ISSN: 2348 - 2117

International Journal of Engineering Technology and Computer Research (IJETCR)
Available Online at www.ijetcr.org
Volume 3; Issue 6; November-December-2015; Page No. 87-93

Design Ontology with Semantic Web Service Innovation for Bootstrapping Approach using Improved Enhanced Traversal Algorithm

Sasanka Bhimavaram[1], Dr.P.Govindarajulu[2]

[1]Research Scholar, Department of Computer Science, Sri Venkateswara University, Tirupati, Andhra Pradesh, India

sasankabimavaram@gmail.com

[2] Professor, Department of Computer Science, Sri Venkateswara University, Tirupati, Andhra Pradesh, India

pgovindarajulu@yahoo.com

Abstract

Web Information Extraction is essential to many application areas, such as ontology mapping and bootstrapping Semantic Web etc. This is a stimulating task, because information mining is typically a noisy effort whereas ontologies generally involve hygienic and crispy data. We discussed approaches that extract ontology in this spirit from the relational databases and also present methodologies that objective to extract ontology from documents or by extension from the entire web then implemented with the improved enhanced traversal algorithm with bootstrapping approach applied on the databases such as bank and hospital to obtain better accuracy performance for registered services. We have shown the ontology creation and web service registration procedure for bank and hospital databases. Based on these components of the research, we have build an object oriented model for an application domain and use this model for sharing and reusing domain information on the Web.

Key Words: Ontology Generation, Semantic Web Services, Bootstrapping Approach, Improved ET Algorithm, WSDL.

INTRODUCTION

Ontology is a formal representation of the world. It defines a set of representational primitives that are relevant for modeling a domain of knowledge or discourse. The representational primitives typically consist of a set of concepts or entities within a domain, relationships that may exist among these concepts and properties. The Internet has become very popular nowadays. Millions of people access the Web to search information, do online shopping, get entertained or just learn. From its early stages, the Web has provided a magnificent opportunity for persons, businesses or communities that want world wide exposure. An ontology may include individuals, instances, classes, attributes, relations, restrictions, rules, and axioms. Based on these components, we can build an object oriented model for an application domain and use this model for sharing and reusing domain information on the Web. The secure of semantics and challenges in escalating semantic techniques are new to researchers in the database and information system field either. For instance, semantics has been studied or applied in the context of data modeling, query and transaction processing, etc. We review some applications developed using business technologies to present insights into what Semantic and Web Technology can do at present[4,9]. Based on the increasing difficulty and the deeper role of semantics, we partition the applications into three types.

1. Semantic search and contextual browsing
2. Semantic integration
3. Analytics and Knowledge Discovery

Ontology's, which are used in order to sustain interoperability and regular concerned between the different parties, are a key module in solving the complexity of semantic heterogeneity, thus enabling semantic interoperability among divergent web applications and services. This is why several newer approaches have embarked to extract ontological information from the entire Web.

- Information Extraction/Mining for Ontologies
- Information Extraction from the Web

The relational databases contains already much information, it contains only part of the huge amount of data that is available on the Web. This is why several newer approaches have embarked to extract ontological information from the entire Web. The Web is much more heterogeneous than databases, with different file formats, different languages, different page layouts, and only creeping standardization. Furthermore, the information on the Web exhibits various degrees of credibility. Data may be faulty, incomplete, contradictory or wrong. In addition, the Web is one of the largest computer processable resources at all[13,17]. Web Service innovation refers to reusing existing services in order to provide novel, improved ones.

BACK GROUND:

Ontology Learning from Databases: Early methods exclusively based on the transformation of database schemas often result in incomplete ontologies that need to be further refined at the cost of huge manual post-editing efforts. Such manual tasks might be deemed too tedious and costly by many practitioners. To provide an extended automated support to facilitate the production of high quality ontologies from databases, adequate ontology learning methods should be elaborated.

We give in this section an overview through selected examples of the structuring patterns that will be further explored in this chapter. We introduce here some interesting structuring patterns without paying too much attention to how they can be automatically identified and used to generate appropriate ontology fragments. Some of the most relevant content-driven transformation techniques will be addressed in next sections and we show how to exploit them in order to complement ontologies derived from database schemas[7,10].

A. Preliminary Definitions:

A relational database schema D is defined as a finite set of relation schemas D= $\{R_1,\ldots,R_n\}$ where each relation schema R_i is characterized by its finite set of attributes $\{A_{i1},A_{i2},\ldots,A_{im}\}$. A function pkey links to every relation its primary key which is a set of attributes K⊆ R.A relation r on a relation schema R is a set of tuples which are classifications of |R| values. Correspondingly, a database d on D is defined as a set of relations d = $\{r_1,r_2,\ldots,r_n\}$. By resolution, if a relation schema is represented by a capital letter, the corresponding lower case letter denotes an instance of the relation schema. A projection of a tuple t on a set of attributes X⊆R, denoted t[X], is a restriction on t, resulting in the subsequence with values corresponding to attributes of X. the projection of a relation r on X, denoted ? ? (?), is defined by ? ? (?) = {t{X]|t€ r}. The concept of inclusion dependency is used to account of correlations between relations. An inclusion dependency is an expression R[X] ⊆ S[Y] where X and Y are respectively attribute arrangements of R and S relation schemas with the constraint |X| = |Y|. The dependency holds between two instances r and s of the relation schemas for each tuple u in r there is a tuple V in S such that u[X] = v[Y]. Informally, an inclusion dependency is a suitable approach to state that data items are derivative from a new relation. Foreign key relationship can be defined as additional dependencies filling the additional property: Y= pkey(S). The notation R[X] ⊆ S[pkey(S)] will be used for these definite dependencies. Proper metaphors of ontology remains are articulated in OWL theoretical syntax [2,8].

2. DESIGN AND IMPLEMENTATION PROCESS OF PROPOSED METHOD:

A. Ontology Bootstrap equation:

The best known application of the bootstrap is to estimating the mean, μ say, of a population with distribution function F, from data drawn by sampling randomly from that population. Now,

$\mu = \int x \, dF(x)$

The sample mean is the same functional of the empirical distribution function, i.e. of

$$F(x) = \frac{1}{?} \sum_{?? \, ?}^{?} I(X_? \leq x)$$

Where$X_1, \ldots, X_n$ denote the data. Therefore the bootstrap estimator of the population mean, μ is the sample mean.

$\mu = \int x \, dF(x) = \frac{?}{?} \sum_{?? \, ?}^{?} X_?$

Likewise, the bootstrap estimator of a population variance is the corresponding sample variance; the bootstrapping estimator of a population correlation coefficient is the corresponding empirical correlation coefficient; and so on.

More generally, if $\theta_0 = \theta(F)$ denotes the true value of a parameter ,where θ is a functional, Then $\theta_0 = \theta$ (F) is the bootstrap estimator of θ_0.

B. Performance Similarity:

We use for effectiveness of proposed system from parameters of the accuracy which is the probability of the services retrieved that are significant to the user's information requirement and recall is the probability of the services that are significant to the query that are effectively retrieved.

A={relevant_Webservices}
B={retrived_Webservices}

Precision= $\frac{|? \cup ?|}{|?|}$|

Recall= $\frac{|? \cap ?|}{|?|}$

C. Traversal Method:

In order to avoid many of the "brute force" method comparisons and in its place of testing the new element c blindly with all elements; in the top search phase of "simple traversal" method, c is pushed down the tree, and in the bottom search, phase c is pushed up, stopping when immediate predecessors or successors have been determined.

D. Top Search Method:

Algorithm1: Top Search Phase of the Traversal Method
Step 1: Top_search(c,x)
Step 2: mark(x, "visited")
Step 3: for all y € successors(x) do
Step 4: if top_subs(y,c) then
Step 5: if pos-succ→pos-succ {y}
Step 6: result →x
Step 7: else
Step 8: for all y € pos-succ do
Step 9: if y not marked as "visited"
Step 10: then result→result Ŭ top_search(c,y)
Step 11: stop

Algorithm 2: Top subsumption of the traversal method
Step 1: Top_sub(y,c)
Step 2: if y marked as 'positive' then
Step 3: result→true
Step 4: else if y marked as 'negative' then
Step 5: result→false
Step 6: else if subs(y,c) then
Step 7: mark(y, 'positive')
Step 8:result→true
Step 9:else
Step 10:mark(y, 'negative')
Step 11:result→false
Step 12: Stop

E. Improved Top search phase of the "Improved Enhanced Traversal" Algorithm:

Step 1: Enhanced_Top_Subs(y,c)= Σ_π,w
Step 2: if y marked as 'positive' then
Step 3: result of w1≡ w2 → true
Step 4: else if y marked as 'negative' then
Step 5: result → false
Step 6: else if for all z € predecessors(y)
Step 7: always Enhanced_Top_Subs(z,c)
Step 8: subs(y,c) then w1 = w2
Step 9: mark (y, 'positive')
Step 10: result R⊆ Σ_π → true
Step 11: else if mark(y, 'negative')
Step 12: else σ: w→ Σ_π
Step 13: result → false

F. Ontology Evolution:

Ontology evolution is the useful module where, the descriptor is more authenticated using the textual service descriptor. The analysis is based on the improvement that a Web service can be divided into two descriptions: the WSDL description and a textual description of the Web service in free text. The WSDL descriptor is analyzed to extract the context descriptors and possible concepts as described [4,11].

G. Domain Extraction:

In this module we extend the data extraction procedure using Web service that allows domain particulars to be recognized based on the domain name, that maintains a web services associated with operations and services. Finally it extracts the URL's list for user specific domains and provides those URL's to access.

H. Creating Required Tables:

The CREATE TABLE command can also be entered at the mysql> prompt or can be written into a file and sent into MySQL. The latter is preferable because you keep hold of records of how created the tables. The tables may be created as follows:

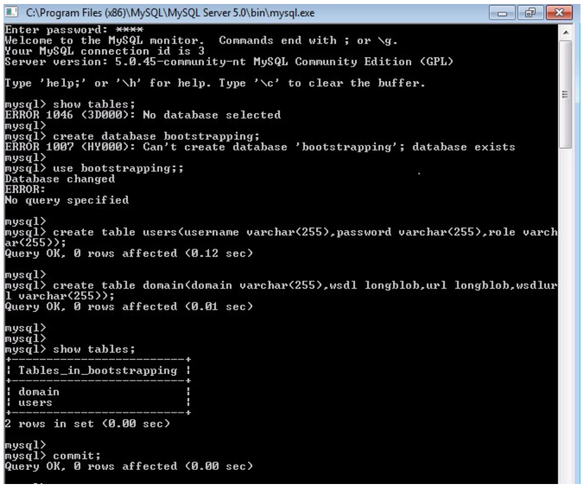

Figure 1: Creation of Tables for Ontology

I. Adding data to MySQL:

Once you have created tables, you can start filling it with data. One easy way of adding a lot of data is by using the mysql import system command. This will read in a text file where data for each table row are divided by newlines and data for every column are separated by tabs and in the identical order as the columns were defined. The file should be named according to the principle tablename.txt.table, replacing "tablename" appropriately.

J. Activity Diagram:

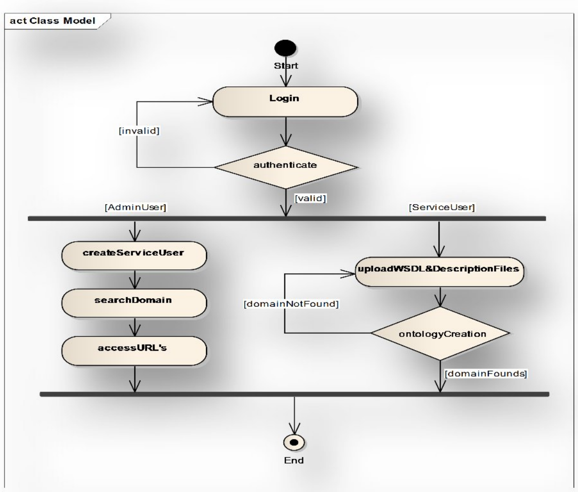

Figure 2: Activity diagram for ontology creation

K. Use Case Diagram:

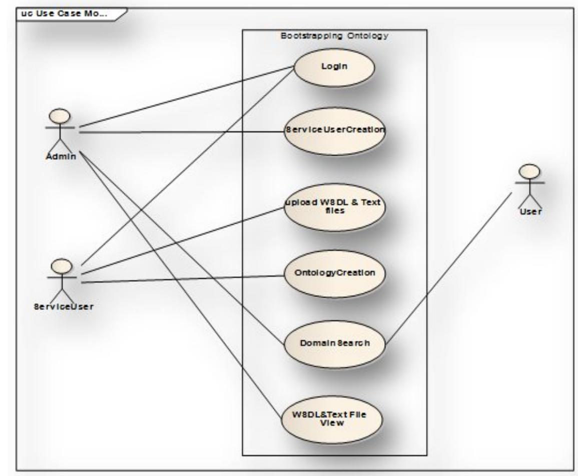

Figure 3 (A): Use Case Diagram for Admin and Service user

L. Class Diagram:

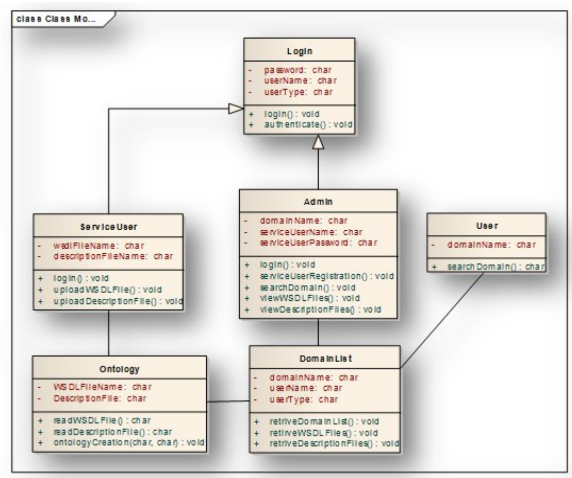

Figure 3 (B):

3. WEB SERVICE AND ONTOLOGY INVENTION WITH PROPOSED METHOD RESULTS:

Ontology languages have been developed for semantic knowledge management. Because XML and WSDL serves for exchanging data between parties who have agreed to the definitions beforehand, it is not enough to express the meaning of concepts and to organize the relations between concepts in an application domain. Therefore, it is necessary to use Web ontology languages which can support the richer expressiveness. OWL is a main Web ontology language which satisfies the requirements of building a domain ontology, including a well-defined syntax, a well-defined semantics, efficient reasoning support, sufficient expressive power, and convenience of expression. The proposed Improved Enhanced

Traversal Algorithm is applied for web services and ontology generation of bank and hospital databases with bootstrapping approach. The following screens show the procedure for web service registration and generation of ontology's with WSDL.

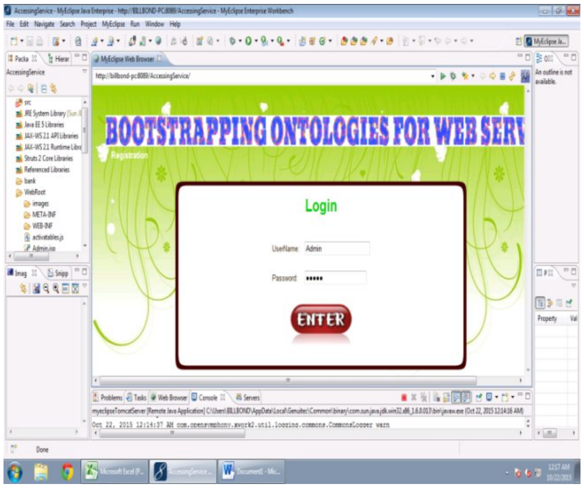

Figure 4: Admin Login Page for Bootstrapping Ontology

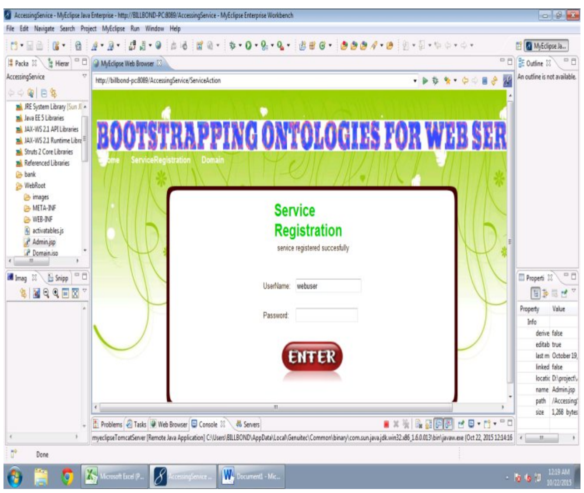

Figure 5: Service Registration Process

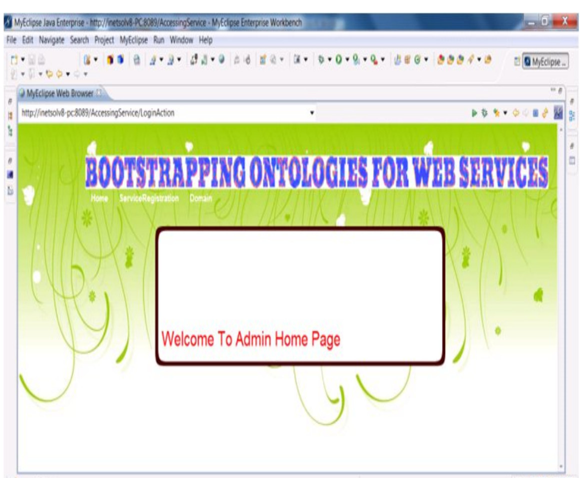

Figure 6: Welcome to Admin Home Page

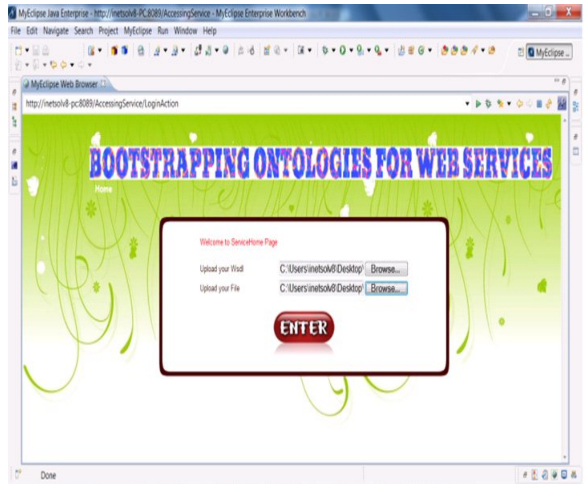

Figure 7: Welcome to Service Home page

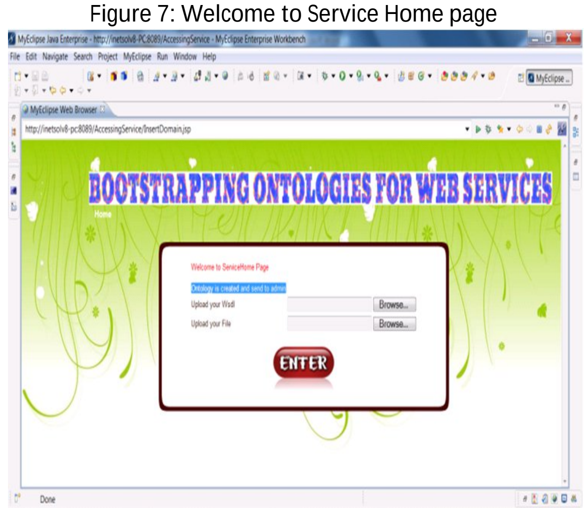

Figure 8: Ontology Created Screen

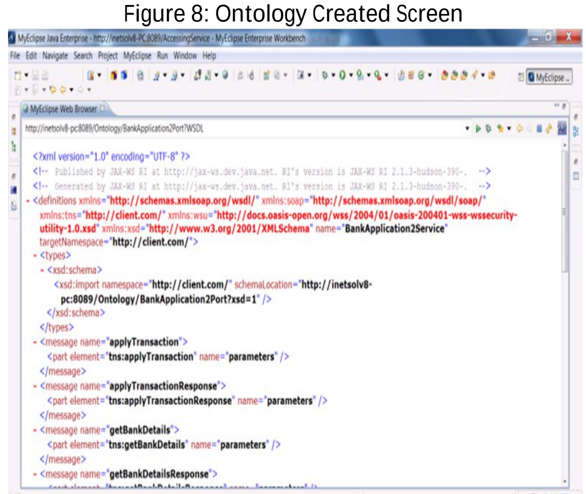

Figure 9: Created Ontology with WSDL

4. CONCLUSION:

We are applied the top search method to enhanced traversal equation with anticipated structural design which has been instantiated with WSDL the application of our approach will emerge in the Semantic Web, in the area of semantic annotation and metadata generation. We deliberated about ontology learning for relational databases with a focus on methods for the identification of semantic patterns in the stored data. Ontology learning in this context should not be conceived as an endogenous process that only aims at discovering domain semantics from the data and metadata of some source legacy databases. In this research, we applied improved enhanced traversal algorithm with bootstrapping mechanism to relational databases such as bank application service and hospital data for constructing the web services to permit web service clients to discover related services easily for finding the best set of semantic web services. It was shown that the proposed system gives better accuracy performance for bootstrapping ontology creation and semantic web services. Moreover, Interoperability of the resulting ontologies can significantly be increased by mapping the application specific concepts extracted from these databases to equivalent or closely related concepts from widely shared reference ontologies. The use of Semantic Web Services provides the flexibility required in the execution phase. At runtime goals can be bound to explicit Semantic Web Services preferred on the basis of the existing conditions and informed by contextual knowledge, which includes monitoring data. Furthermore, since services are described semantically, both functional and non functional properties are properly represented using domain ontologies.

In future work, we will incorporate to discover complex axioms from these heterogeneous sources is a major challenge for bootstrapping ontology learning and web usage mining systems with different set of semantic web services.

5. REFERENCES:

1. Andrew D. Bagdanov, Marco Bertini, Alberto Del Bimbo, Giuseppe Serra and Carlo Torniai. "Semantic annotation and retrieval of video events using multimedia ontologies", IEEE Computer Society ,the International Conference on Semantic Computing, pages 713–720, Washington, DC, USA, 2007.
2. A. Gomez-Perez and O. Corcho. "Ontology languages for the semantic web", IEEE Intelligent Systems, 17(1):54–60, 2002.
3. Amjad Umar and Adalberto Zordan. "Enterprise Ontologies for Planning and Integration of Business: A Pragmatic Approach", IEEE Transactions on Engineering Management, vol. 56, no. 2, pages 352–371, 2009.
4. C. Golbreich, S. Zhang, and O. Bodenreider. "The foundational model of anatomy in OWL: Experience and perspectives", *Journal of Web Semantics*, 4(3):181–195, 2006.
5. Chiu, D.-Y., Wu, Y.-H., and Chen, A. L. P." An Efficient algorithm for mining frequent sequences by a new strategy without support counting", In ICDE, pages 375 386. IEEE Computer Society, 2004.
6. D. Gessler , G. Schiltz, G. May, S. Avraham, C. Avraham, D. Grant and R. Grant. "SSWAP: A Simple Semantic Web Architecture and Protocol for semantic web services", *BMC Bioinformatics*, 10(1):309, 2009.
7. Dai, H. & Mobasher, B. "Integrating Semantic Knowledge with Web Usage Mining for Personalization in A. Scime (ed.), *Web Mining: Applications and Techniques"*, IGI Global, Hershey, PA, USA, pp. 276 - 306, 2005.
8. E. Al-Masri and Q.H. Mahmoud, "Investigating Web Services on the World Wide Web", Proc. Int'l World Wide Web Conf. (WWW ’08), 2008.
9. Gomadam K., Ranabahu A. and Sheth A. SA-REST: Semantic Annotation of Web Resources. Member submission, W3C, April 2010.
10. G. Zhang, A. Troy, and K. Bourgoin, "Bootstrapping Ontology Learning for Information Retrieval Using Formal Concept Analysis and Information Anchors", Proc. 14th Int’l Conf. Conceptual Structures (ICCS ’06), 2006.
11. H. Davulcu, S. Vadrevu, S. Nagarajan, and I. Ramakrishnan, "OntoMiner: Bootstrapping and Populating Ontologies from Domain Specific Web Sites", IEEE Intelligent Systems, vol. 18, no. 5, pp. 24-33, 2003.
12. Mabroukeh, N.R. & Ezeife, C.I, "Semantic-Rich Markov Models for Web Prefetching", *IEEE International Conference on Data Mining Workshops*, Miami, Florida, USA, 2009.
13. N. F. Noy, M. Sintek, S. Decker, M. Crubezy, R.W. Fergerson, and M. A. Musen. "Creating semantic web contents with protege-2000", IEEE Intelligent Systems, 16(2):60–71, 2001.
14. N.F. Noy and M.A. Musen, "PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment", In Proceedings of AAAI/IAAI, pp.450-455, 2000.

15. P. Kungas; M. Matskin. "Web Services roadmap: the Semantic Web perspective", Proceedings of the International Conference on Internet and Web Applications and Services, 2006.
16. Q.A. Liang and H. Lam, "Web Service Matching by Ontology Instance Categorization", Proc. IEEE Int'l Conf. on Services Computing (SCC '08), pp. 202-209, 2008.
17. Ruoming J., Yang G., and Agrawal G., "Shared memory parallelization of data mining algorithms: Techniques, programming interface and performance", IEEE Transactions on Knowledge and Data Engineering, vol. 17, no .1, pp. 71-89, 2005.
18. Shanfeng Qi, Xinhuai Tang, and Delai Chen, "An Automated Web Services Composition System Based on Service Classification and AI Planning", Second International Conference on Cloud and Green Computing, Xiangtan, pp. 537 – 540, 2012.
19. T.R. Gruber, "A Translation Approach to Portable Ontologies", knowledge Acquisition, vol. 5, no. 2, pp. 199-220, 1993.
20. Wei, L. & Lei, S , "Integrated Recommender Systems Based on Ontology and Usage Mining", in J. Liu, J. Wu, Y. Yao & T. Nishida (eds), *Active Media Technology*, vol. 5820, Springer-Verlag Berlin Heidelberg, pp. 114–125, 2009.
21. Xu, G., Zhang, Y. & Yi, X, "Modelling User Behaviour for Web Recommendation Using LDA Model", paper presented to the *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, 2008.
22. Y. Zhao, J. Dong, and T. Peng, "Ontology Classification for Semantic-Web-Based Software Engineering", IEEE Trans. Services Computing, vol. 2, no. 4, pp. 303-317, Oct.-Dec. 2009.