Intelligent Information Management, **2010, 2, 110-119**
doi:10.4236/iim.2010.22014 Published Online February 2010 (http://www.scirp.org/journal/iim)

Sensing Semantics of RSS Feeds by Fuzzy Matchmaking

Mingwei Yuan[1], Ping Jiang[1, 2], Jin Zhu[1], Xiaonian Wang[1]
[1]*Department of Information and Control Engineering, Tongji University, Shanghai, China*
[2]*Department of Computing, University of Bradford, Bradford, UK*
Email: *yuan_mingwei@yahoo.com.cn, p.jiang@bradford.ac.uk,* {*zhujintj, dawnyear*}*@tongji.edu.cn*

Abstract

RSS feeds provide a fast and effective way to publish up-to-date information or renew outdated contents for information subscribers. So far RSS information is mostly managed by content publishers but Internet users have less initiative to choose what they really need. More attention needs to be paid on techniques for user-initiative information discovery from RSS feeds. In this paper, a quantitative semantic matchmaking method for the RSS based applications is proposed. Semantic information is extracted from an RSS feed as numerical vectors and semantic matching can then be conducted quantitatively. Ontology is applied to provide a common-agreed matching basis for the quantitative matchmaking. In order to avoid semantic ambiguity of literal statements from distributed and heterogeneous RSS publishers, fuzzy inference is used to transform an individual-dependent vector into an individual-independent vector. Semantic similarities can be revealed as the result.

Keywords: RSS Feeds, Matchmaking, Multi-Agent, Semantics

1. Introduction

Internet is a complex environment with dynamically changing contents and large-scale distributed users. An incessant research topic for the web based applications is how to acquire information more efficiently and effectively from the Internet. Nowadays there are mainly two approaches. One is user-active that a user visits websites to find interests manually. In order to improve its sear ching efficiency, favourite websites could be bookmarked for later usage. However, manual search or research could be a tedious process for information acquisition. An alternative approach is publisher-active that a user subscribes relevant websites and waits for updates from publishers. It is obvious that the latter mode is more convenient and instant in terms of variant interests and effortless information retrieval. RSS (RDF Site Summary or Really Simple Syndication) feeds are such sources to support automatic information acquisition from the Internet. A user can select which RSS feeds to monitor and avoid unnecessary visit.

RSS is a term used by two independent camps for web content publishing. Therefore, there are two translations about what the acronym RSS stands for: RDF (Resource Description Framework) Site Summary or Really Simple Syndication. Currently the latest versions of the two formats are RSS1.0 (http://web.resource.org/rss/1.0/spec) and RSS2.0 (http://blogs.law.harvard.edu/tech/rss), respectively. The two formats are both XML-based (Extensible Markup Language) and provide similar functions for information updating. RSS1.0 is RDF- compliant so it has more flexible and extendable features than RSS2.0, but RSS2.0 is simpler and more widely used today. In general, the RSS is a metadata language for describing web content changes. Nowadays RSS is adopted by almost all mainstream websites for web content publishing, e.g., web site modifications, news, wiki, and blog updates. But research shows that awareness of RSS is still quite low in the Internet users, 12% of users are aware of RSS, and only 4% have knowingly used RSS [1]. Therefore, it is necessary to develop the RSS based applications to be more conveniently accessible by the ordinary Internet users.

Although a RSS feed is a standardized format with some simple semantics, such as authorship, published date and summary, filtering the received RSS documents using such simple attributes is not able to reflect a user's complex intention. For efficient and effective information acquisition, a user may want to further narrow his/her focus and ignore irrelevant information. This requires a new RSS reader with the features of:

1) semantic awareness

In a distributed web environment, the published information and knowledge can be very complex and diverse. Information acquisition requires a semantics oriented approach that knowledge is represented in a hierarchical data structure. Traditionally agent based information matchmaking often flattens the structure of knowledge into a *free text vector* with simple valued attributes e.g. with keywords, price, delivery time. Semantic relations of knowledge could be lost [2–4].

2) fuzzy sensing

Information acquisition from RSS feeds requires more capability to deal with uncertainties because there is no prior agreement on how information is represented by heterogeneous publishers [4]. Logic based approaches have been widely used to support rule-based matchmaking or consistency checking by proving subsumption and (un) satisfiability [5–6]. However, distributed web applications usually cannot retain a closed-world knowledge base for logic based inference. It would be more realistic to recognise the degree of similarity for flexible matches [7]. It requires more intelligence but less precision, *i.e.* fuzzy sensing.

Today RSS research rests mostly on effective ways to aggregate/syndicate content [8,9] and to improve its applicability [10]. For information acquisition from RSS feeds, current studies often adopt classical text mining methods. In paper [11], fuzzy concept based [12] and word sequence kernels based [13–14] text classifications were applied to measure the similarity of RSS-formatted documents. They intended to reveal similarity by direct comparison of literal texts but ignoring the inherent correlation of individual words, *i.e.* semantic similarity. Due to the autonomy of heterogeneous RSS publishers in an open environment it is impossible to force them to use strictly consistent terminologies and sentences. This introduces ambiguities into text mining of RSS feeds and makes the keyword based vector space model [15] difficult for RSS based applications. Similar text mining approach was used for learning of user preference without consideration of text semantics [16]. Statistical feature selection methods in text mining were also evaluated for classification of RSS feeds corpus [17] and the authors pointed out that topic detection [18] and automatic text classification methods [19] were important to RSS based applications.

It is undoubted that such classical textual analysis methods can be applied to RSS documents since they are formatted textual documents. It should not be ignored that RSS has a close relationship with the semantic web; especially RSS1.0 builds on the RDF. Hence the performance of RSS document classification can be improved by the semantic web technology [20], which takes into account correlations among concepts. Considering semantics, paper [21] developed a weighted schema graph for semantic search of RSS feeds. However, the weights need to be assigned manually. In fact, ontology defines the concepts and correlations in a domain. It provides a powerful tool for semantics based classification, taking into account meaning behind words. The architecture of Personalized News Service (PNS) [22] was proposed, consisting of RSS News Feed Consolidation Service, Ontology Reasoner and Personalized News Service. A user can query interests from RSS feeds based on ontology reasoning. A logic based approach was proposed for implementation.

There have been increasing research interests in recent years addressing the semantic search in distributed applications, especially in peer-to-peer environments e.g. Bluetooth service discovery [22], grid computing [23] and the electronic marketplace [24]. Although description logic can be used for similarity ranking by counting missing/not-implied concept names and loose characteristics between documents [24], the distinguishable granularity is usually coarse and the ability to handle fuzziness and uncertainty is limited. Fuzzy logic has been extended to description logic for representing fuzzy knowledge using continuous membership, such as f-Shin [25] and rule-based f-SWRL [26]. However reasoning for fuzzy description logic still relies on a consistent fuzzy knowledge basis. Ontology can become the knowledge basis for fuzzy reasoning [27].

This paper proposes a quantitative method for information acquisition from RSS feeds with the aid of the semantic web technique. It is an intelligent agent to detect interests for a user. Ontology is used as a semantic bridge linking RSS feeds with a user's intention. Fuzzy matchmaking is carried out for ranking RSS feeds. The method proposed in this paper is for general formats of RSS feeds, and hence RSS 1.0, RSS 2.0 or any other RSS-like formats (e.g. Atom) can be applied. It acts as a real-time sensor of RSS feeds in the Internet with the capability of semantic awareness and fuzzy sensing. First, it transfers received RSS feeds into numerical vectors, *feature vectors*, underpinned by an ontology. Semantic matching of information is conducted by correlation computation. Because distributed publishers may express their opinions using different jargons or words, the obtained numerical vectors are usually individual- dependent. To solve the inherent semantic ambiguity of RSS feeds, fuzzy inference is introduced to transform an individual-dependent vector into an individual- independent vector, so that semantic matchmaking of RSS feeds is accomplished.

This paper is organized as follows, section two proposes the algorithms to extract semantic distance from a domain ontology; section three discusses the method to formulate RSS feeds into ontology instance for facilitating semantics based matchmaking; a job finding agent is developed using the proposed approach in section four. Section five summarizes the proposed method.

2. Semantic Distance between Concepts in Ontology

Information providers publish their information in RSS.

A web user subscribing favorite RSS feeds is often interested in some specific topics. Therefore, irrelevant RSS items should be filtered out. A virtual sensor can be developed for this purpose, which connects with RSS channels and monitors incoming RSS items for those close to the topics semantically.

An illustrative RSS feed from a job publishing site is shown below:

```
   <rss version =”2.0”>
   <channel>
   <title> Yahoo! HotJobs:DVR</title>
   <link>http://hotjobs.yahoo.com/jobs/USA/All/All-jobs</l
ink>
   <description>Top HotJobs results for jobs matching:
DVR</description>
   <webMas-
ter>webmaster-rss@hotjobs.com</webMaster>
   <language>en-us</language>
   …
<item>
      <title>Java Developers - Beta Soft Systems - Fre-
mont, CA USA</title>
      <link>http://pa.yahoo.com/*http://us.rd.yahoo.com/h
      ot-
      jobs/rss/evt=23685/*http://hotjobs.yahoo.com/jobsee
      ker/jobsearch/job_detail.html?job_id=J987065YO
      </link>
      <description> ... to work with our top clients in USA
      belonging to any industry ... .-BS/MS/MBA de-
      gree/Eng. (CS, MIS,CIS,IS,IT,CS... </description>
      </item>
      <item>
      <title>Software Engineer - MPEG, Video, Compres-
      sion - Sigma Designs, Inc. - Milpitas, CA USA</title>
      <link>http://pa.yahoo.com/*http://us.rd.yahoo.com/h
      ot-
      jobs/rss/evt=23685/*http://hotjobs.yahoo.com/jobsee
      ker/jobsearch/job_detail.html?job_id=J497490PV
      </link>
      <description> ... </description>
      </item>
</channel>
</rss>
```

From the example, an RSS feed is composed of a channel and a series of items. Within an item tag pair, a summary of an article or a story is presented. A virtual sensor needs to detect relevant items according to subscriber’s interests. In order to simplify the presentation, only the title of an item is taken into account in this paper, which is a condensed abstract of the content and is the foremost factor influencing information selection. However, the method is applicable to other tags, such as the description tags.

Selecting relevant RSS feeds relies on semantic matchmaking rather than textual matchmaking. A textual or word-to-word comparison makes little sense because distributed and heterogeneous RSS publishers may have totally different writing styles. In fact, words and concepts used in RSS feeds have certain correlations, which can be described by ontology as domain knowledge. The domain knowledge is generally defined as a meta-data model by domain consortia or standard bodies, for example, STEP(Standard for the Exchange of Product model data) AP203[28] and DIECoM (Distributed Integrated Environment for Configuration Management) meta-data model[29] in the domain of product manufacturing. The domain knowledge can then be formulated as a domain ontology in XML using semantic web technologies, e.g. OWL (Web Ontology Language). RSS feeds from distributed sources can be understandable and interchangeable by software agents if the domain ontology is provided.

Suppose domain ontology is defined as

$$\Omega \equiv R(e_1, e_2, ..., e_N) \tag{1}$$

where e_i $(i = 1,...,N)$ denotes entities (concepts, terminologies, properties, attributes) used in a domain; R is the set of relationships between the entities and can be represented as a graph. For example, a domain ontology for DVR (Digital Video Recorder) development is shown in Figure 1, which is edited using Protégé editor (http:// protege.stanford.edu/plugins/owl/index.html) and expressed in OWL(http://www.w3.org/TR/2004/ REC-owl-ref-200 40210/). A DVR is a consumer video/ audio product that can record and play video/audio encoded by using various video/ audio compression standards. The storage media includes hard disks and recordable CD/DVD disks.

Domain ontology provides a semantic bridge for mutual understanding between publishers and subscribers. Concepts defined in ontology may semantically relevant. Hence *semantic distance* is introduced as a measure of semantic difference or similarity between two concepts, which has been applied in semantic web matchmaking [30] and conceptual clustering of database schema [31]. In general, a semantic distance is defined as an application of $E{\times}E$ into R^+, where E is a set of entities in a domain ontology Ω with the following properties:

1) $\forall x \in E, \forall y \in E, \quad \mathrm{Dis}(x, y) = 0 \Leftrightarrow x = y$

2) $\forall x \in E, \forall y \in E, \quad \mathrm{Dis}(x, y) = \mathrm{Dis}(y, x)$

3) $\forall x \in E, \forall y \in E, \forall z \in E \quad \mathrm{Dis}(x, y) \le \mathrm{Dis}(x, z) + \mathrm{Dis}(z, y)$

From Figure 1, it can be observed that an ontology exhibits a hierarchical structure. According to the *visual* distance [31], the semantic distance between two concepts can be calculated as the shortest path length, in which the unit length is assumed to be 1 if two nodes have a direct link. A concept distance matrix to represent semantic differences between two concepts can be obtained by processing the ontology.

First a concept vector is defined, which consists of all entities in the ontology.

$$V(\Omega)=[e_1,e_2, \ldots, e_N]^T \tag{2}$$

To simply computation, the elements in a concept vector are arranged in order by taking a “breadth-first” scan of the ontology hierarchy. A higher level concept

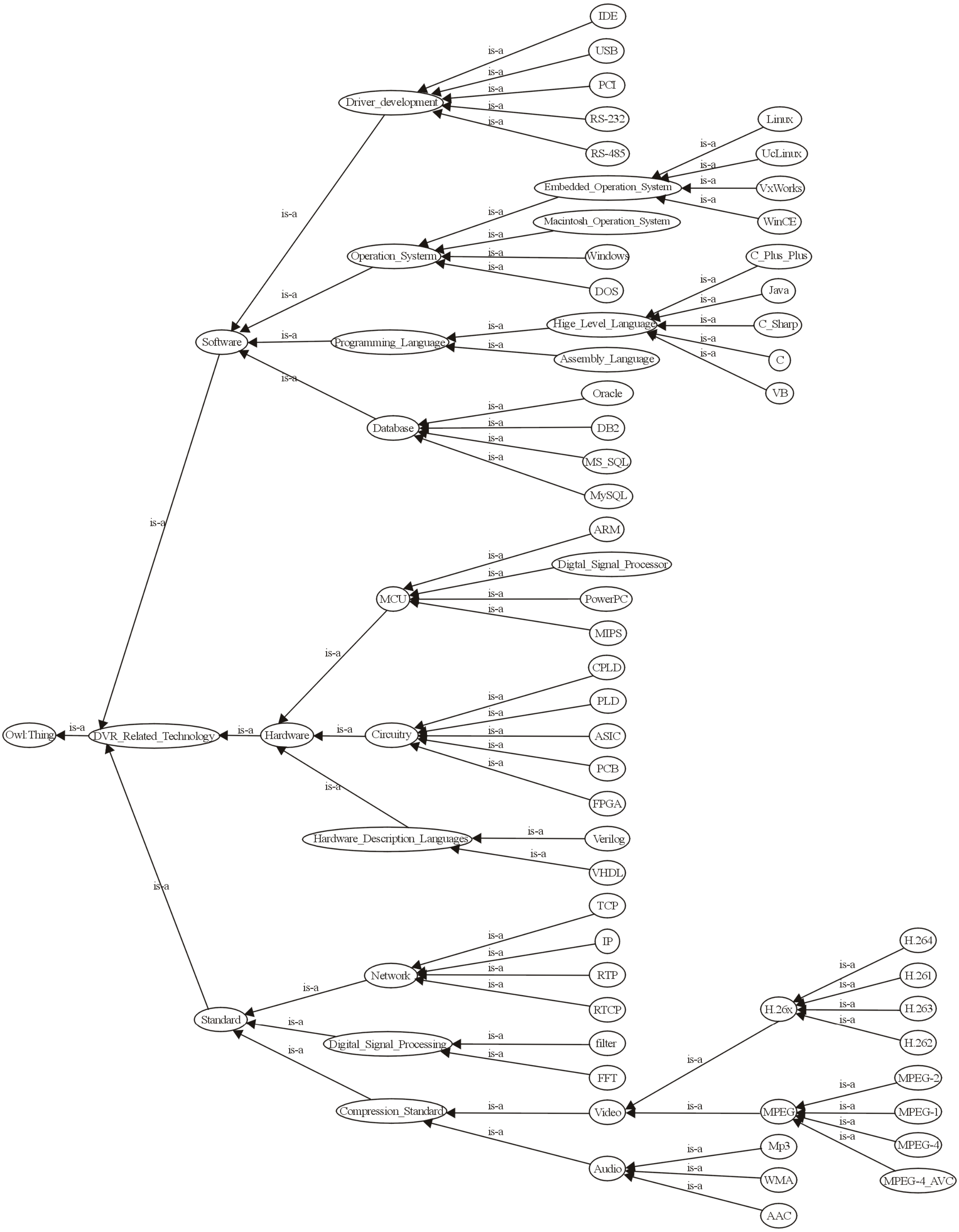


Figure 1. A DVR development ontology represented by OWLViz of protégé.

will be allocated more ahead in the vector. If a concept has multiple father concepts, the first appearance of a concept by following the "breadth-first" scan will be indexed. The following algorithm can be used to convert an OWL document into a concept vector.

// ***Algorithm 1:*** Obtain *a conceptVector from an OWL document and list entities in the Breadth-First order.*

```
BEGIN
  Get root;
  conceptVector (0) = root.Name ;
  pos = 0;      // the index of first concept in every level
  count the childNum of root;
  While childNum > 0
     k=0;      //count the children numbers of level l
     For each entity i in level l
          For each child j of entity i
               k++;
               conceptVector(pos+k) = j.Name;
          Endfor
     Endfor
  pos = pos + childNum;
  childNum = k;
  Endwhile
```

After flattening an ontology graph into a concept vector, a semantic distance matrix can be obtained to depict semantic difference between any two concepts defined in ontology Ω.

//***Algorithm 2:*** Calculate Semantic Distance Matrix

Step 1 Obtain *an N×N initialMatrix=*{ *e*(*i,j*) }*: e*(*i,j*) *denotes the distance value between the* i^{th} *element and the* j^{th} *element which have a direct link.*

The initial relation matrix (*initialMatrix=* {*e*(*i,j*)}*,i=* 1*..N, j=*1*..N*) denotes the semantic distance between two concepts, e_i and e_j, which have a direct link in the ontology. If two concepts have a direct relationship linked by "subClassOf" or "ObjectProperty", a distance 1 or distance 2 is assigned respectively.

$$e(i,j)=\begin{cases}0 & \textit{if } i=j\\ 1 & \textit{if there exists } \text{rdfs:subClassOf } \textit{between i and j}\\ 2 & \textit{if there exists } \text{owl:ObjectProperty } \textit{between i and j}\\ X & \textit{else}\end{cases}, \text{ and}$$

$$e(i,j)=e(j,i).$$

In this initial relation matrix, only the direct links are counted and *X* represents indirect links that will be calculated in step 2 by updating the initial matrix recursively.

Step 2: Obtain *a semantic distance Matrix, DisMatrix* =*{d*(*i,j*)*}: d*(*i,j*) *denotes the distance value between the* i^{th} *element and the* j^{th} *element.*

For ∀ concepts *A*, *B* and *C*,
if *B*= ChildOf(*A*) and C = ¬ DescendantOf(*B*)
  Distance(B,C)=Min(Distance(B,A)+Distance(A,C));
if there is a "SubClassOf" relation between B and A
  Distance(B,A)=1 and
  Distance(B,C) =Min(1+Distance(A,C));
if there is an "ObjectProperty" relation between B and A
  Distance(B,A)=2.

where Distance(*i*, *j*) denotes the graph distance between node *i* and node *j*. Therefore, the following recursive algorithm can produce semantic distances of indirect links between concepts in an ontology graph.

```
BEGIN
Input initialMatrix;
For each column j in initialMatrix
        If ((the element in i^th row)> 0)
        fatherRow(k) = i; // record the index of k-th father if
                              // there is multiple fathers
      Endif
      For each i<j;
      For each father concept k
      d(i,j)= MIN_k  (d(fatherRow(k),i)
+d(fatherRow(k),j) );
      d(j,i)=d(i,j);
        Endfor
      Endfor
Endfor
END
```

The semantic distance matrix can be extracted from an OWL document as:

$$\text{DisMatrix}=\begin{bmatrix}0 & d(1,2) & \cdots & d(1,N)\\ d(2,1) & 0 & \cdots & d(2,N)\\ & & \cdots & \\ d(N,1) & d(N,2) & \cdots & 0\end{bmatrix}, \tag{3}$$

where *d*(*i*, *j*) is the semantic distance between concept e_i and concept e_j.

3. Semantic Matchmaking of RSS Feeds

Since the Internet publishers are distributed and heterogeneous, the literal announcements in RSS have inherent ambiguity and uncertainty due to, for instance, the way to utilize synonyms and jargons. The ambiguity and uncertainty in RSS documents can be alleviated by ontology, which provides a common basis for understanding of frequently used terms and concepts. An RSS feed needs to be parsed into ontology instance first, which is composed of only the concepts defined in the ontology. Hence matchmaking of RSS feeds is transformed to comparison of ontology instances.

At first, the received RSS feeds are preprocessed with the Jena RSS package. The title and the link of each item are obtained. Then concepts defined in the ontology are found out from the title. If a word in the title does not exist in the ontology but is part of a concept in the ontology, the concept will be used as a replacement of this word. For example, *Software*, *MPEG*, *Video*, and *Compression_Standard* are captured from the title of the last item in the example of Section 2, where *Compression Standard*, a concept defined in the ontology, replaces *Compression* found in the title.

Secondly, the concepts extracted from each title are arranged into a hierarchical concept graph, *i.e.* an ontology instance. It has its URI (link) as the root and reorganizes the concepts by retaining all ancestor descendant and sibling relationships in the ontology. For example, "*Software*, *MPEG*, *Video*, *Compression_Standard*" can be transformed into a concept graph as.

Assume that a subscriber of RSS feeds intends to detect information which he/she is interested in. An expression of the interest could be written into an ontology instance in OWL. For example, a job hunter expresses himself as "*Software Engineer*, *experiences in C language*, *Video Compression standard such as H.264, MPEG*"; the ontology instance can be obtained as shown in Figure 3.

Now the information from an RSS feed and a subscriber has been represented formally in accordance with the ontology definition for facilitating semantic matchmaking. As a quantitative description, an ontology instance can be transformed to a numerical vector, *i.e.* feature vector.

***Algorithm* 3:** A feature vector of an ontology instance can be represented as:

$$V(\mathrm{i})=[s_1,s_2,\ \ldots,\ s_N]^T \tag{4}$$

The element s_i in $V(i)$ has a one-to-one correspondence to the concept e_i defined in the ontology concept vector (2). The $s_i \in [0,1]$ indicates the semantic closeness between a concept, e_i, and the root of an ontology instance.

$$s_i = \begin{cases} e^{-\alpha Dis(e_i,root)} & \text{if } e_i \text{ appeares in the instance} \\ 0 & \text{if } e_i \text{ does not appeare in the instance} \end{cases} \tag{5}$$

where Dis(e_i,*root*) is a semantic distance between entity

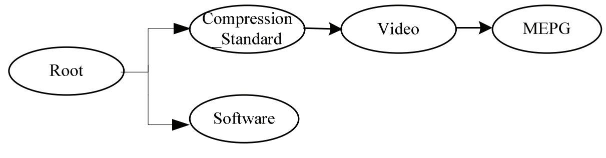


Figure 2. The concept graph of a publisher's ontology instance.

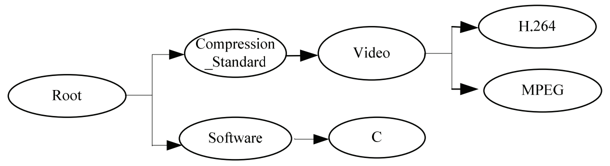


Figure 3. The concept graph of a subscriber's ontology instance.

e_i and the *root*. α is a steepness measure [32] for fuzzy modeling, which is often selected to be -7/MAX(Dis) because $e^{-7}\approx 0$ when Dis(e_i, *root*) reaches its maximum. The semantic distance is computed according to Algorithm 2.

Assume that a user's interest and a received RSS feed can be represented by two feature vectors, V_{user} and V_{rss}. It is expected that the similarities between a user's interest and received RSS feeds are measured by semantic matchmaking, rather than literal matching only. For example an RSS feed from a job-publishing website has information to find "*a Linux developer*". If a user is interested in a job relevant to "*Embedded Operation System*", the published job could be missed by using literal matchmaking. In fact "*Embedded Operating System*" and "*Linux*" have a tight semantic relation that can be observed from the ontology in Figure 1.

Due to the distributed nature of Internet applications, it is impossible to force all information publishers using strictly consistent terminologies and sentences. In fact using words in their RSS feeds relies on their own viewpoint and understanding. So the feature vectors are individual-relevant.

In fact the words or concepts used by publishers are fuzzy; any concept implies some extent of others due to the semantic correlations that can be defined by memberships in fuzzy set theory [33]. Suppose the entities of $\{e_1, e_2, \ldots, e_N\}$ in ontology form a universe of discourse in a community. Any announcement *i* in an RSS feed, such as "*design driver program using C*", can be considered as a linguistic variable. Then the corresponding feature vector $V(i)=[s_1,s_2,\ \ldots,\ s_\mathrm{N}]^\mathrm{T}$ in (4) is a fuzzy representation of *i* from an individual's point of view, where s_i , i=1…N, is a grade of membership corresponding to the i^{th} entity in the universe.

Now an individual-dependent $V(i)$ can be transformed into a fuzzy variable $VI(i)$ that becomes individual-independent by taking account of semantic relationships among concepts ($e_1 \ldots e_N$)

$$V(i) \wedge r(\Omega) \Rightarrow VI(i) \tag{6}$$

where $r(\Omega)$ is a fuzzy relation matrix and each element of r_{ij} reflects correlation or similarity between entity e_i and entity e_j based on ontology Ω. In this case, similar entities can be taken into account even though they are not explicitly cited in an RSS document.

The fuzzy relation $r(\Omega)$ can be obtained from the ontology definition, e.g. in Figure 1. It is the inverse of the distance matrix in (3):

$$r(\Omega_c) = \begin{bmatrix} 1 & r(1,2) & \cdots & r(1,N) \\ r(2,1) & 1 & \cdots & r(2,N) \\ & & \cdots & \\ r(N,1) & r(N,2) & \cdots & r(N,N) \end{bmatrix} \tag{7}$$

where $r(i,j)=e^{-\alpha d(i,j)}$ and α is a steepness measure; *d(i,j)* is calculated by Algorithm 2. As an inverse of semantic distance matrix (3), equation (7) tells us the closeness between two concepts in ontology Ω.

Therefore for any linguistic item *i*, the fuzzy inference can be applied to consider implied semantic relations described by ontology.

***Algorithm* 4:** Obtain *an individual-independent feature vector*:

$$
\begin{aligned}
VI(i) &= V(i) \vee . \wedge r(\Omega) \\
&= \begin{bmatrix} s_1 & s_2 & \cdots & s_N \end{bmatrix} \vee . \wedge \begin{bmatrix} 1 & r(1,2) & \cdots & r(1,N) \\ r(2,1) & 1 & \cdots & r(2,N) \\ & & \cdots & \\ r(N,1) & r(N,2) & \cdots & 1 \end{bmatrix} \\
&= \begin{bmatrix} x_1 & x_2 & \cdots & x_N \end{bmatrix}
\end{aligned} \tag{8}
$$

where $\vee . \wedge$ is an inner product of fuzzy relation, such as max-min composition [33]:

$$
\begin{aligned}
x_i = Max(&\min(s_1, r(1,i)), \min(s_2, r(2,i)), \cdots, \\
&\min(s_N, r(N,i)))
\end{aligned} \tag{9}
$$

Now RSS feeds and a user's interest can be represented as a set of individual-independent vectors. Selecting the interested information from RSS feeds becomes a process of similarity measuring between VI_{rss} and VI_{user}. The following fuzzy operation can be used to filter RSS information.

$$U_i = \frac{|VI_{user} \wedge VI_{rss}|}{|VI_{user}|} \geq \rho, \text{ where } \rho \in [0,1] \text{ is a threshold} \tag{10}$$

where $VI_{user} \wedge VI_{rss}$ is the fuzzy min-operation whose i^{th} component is equal to the minimum of $VI_{user}(i)$ and $VI_{rss}(i)$; $|VI_{user}|$ is the norm of VI_{user} which is defined to be the sum of its components. If every element in VI_{rss} is equal to or greater than that in VI_{user}, then U_i=1 that means VI_{rss} is regarded as a perfect match of VI_{user}. If not, VI_{rss} with U_i higher than ρ will be considered as a close match.

4. An RSS Filter Agent for Job Hunting

An RSS filter agent for job hunting is developed by using the proposed algorithms. Suppose a job hunter wants to find a job via RSS feeds from website http://hotjobs.yahoo.com/jobs/. The job hunter is only interested in the jobs in a specific knowledge domain, for example the DVR developing domain in Figure 1. The job hunter will provide a favorite profile to the agent. The agent will detect relevant RSS feeds and prompt automatically.

The software architecture of the RSS filter agent is shown in Figure 4. The core components are the Quantitative Module, Matchmaking Module, Service Management Module and Ontology. The Service Management Module takes charge of coordination among the modules and configuration of individual modules.

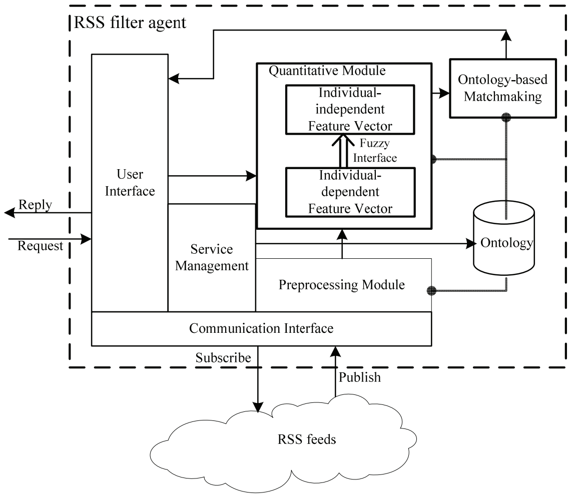


Figure 4. The software architecture of the RSS filter agent.

Protégé(http://protege.stanford.edu/plugins/owl/index.html) was used to construct a domain ontology and exports the OWL document. Jena RSS package (http://jena.sourceforge.net/) was adopted to parse an RSS feed and Jena Ontology API was used to create and parse the OWL document.

The resulted concept vector from Algorithm 1 is as [*DVR_Related_Technology, Standard, Hardware, Software, Compression_Standard, Digital_Signal_Processing, Network, Hardware_Description_Languages, Circuitry, MCU, Operation_System, Programming Language, Driver_Development, Database, Video, Audio, FFT, filter, RTP, TCP, IP,RTCP, VH- DL, Verilog, PLD, CPLD, FPGA, ASIC, PCB, ARM, MIPS, PowerPC, Digtal_Signal_Processor, Embedded_Operation_System, Macintosh_Operation_System, Windows, DOS, High_Level_Language, Assembly_Language, RS- 232, USB,RS-485, IDE, PCI, DB2, MS_SQL, Oracle, MySQL, MPEG, H.26x, AAC, MP3, WMA, WinCE, ucLinux, VxWorks, Linux, Java, C_Plus_Plus, C, VB, C_Sharp, MPEG-4, MPEG -4_AVC, MPEG-2, MPEG-1, H.263, H.264, H.261, H.262*].

The corresponding initial relation matrix is obtained and the distance matrix can then be computed according to Algorithm 2, which is a 70*70 matrix and is illustrated in Figure 5.

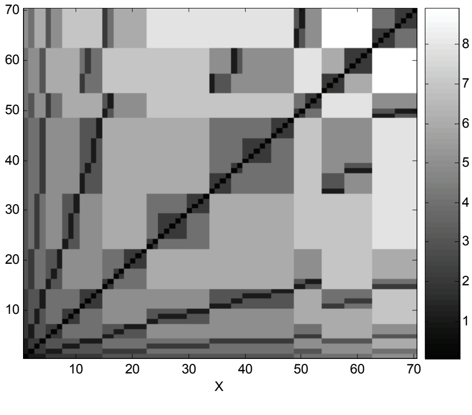


Figure 5. Distance matrix of the DVR developing ontology.

From the graph we can observe that it is a symmetry matrix. The grayscale indicates the semantic distance between an entity in x axis and an entity in y axis, which are entity indexes of the concept vector.

The following job titles were received from http://hotjobs.yahoo.com/jobs/ RSS feeds.

J0: "*Software Engineer, MPEG, Video, Compression*"

J1: "*Senior Firmware Engineer W/ MPEG And ARM*"

J2: "*Software Engineer - Programmer - Developer - C++ - Java*"

J3: "*C/C++/Linux/Oracle Developers*"

J4: "*Embedded Software Engineer –embedded OS, C, Assembly, DSP, Video*"

J5: "*Application Engineer, Audio/Video, Hardware, ASIC, PCB*"

J6: "*MPEG ASIC/Hardware Engineer*"

J7: "*Video Systems, H.264, MPEG, Decoder, FPGA, HDTV*"

There are three users who want to find jobs and announce themselves as:

user0: "*A hardware engineer, experience in using CPLD/FPGA with Verilog for design*"

user1: "*Software Engineer, experience in C language, Video Compression standard such as H.264 , MPEG* "

user2: "*H.264, MPEG, Video, Assembly, FPGA, DSP*"

From these statements, it is easy to observe that they are individual-dependent and do not follow a strict format.

The extracted concept graphs from J0 and J1 have been given in Figure 2 and Figure 3, respectively. The resulted feature vectors to denote the above ontology instances are listed below:

J0: $[0, 0, 0, e^{-\alpha*1}, e^{-\alpha*1}, 0, \ldots, 0, e^{-\alpha*2}, 0, \ldots, e^{-\alpha*3}, 0, \ldots, 0]$

J1: $[0, \ldots, 0, e^{-\alpha*1}, 0, \ldots, 0, e^{-\alpha*1}, 0, \ldots, 0]$

J2: $[0, 0, 0, e^{-\alpha*1}, 0, \ldots, 0, e^{-\alpha*2}, e^{-\alpha*2}, \ldots, 0, 0]$

J3: $[0, \ldots, 0, e^{-\alpha*1}, 0, \ldots, 0, e^{-\alpha*1}, 0, e^{-\alpha*1}, e^{-\alpha*1}, 0, \ldots, 0]$

J4: $[0, \ldots, 0, e^{-\alpha*1}, 0, \ldots, 0, e^{-\alpha*1}, e^{-\alpha*1}, 0, \ldots, 0, e^{-\alpha*1}, 0, \ldots, 0, e^{-\alpha*1}, 0, \ldots, 0]$

J5: $[0, 0, e^{-\alpha*1}, 0, \ldots, 0, e^{-\alpha*1}, e^{-\alpha*1}, 0, \ldots, e^{-\alpha*2}, e^{-\alpha*2}, 0, \ldots, 0]$

J6: $[0, 0, e^{-\alpha*1}, 0, \ldots, 0, e^{-\alpha*2}, 0, \ldots, e^{-\alpha*1}, 0, \ldots, 0]$

J7: $[0, \ldots, 0, e^{-\alpha*1}, 0, \ldots, e^{-\alpha*1}, \ldots, e^{-\alpha*2}, \ldots, e^{-\alpha*2}, 0, 0]$

user0: $[0, 0, e^{-\alpha*1}, 0, \ldots, 0, e^{-\alpha*1}, e^{-\alpha*1}, 0, \ldots, e^{-\alpha*2}, e^{-\alpha*2}, 0, \ldots, 0]$

user1: $[0, 0, e^{-\alpha*1}, 0, \ldots, e^{-\alpha*2}, \ldots, e^{-\alpha*1}, 0, \ldots, 0]$

user2: $[0, 0, 0, 0, \ldots, 0, e^{-\alpha*1}, 0, \ldots, e^{-\alpha*1}, 0, \ldots, e^{-\alpha*2}, 0, \ldots, e^{-\alpha*2}, 0, 0]$

The elements in each feature vector are corresponding to literal labels in the concept vector respectively. α is set to 1.

According to Algorithm 4 and (10), the resulted similarities corresponding to every job J_i are:

user0: [0.0, 0.0, 0.0, 0.0, 0.0, 0.475, 0.475, 0.175]

user1: [0.833, 0.045, 0.333, 0.122, 0.577, 0.122, 0.045, 0.212]

user2: [0.135, 0.098, 0.0, 0.0, 0.634, 0.268, 0.098, 0.465]

The relationships between the published jobs and the announcements of users are illustrated in Figure 6.

From this figure, the agent can identify suitable jobs for users. For example, the most relevant jobs for user0 are J5 and J6.

As a general illustration, Figure 7 shows the user interface for configuration of an RSS filter agent. A user registered as Tom and subscribed RSS feeds from Yahoo hotjobs (http://hotjobs.yahoo.com/rss/0/USA/-/-/-/IT), UK academic employment (http://www.jobs.ac.uk/rss/disc/2516. xml), and CareerBuilder (http://rtq.careerbuilder.com/RTQ/rss20. aspx? lr=cbcb_ct&rssid= cb_ ct_ rss_ engine&cat=JN004&state=IL&city=chicago) for job

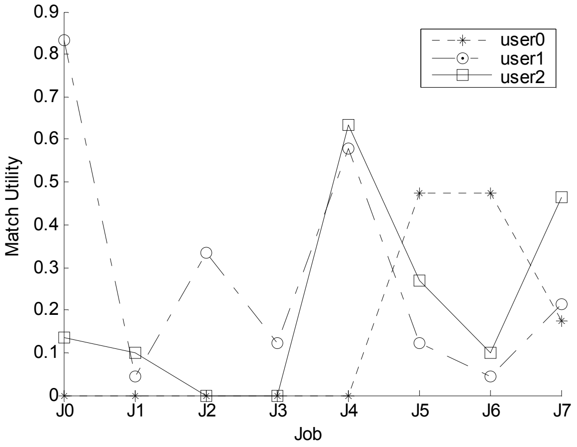
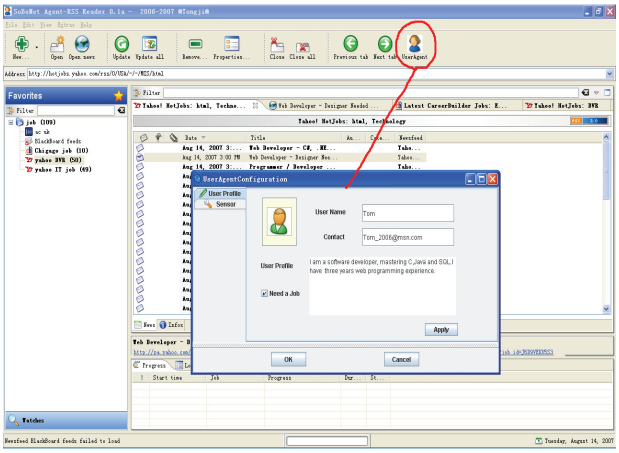


Figure 6. Jobs vs users.



Address http://hotjobs.yahoo.com/rss/0/USA/-/-/MIS/html

Date	Title	Newsfeed
Aug 14, 2007 3:...	Web Developer - C#, .NE...	Yaho...
Aug 14, 2007 3:00 PM	Web Developer - Designer Nee...	Yahoo...
Aug 14, 2007 3:...	Programmer / Developer ...	Yaho...

User Name: Tom

Contact: Tom_2006@msn.com

User Profile: I am a software developer, mastering C,Java and SQL.I have three years web programming experience.

☑ Need a Job

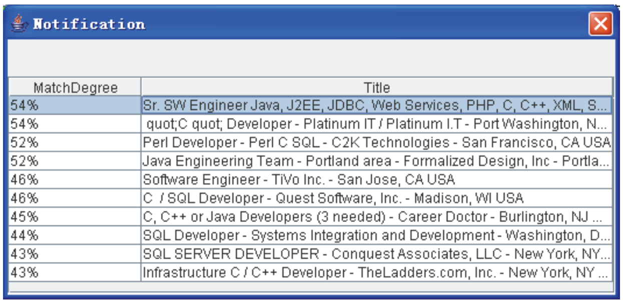
Web Developer - D

http://pa.yahoo.com/ ... job id=J6D9VEKU5S3

Newsfeed BlackBoard feeds failed to load

Tuesday, August 14, 2007

Figure 7. RSS filter agent registration.



MatchDegree	Title
54%	Sr. SW Engineer Java, J2EE, JDBC, Web Services, PHP, C, C++, XML, S...
54%	quot;C quot; Developer - Platinum IT / Platinum I.T - Port Washington, N...
52%	Perl Developer - Perl C SQL - C2K Technologies - San Francisco, CA USA
52%	Java Engineering Team - Portland area - Formalized Design, Inc - Portla...
46%	Software Engineer - TiVo Inc. - San Jose, CA USA
46%	C / SQL Developer - Quest Software, Inc. - Madison, WI USA
45%	C, C++ or Java Developers (3 needed) - Career Doctor - Burlington, NJ ...
44%	SQL Developer - Systems Integration and Development - Washington, D...
43%	SQL SERVER DEVELOPER - Conquest Associates, LLC - New York, NY...
43%	Infrastructure C / C++ Developer - TheLadders.com, Inc. - New York, NY ...

Figure 8. Virtual sensor output.

hunting. A user profile, "*I am a software developer, mastering C, Java and SQL. I have three years web programming experience*", was announced as the interest. The parameters of the virtual sensor were set as a sensor resolution of 56% and display of top 10 candidates. The detected RSS feeds were shown in Figure 8, where candidates were identified and the match degrees were shown.

The method proposed in this paper can be applied for both content publishers and content subscribers. For example, on a business website side, the agent can be used to find potential customers and, on a user side, the agent can be used to perceive favorite information up to a certain match-level by adjusting the threshold of ρ .

5. Conclusions

In this paper, the user-oriented active choice of information from RSS feeds was discussed. A fuzzy method for matchmaking between a subscriber's interest and RSS items was proposed, which converted headlines of an RSS feed into numerical vectors and semantic closeness to the interest was measured. Ontology acted as a bridge to link heterogeneous publishers with subscribers in semantics rather than in words. Concept graphs were firstly extracted from the title of each RSS item and transformed into an individual-dependent feature vector with the aid of ontology. In order to eliminate ambiguity due to different literal expression of individuals, fuzzy inference was applied to obtain the grade of membership in terms of ontology, which became an individual independent feature vector. A job seeking agent was developed to illuminate the method and the result showed its validity.

6. References

[1] J. Grossnickle, T. Board, B. Pickens, and M. Bellmont, "RSS-crossing into the mainstream," October 2005. http://publisher.yahoo.com/rss/RSS_whitePaper1004.pdf.

[2] D. Kuokka and L. Harada, "Integrating information via matchmaking," Journal of Intelligent Information Systems, Kluwer Academic Publishers, Vol. 6, pp. 261–279, 1996.

[3] K. Kurbel and I. Loutchko, "A model for multi-lateral negotiations on an agent-based marketplace for personnel acquisition," Electronic Commerce Research and Applications, Vol. 4, No. 3, pp. 187–203, 2005.

[4] R. Hishiyama and T. Ishida, "Modeling e-procurement as co-adaptive matchmaking with mutual relevance feedback," M. Barley and N. K. Kasabov (Eds.): Intelligent Agents and Muli-Agent Systems, the 7th Pacific Rim International Workshop on Multi-Agents (PRIMA'04), Auckland, New Zealand, pp. 67–80, 2004.

[5] D. Trastour, C. Bartolini, and C. Preist, "Semantic web support for the business-to-business e-commerce pre-contractual lifecycle," Computer Networks, Vol. 42, pp. 661–673, 2003.

[6] J. Kopena and W. C. Regli, "DAMLJessKB: A tool for reasoning with the semantic web," IEEE Intelligent Systems, Vol. 18, pp. 74–77, 2003.

[7] S. A. Ludwig and S. M. S. Reyhani, "Introduction of semantic matchmaking to grid computing," Journal of Parallel and Distributed Computing, Vol. 65, pp. 1533–1541, 2005.

[8] D. Sandler, A. Mislove, A. Post, and P. Druschel, "Feedtree: Sharing web micronews with peer-to-peer event notification," In Proceedings of the 4th International Workshop on Peer-to-Peer Systems (IPTPS'05), Ithaca, NY, USA, pp. 141–151, 2005.

[9] B. Hammersley, "Content syndication with RSS,' O' Reilly, ISBN: 0-596-00383-8, 2003.

[10] E. Jung, "UniRSS: A new RSS framework supporting dynamic plug-in of RSS extension modules," In Proceedings of the 1st Aisan Semantic Web Conference (ASWC'06), Beijing, China, pp. 169–178, 2006.

[11] K. Wegrzyn-Wolska and P. S. Szczepaniak, "Classification of RSS-formatted documents using full text similarity measures," In Proceedings of the 5th International Conference on Web Engineering (ICWE'05), Sydney, Australia, pp. 400–405, 2005.

[12] P. S. Szczepaniak and A. Niewiadomski, "Clustering of documents on the basis of text fuzzy similarity," Abramowicz W. (Eds.): Knowledge-based Information Retrieval and Filtering from the Web, pp. 219–230, Kluwer Academic Publishers, 2003.

[13] N. Cancedda, E. Gaussier, C. Goutte, and J. Renders, "Word-sequence kernels," Journal of Machine Learning Research, 3, pp. 1059–1082, 2003.

[14] H. Lodhi, N. Cristianini, J. Shave-Taylor, and C. Watkins, "Text classification using string kernel," Advances in Neural Information Processing System, Vol. 13, pp. 563–569, 2001.

[15] G. Salton and M. J. McGill, "Introduction to modern information retrieval," McGraw-Hill, New York, 1983.

[16] J. J. Sampera, P. A. Castillob, L. Araujoc, J. J. Merelob, O. Cordon, and F. Tricas, "NectaRSS, an intelligent RSS feed reader," Journal of Networks and Computer Applications, Vol. 31, pp. 793–806, 2008.

[17] R. Prabowo and M. Thelwall, "A comparison of feature selection methods for an evolving RSS feed corpus," Information Processing and Management, Vol. 42, pp. 1491–1512, 2006.

[18] N. S. Glance, M. Hurst, and T. Tomokiyo, "BlogPulse: Automated trend discovery for weblogs," In Proceedings of the 13th International WWW Conference: Workshop on Weblogging Ecosystem: Aggregation, Analysis and Dynamics, New York, USA, pp.1–8, 2004.

[19] Y. Yang and J. O. Pedersen, "A comparative study on feature selection in text categorization," In Proceedings of the Fourteenth International Conference on Machine Learning (ICML'97), San Francisco, USA, pp. 412–420, 1997.

[20] T. Berners-Lee, "Semantic web road map," 1998. http:// www. w3.org/DesignIssues/Semantic.html.

[21] X. Ning, H. Jin and H. Wu, "RSS: A framework enabling ranked search on the semantic web," Information Processing and Management, Vol. 44, pp. 893–909, 2008.

[22] S. Avancha, A. Joshi, and T. Finin, "Enhanced service discovery in Bluetooth," Communications, pp. 96–99, 2002.

[23] S. A. Ludwig and S. M. S. Reyhani, "Semantic approach to service discovery in a grid environment," Journal of Web Semantics, Vol. 4, pp.1–13, 2006.

[24] S. Colucci, T. D. Noia, and E. D. Sciascio, F. M. Donini, M. Mongiello, "Concept abduction and contraction for semantic-based discovery of matches and negotiation spaces in an E-marketplace," Electronic Commerce Research and Applications, Vol. 4, pp. 345–361, 2005.

[25] G. Stoilos, G. Stamou, V. Tzouvaras, J. Z. Pan, and I. Horrocks, "The fuzzy description logic f-SHIN," International Workshop on Uncertainty Reasoning For the Semantic Web, 2005.

[26] J. Z. Pan, G. Stoilos, G. B. Stamou, V. Tzouvaras, and I. Horrocks, "f-SWRL: A fuzzy extension of SWRL," Journal on Data Semantics, Vol. 6, pp. 28–46, 2006.

[27] P. Jiang, Q. Mair, and Z. Feng, "Agent alliance formation using ART-networks as agent belief models," Journal of Intelligent Manufacturing, Vol. 18, pp. 433–448, 2007.

[28] ISO, "Application protocol: Configuration controlled design," IS 10303 – Part 203, 1994.

[29] P. Jiang, Q. Mair, and J. Newman, "The application of UML to the design of processes supporting product configuration management," International Journal of Computer Integrated Manufacturing, Vol. 19, pp. 393–407, 2006.

[30] K. P. Sycara, M. Klusch, S. Widoff, and J. Lu, "Dynamic service matchmaking among agents in open information environments," ACM SIGMOD Record (ACM Special Interests Group on Management of Data), Vol. 28, pp. 47–53, 1999.

[31] J. Akoka and I. Comyn-Wattiau, "Entity-relationship and object-oriented model automatic clustering," Data and Knowledge Engineering, Vol. 20, pp. 87–117, 1996.

[32] J. Williams and N. Steele, "Difference, distance and similarity as a basis for fuzzy decision support based on prototypical decision classes," Fuzzy Sets and Systems, Vol. 131, pp. 35–46, 2002.

[33] L. A. Zadeh, "Fuzzy sets," Information and Control, Vol. 8, pp. 338–353, 1965.