarrow_{s,0.9} Zip.$ So that it hold that $\tau_2(\tau_2(0.9, 0.9)0.9) = 0.729)$, following Table 1-3, that means $State \Longrightarrow_{p,0.729} Zip$, finally we get $Dis_s(t_1, t_2) = 0.729$.

- Terms, t_1 and t_2, are concept classes, respectively consisting of a set of properties and instances as ontology vocabulary according to Section 3.3.

  Assuming that their cardinality are $|t_1| = 9$, $|t_2| = 6$, and they share the number of elements $|t_1 \cap |t_2| = 5$, we obtain $Dis_i(t_1, t_2) = \frac{5}{5+4+1} = 0.5$, where $\gamma, \delta = 0.5$.

3.5 Concept Clustering for Compound Term

Clustering is also a well known approach to group data on the basis of a certain similarity criteria. We adapt this clustering mechanism here to group the compound Ontology concept terms, which are from different service description Ontologies, e.g. *findZipCodeDistance* and *CalcDistTwoZipsKm*, in order to calculate their similarity by the distance dis_c.

In the clustering algorithm, the association rule of two terms t_1 and t_2 is defined as follows [4]:

$$t_1 \longrightarrow t_2(s, c)$$

where, the support s is the probability $s = P(t_1) = \frac{\|T_{t_1}\|}{\|T\|}$ that t_1 occurs in T, $\|T\|$ is the cardinality of the ontology terms' domain, $\|T_{t_1}\|$ is the cardinality of the set which contains t_1, the confidence c is the occurrence probability of t_2 in the case that t_1 occurred, i.e., $c = P(t_2|t_1) = \frac{\|T_{t_1,t_2}\|}{\|T_{t_1}\|}$, with $\|T_{t_1,t_2}\|$ is the cardinality of the set containing both t_1 and t_2. The distance of two terms is weighted by their conditional probability c. The center of a cluster is the term which has the highest occurrence probability of the cluster.

In detail, including the natural language term extraction the clustering algorithm is used by us as follows:

1. Read service description document *.owl*, move all OWL-Lite tags, extract names and parameters, and delete redundancies in the vocabularies. The result is a bag of unique words including composted concept terms, denoting $T = \{t_1, t_2, ...\}$.

2. Preprocess all composted terms in T as follows.

   Suppose that $t_i \in T$ is a composite term, we split it up on the basis of its delimiters, such as capital letters, into several parts. Then, we deal with each part towards extracting the word stem by removing stop words, suffixes and prefixes, restituting abbreviations or correcting misspelling, deleting redundant vocabulary terms and so on, resulting in the set $t_i = \{t_{i1}, t_{i2}...\}$. Substituting t_i by all $t_{ij} \in t_i$ for all i, ultimately yields T'.

3. Compute the values s and c for any two terms in T', store them into a table in descending order, cluster them on the basis of their confidence $c \geq \tau_c$ and support $s \geq \tau_s$ (τ_s and τ_c are thresholds either assigned or obtained experimentally), resulting in the set $T'' = (X_1, X_2, ..., X_k)$ of k clusters.

   Roughly speaking, $X_i, 1 \leq i \leq k$ is a cluster including those terms whose co-occurrence probabilities exceed the threshold τ_c. In traditional agglomeration clustering algorithms, T'' is an intermediate result, while in our context we should improve it in order to find an optimal clustering for our computation. This is the rationale of the algorithm's further steps.

4. In each $X_i \subseteq T''$, remove the frequent and rare parameters to avoid the query expansion and over-fitting problems, which are discussed in the field of information retrieval [8].

5. Split and merge the clusters in T'', in order to wipe off the noise terms and optimize clusters by agglomerating terms according to concentric circularities with different radii.

   The inner circularity consists of those terms, which are, at least, close to half of the other terms. Similarly, the terms in the outer circularity are, at least, close to a quarter of the other ones. They are called them $\frac{1}{2}$ radius [4]. And wiping off the terms, which are not in any circularity.

   For example, to merge two cluster X_1 and X_2, when $\forall i \in X_1 \cup X_2$,

$$\parallel j | j \in X_1 \cup X_2, i \neq j, i \longrightarrow) j(c > \tau_{c1}) \parallel \geq \frac{1}{2}(\parallel X_1 \parallel + \parallel X_2 \parallel -1) \tag{4}$$

Now, when calculating the distance between two random composite terms, here we used c_1 and c_2 distinctively, first, preprocess them using step (2) to obtain $c_1 = \{c_{11}, c_{12}, ...\}$ and $c_2 = \{c_{21}, c_{22}, ...\}$, and then measure their similarity dis_c by the probability of pairs of two terms to occur in the same cluster. As measure the maximum, minimum or or mean may be employed. Here we take the maximum as the optimistic way, the formula is as follows,

$$Dis_c = \begin{cases} max(sim(t_{1i}, t_{2j}) | \forall t_{1i} \in t_1, t_{2j} \in t_2), & if \quad t_{1i}, t_{2j} \in X_k \\ 0, & otherwise. \end{cases} \tag{5}$$

Obviously, such a formula implies as extreme case, that is, all of the sub-terms of c_1 and c_2 have been wiped off as the noise words, such case have no way to scale the distance of ontology concepts. This part of work is right what our experiment will analysis, to evaluate the frequency of its occurrence.

4 Service Similarity

In our previous work [16], a semantic service model for selection is proposed as $s = (NF, F, Q, C)$. By this model, the service selection can happen by filtering single property as Non-functional (in this model, only the service name and service category and short service text description defined as non-function) or combined properties as Non-functional, functional (basing on logical subsumption computing) together with qualities of services. Obviously, either Non-function or function-based based selection, the ontology concept similarity is critical fact for service selection.

Under this selection model, we define an ontology-based service similarity algorithm. Especially when the non-function properties are considered during service selection, because the non-functional related service selection is ontology based.

The idea is to measure the service similarity by the similarity of service name, service operations name, which are defined as Ontology concepts. We do not compare the whole piece service Ontologies, for example $sim_{SO} : (SO_i) \times (SO_j) \rightarrow [0..1]$, where SO_i is the service ontology for service s_i; We only consider how similar two single ontology concepts are in service ontology context, as $sim(c_i, c_j) = \{f(c_i, c_j) \mid c_i \in SO_i \wedge c_j \in SO_j\}$ and the function $f(c_i, c_j) = \min_{k=1,...,j} dis(c_i, c_k)$. Therefore, our work is different from Ontology mapping.

The proposed Ontology-based service selection basically measure by the service name concepts and operations similarity, called lexical semantic level. It is defined as $sim_{Service} = sim_{Concepts} + sim_{oPeration}$, where $sim_{Concept}$ is the sum similarity of all the concepts of services, and $sim_{oPeration}$ is the sum similarity of the operation parameters with their data types.

5 Related Work

Similarity of ontologies has widely been researched, e.g., in the fields of information retrieval, artificial intelligence, databases, and especially in data mining and web mining. Many similarity measures are applied, e.g., Bernstein et al. in [3] use two ways to measure the semantic similarity of objects in an ontology, which are organized in a hierarchical ontology structure, viz., the edge-based [10] (a shorter path from one node to the other) and the node-based [14] (the notion of shared information content) approach. Actually, they present five different distance measures of ontologies, where *ontology distance* stands for the shortest path through a common ancestor in a directed acyclic graph. However, computational degree and weight of edge are not considered. The *vector space approaches* computing cosine or Euclidean distances of k-dimensional vectors [1, 13] do not easily apply to nominal concepts, as it is difficult to represent them as vectors. The *Full-text Retrieval Method (TF/IDF)* is mostly used in information retrieval [1] to compare documents, which are considered as bags of words. However, it is inadequate for structure concepts as semantic relations between them are ignored.

The work most closely related to ours are the studies on ontologies in the semantic web or in semantic web services, such as [7, 4] and [12]. While they consider to cluster the similar terms, and most recur to TF/IDF to measure concept similarity, we follow Dong's notion of name clustering agglomeration algorithms. Maedche et al. also propose an approach to cluster ontology-based data, using the hierarchical clustering algorithm to consider instances of concept similarity. Hau et al. elaborate a metric to measure the similarity of semantic services annotated with OWL ontologies. They mainly depend on the information-theoretic approach to match similar ontology instances. Doan et al. computes the common information content of ontologies to scale their similarity. We combine multiple approaches to adapt to SWS environments. Based on a study of definitions and features of ontologies expressed in OWL, and from a computational point of view, we calculate the distance of two ontologies.

6 Conclusions

Ontology similarity is unquestionable important for Semantic Web Service similarity when we consider the semantic service discovery, selection, composition, and even execution. This paper tries to propose a ontology similarity-based approach to measure service similarity and presents the primary work on it. The contributions of this paper are summarized as, 1) analysis the ontology similarity problem in semantic service context, and classify the ontology concept name features used by service description; 2) present a hybrid ontology concept distance method, and further to measure the service similarity.

As the complexity of ontology-based service similarity, under our model, there is still a lot left for our future work, including the set matching of the ontology-based concept and its type, also the detailed implementation and evaluation. However, fortunately the preliminary experiments show that this new methodology works well.

References

[1] Baeza-Yates, R., Ribeiro-Neto, B., Modern Information Retrieval, ACM Press 1999.

[2] Berners-Lee, T., Hendler, J., and Lassila, O., The Semantic Web, Scientific American, 284(5):34-43, 2001.

[3] Bernstein, A., Kaufmann, E., Buerki, C., and Klein, M., How Similar Is It? Towards Personalized Similarity Measures in Ontologies, International Tagung Wirtschaftsinformatik, February 2005.

[4] Dong, X., Alon Y. Halevy, Madhavan, J., Nemes, E., and Zhang, J., Simlarity Search for Web Services, VLDB 2004.

[5] Fankhauser, P., Kracker, M., and Neuhold, E.J., Semantic vs. structural resemblance of classes, ACM SIGMOD RECORD, vol(20), 4:59-63, 1991.

[6] Gomez-Perez, A., and Corcho, O., Ontology languages for the Semantic Web, Intelligent Systems, IEEE, Jan/Feb pp.54-60, 2002.

[7] Hau, J., Lee, W., and Darlington, J., A Semantic Similarity Measure for Semantic Web Services, Conference *WWW2005*, Japan 2005.

[8] Jones, K.S., Automatic keyword classification for information retrieval, Archon Books, 1971.

[9] Klir, G.L., and Folger, T.A., Fuzzy Sets, Uncertainty and Information, Prentice Hall, 1988.

[10] Lee, J.H., Kim H., and Lee, Y.J., Information Retrieval Based on Conceptual Distance in IS-A Hierarchies. J. of Documentation, 49:188-207, 1993.

[11] Lin, D., An Information-Theoretic Definition of Similarity, Fifteenth International Conference on Machine Learning, 1998.

[12] Maedche, A., and Zacharias, V., Clustering Ontology-based Metadata in the Semantic Web, Pro. of the Joint Conferences ECML'02 and PKDD'02, 2002.

[13] Mitchell, T. M., Machine Learning, McGraw-Hill, New York, 1997.

[14] Resnik, P., Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language, J. of Artificial Intelligence Research, 11:95-130, 1999.

[15] Paolucci, M., Kawmura, T., Payne, T., and Sycara, K., Semantic Matching of Web Services Capabilities, ISWC, Italy, 2002.

[16] Wang, X., Zhao, Y., and Wolfgan, H., Selection Model of Semantic Web Service, 7th International FLINS Conference on Applied Artificial Intelligence, 2006.

[17] Sycara, K., Widoff, S., Klusch, M., and Lu, J.G., LARKS: Dynamic Matchmaking Among Heterogeneous Software Agents in Cyberspace. Autonomous Agents and Multi-Agent Systems, 5:173-203, 2002.

[18] Witten,I., and Frank, E., Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations, Morgan Kaufmann, 1999.

[19] Weinstein, P., and Birmingham, W., Comparing concepts in differentiated ontologies, In Proc. of KAW-99, 1999.