A Support Environment for Domain Ontology Development with General Ontologies and Text Corpus

Naoki Sugiura[1], Noriaki Izumi[2], and Takahira Yamaguchi[1]

***Abstract*—For constructing semantically rich service descriptions in Grid services, emerging ontologies are being used. To generate ontologies, an issue named "ontology bottleneck", the lack of efficient ways to build ontologies, has been coming up. Therefore, it is an urgent task to improve the methodology for rapid development of more detailed and specialized domain ontologies. However, it has been a hard task because domain concepts have highly-specialized semantics and the number of concepts is fairly large. In order to reduce the cost, DODDLE II (a domain ontology rapid development environment II) has been developed in our research group. In this paper, we confirm the significance of DODDLE II. In addition, we introduce our plan for further extension for the Semantic Web as a future work.**

***Index Terms*—Ontology Development, Knowledge Engineering, Grid services**

I. Introduction

WHILE Grid services deliver dynamic and relevant applications, a key remaining challenge is supporting automated interoperability without human intervention. Although ontologies are being used in many application areas to improve interoperability, we still face the problem of high cost associated with building up ontologies manually. In particular, since domain ontologies have the meaning specific to application domains, human experts have to make huge efforts for constructing them entirely by hand. In order to reduce the costs, automatic or semi-automatic methods have been proposed using knowledge engineering techniques and natural language processing ones [1]. However, most of these environments facilitate the construction of only a hierarchically-structured set of domain concepts, in other words, taxonomic conceptual relationships. For example, DODDLE [2] developed by us uses a machine-readable dictionary (MRD) to support a user in constructing concept hierarchy only.

In this paper, we extend DODDLE into DODDLE II that constructs both taxonomic and non-taxonomic conceptual relationships, exploiting WordNet [4] and domain specific text corpus with the automatic analysis of lexical co-occurrence statistics based on WordSpace [3] and an association rule algorithm [5]. Furthermore, we evaluate how DODDLE II works in the field of business, xCBL (XML Common Business Library)[6]. The empirical results show us that DODDLE II can support a domain expert in constructing domain ontologies.

[1]Department of Computer Science, Shizuoka University, 3-5-1, Johoku, Hamamatsu, Shizuoka, 432-8011, Japan (phone: +81-53-478-1473; fax: +81-53-473 –6421). {sugiura, yamaguti}@ks.cs.inf.shizuoka.ac.jp

[2]Cyber Assist Research Center, National Institute of Advanced Industrial Science and Technology, 2-41-6, Aomi, Koto-ku Tokyo, Japan. niz@ni.aist.go.jp

II. DODDLE II: A DOMAIN ONTOLOGY RAPID DEVELOPMENT ENVIRONMENT

A. Overview

Fig. 1 describes the system overview of DODDLE II. We can build concept specification templates by putting together taxonomic and non-taxonomic relationships for the input domain terms. The relationships should be identified in the interaction with a human expert.

B. Taxonomic Relationship Acquisition

First of all, TRA module does "spell match" between input domain terms and WordNet. The "spell match" links these terms to WordNet. Thus the initial model from the "spell match" results is a hierarchically structured set of all the nodes on the path from these terms to the root of WordNet. However, the initial model has unnecessary internal terms (nodes) and

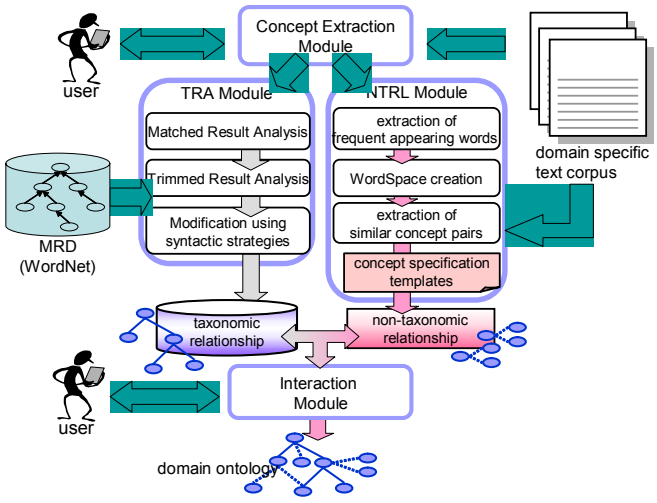

Fig. 1. DODDLE II overview

they do not contribute to keep topological relationships among matched nodes, such as parent-child relationship and sibling relationship. So we get a trimmed model by trimming the unnecessary internal nodes from the initial model (see Fig. 2). After getting the trimmed model, TRA module refines it by interaction with a domain expert, using Matched result analysis (see Fig. 3) and Trimmed result analysis (see Fig. 4). TRA module divides the trimmed model into a PAB (a PAth including only Best spell-matched nodes) and an STM (a Subtree that includes best spell-matched nodes and other nodes and so can be Moved) based on the distribution of best-matched nodes. A PAB is a path that includes only best-matched nodes that have the senses good for given domain specificity.
Because all nodes have already been adjusted to the domain in PABs, PABs can stay in the trimmed model. An STM is such a subtree that an internal node is a root and the subordinates are only best-matched nodes. Because internal nodes have not been confirmed to have the senses good for a given domain, an STM can be moved in the trimmed model.

In order to refine the trimmed model, DODDLE II can use trimmed result analysis. Taking some sibling nodes with the same parent node, there may be big differences about the number of trimmed nodes between them and the parent node. When such a big difference comes up on a subtree in the trimmed model, it is better to change the structure of it. DODDLE II asks a human expert whether the subtree should be reconstructed. Based on the empirical analysis, the subtrees with two or more differences may be reconstructed.
Finally, DODDLE II completes taxonomic relationships of the input domain terms manually from the user.

C. *Non-Taxonomic Relationship Learning*

NTRL module almost comes from WordSpace, which derives lexical co-occurrence information from a large text corpus and is a multi-dimension vector space (a set of vectors). The inner product between two word vectors works as the measure of their semantic relatedness. When two words' inner product is beyond some upper bound, there are possibilities to have some non-taxonomic relationship between them. NTRL module also uses an association rule algorithm to find associations between terms in text corpus. When an association rule between terms exceeds user-defined thresholds, there are possibilities to have some non-taxonomic relationships between them.

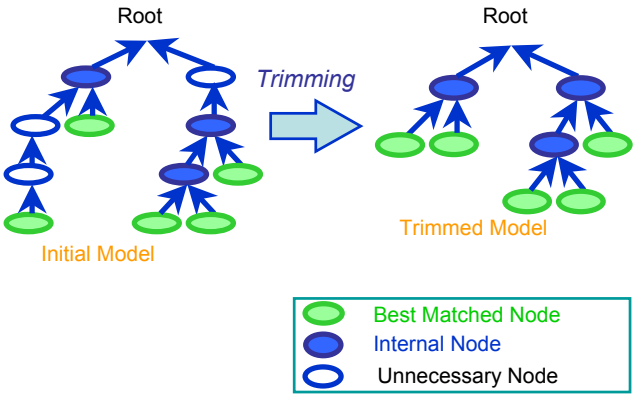

Fig. 2. Trimming Process

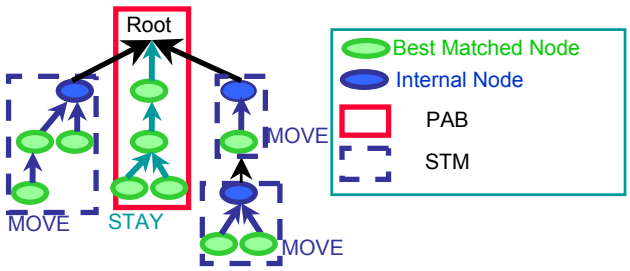

Fig. 3. Matched Result Analysis

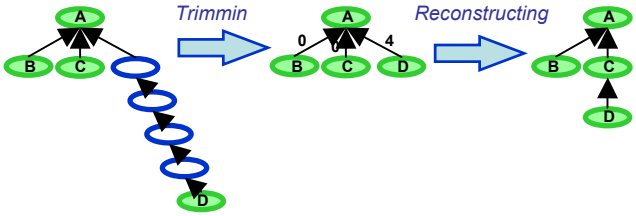

Fig. 4. Trimmed Result Analysis

D. *Construction of WordSpace*

WordSpace is constructed as shown in Fig. 5.
1. *Extraction of high-frequency 4-grams* Since letter-by-letter co-occurrence information becomes too much and so often irrelevant, we take term-by-term co-occurrence information in four words (4-gram) as the primitive to make up co-occurrence matrix useful to represent context of a text based on experimented results. We take high frequency 4-grams in order to make up WordSpace.
2. *Construction of collocation matrix* A collocation matrix is constructed in order to compare the context of two 4-grams. Element $a_{i,j}$ in this matrix is the number of 4-gram f_i which comes up just before 4-gram f_i (called collocation area). The collocation matrix counts how many other 4-grams come up before the target 4-gram. Each column of this matrix is the 4-gram vector of the 4-gram f.
3. *Construction of context vectors* A context vector represents context of a word or phrase in a text. A sum of 4-gram vectors

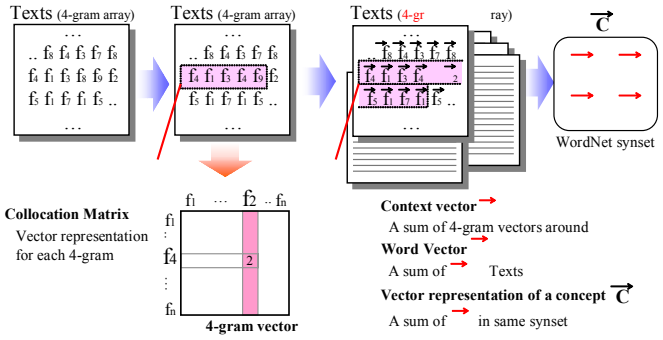

Fig. 5. Construction Flow of WordSpace

around appearance place of a word or phrase (called context area) is a context vector of a word or phrase in the place.

4. *Construction of word vectors* A word vector is a sum of context vectors at all appearance places of a word or phrase within texts, and can be expressed with Eq.1. Here, is a vector representation of a word or phrase *w*, *C(w)* is appearance places of a word or phrase *w* in a text, and *φ(f)* is a 4-gram vector of a 4-gram *f*. A set of vector *τ(w)* is WordSpace.

$$\tau(w) = \sum_{i \in C(w)} \left(\sum_{f\ closetoi} \varphi(f) \right) \quad (1)$$

5. *Construction of vector representations of all concepts* The best matched "synset" of each input terms in WordNet is already specified, and a sum of the word vector contained in these synsets is set to the vector representation of a concept corresponding to an input term. The concept label is the input term.

6. *Construction of a set of similar concept pairs* Vector representations of all concepts are obtained by constructing WordSpace. Similarity between concepts is obtained from inner products in all the combination of these vectors. Then we define certain threshold for this similarity. A concept pair with similarity beyond the threshold is extracted as a similar concept pair.

Finding Association Rules between Input Terms The basic association rule algorithm is provided with a set of transactions, $T := \{t_i \mid i = 1..n\}$, where each transaction t_i consists of a set of items, $t_i = \{a_{i,j} \mid j = 1..m_i, a_{i,j} \in C\}$ and each item $a_{i,j}$ is form a set of concepts C. The algorithm finds association rules $X_k \Rightarrow Y_k : (X_k, Y_k \subset C, X_k \cap Y_k = \{\})$ such that measures for support and confidence exceed user-defined thresholds. Thereby, support of a rule $X_k \Rightarrow Y_k$ is the percentage of transactions that contain $X_k \cup Y_k$ as a subset (Eq.2)and confidence for the rule is defined as the percentage of transactions that Y_k is seen when X_k appears in a transaction (Eq.3).

$$support\,(X_k \Rightarrow Y_k) = \frac{\left|\left\{t_i \middle| X_k \cup Y_k \subseteq t_i\right\}\right|}{n} \quad (2)$$

$$confidence\,(X_k \Rightarrow Y_k) = \frac{\left|\left\{t_i \middle| X_k \cup Y_k \subseteq t_i\right\}\right|}{\left|\left\{t_i \middle| X_k \subseteq t_i\right\}\right|} \quad (3)$$

As we regard input terms as items and sentences in text corpus as transactions, DODDLE II finds associations between terms in text corpus. Based on experimented results, we define the threshold of support as 0.4% and the threshold of confidence as 80%. When an association rule between terms exceeds both thresholds, the pair of terms is extracted as candidates for non-taxonomic relationships.

E. Constructing and Modifying Concept Specification Templates

A set of similar concept pairs from WordSpace and term pairs from the association rule algorithm becomes concept specification templates. Both of the concept pairs, whose meaning is similar (with taxonomic relation), and has something relevant to each other (with non-taxonomic relation), are extracted as concept pairs with above-mentioned methods. However, by using taxonomic information from TRA module with co-occurrence information, DODDLE II distinguishes the concept pairs which are hierarchically close to each other from the other pairs as TAXONOMY. A user constructs a domain ontology by considering the relation with each concept pair in the concept specification templates, and deleting unnecessary concept pairs.

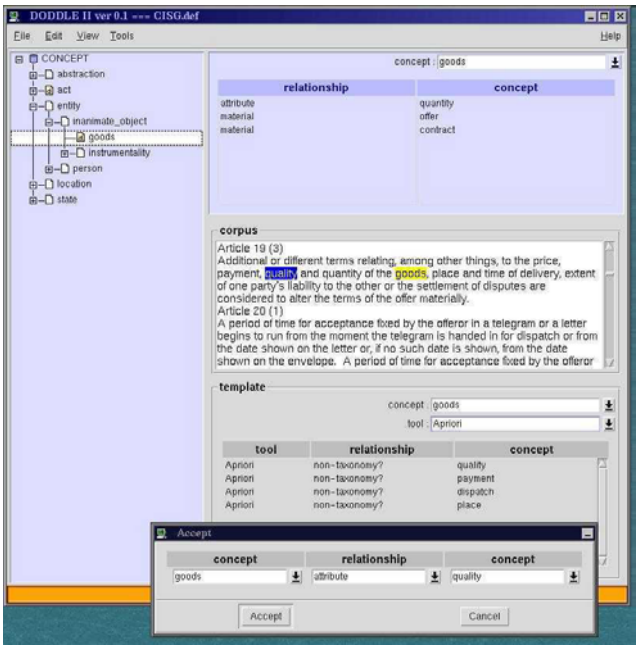

Fig. 6. The Ontology Editor

III. CASE STUDY

In order to evaluate how DODDLE II is going in a practical field, a case study has been done in particular field of business called xCBL (XML Common Business Library) [6]. DODDLE II has been implemented on Perl/Tk. Fig. shows the typical screen of DODDLE II.

A. Input terms

Table 1 shows input terms in this case study. They are 57 business terms extracted by a user from xCBL Document Reference. The user is not an expert but has business knowledge.

B. Taxonomic Relationship Acquisition

Table 2 shows the number of concept pairs in each model under taxonomic relationship acquisition and

Table 3 shows the evaluation of two strategies by the user. The recall per subtree is more than 0.5 and is good. The precision and the recall per path are less than 0.3 and are not so good, but about 80 % portion of taxonomic relationships were constructed with TRA module support. We evaluated TRA module worked well in this case study.

TABLE 1
SIGNIFICANT 57 CONCEPTS IN XCBL

acceptance	agreement	auction	availability	business
buyer	change	contract	customer	data
date	delivery	document	Exchange rate	financial institution
foreign exchange	goods	information	invoice	item
Line item	location	marketplace	message	money
order	organization	partner	Party	payee
payer	payment	period of time	Price	process
product	purchase	Purchase agreement	Purchase order	quantity
quotation	quote	receipt	rejection	request
resource	response	schedule	seller	service
shipper	status	supplier	system	third party
transaction	user			

TABLE 2
THE CHANGE OF THE NUMBER OF CONCEPTS UNDER TAXONOMIC RELATIONSHIP ACQUISITION

Model	Input Terms	Initial Model	Trimmed Model	Concept Hierarchy
# Concept	57	152	83	82

TABLE 3
PRECISION AND RECALL IN THE CASE STUDY WITH XCBL

	Precision	Recall per Path	Recall per Subtree
Matched Result	0.2(5/25)	0.29(5/17)	0.71(5/7)
Trimmed Result	0.22(2/9)	0.13(2/15)	0.5(2/4)

C. Non-Taxonomic Relationship Learning

1) Construction of WordSpace

High-frequency 4-grams were extracted from xCBL Document Description (about 2,500 words), and 1240 kinds of 4-grams were obtained. In order to keep density of a collocation matrix high, the extraction frequency of 4-grams must be adjusted according to the scale of text corpus. As xCBL text is relatively short, the extraction frequency was set as 2 times this case. In order to construct a context vector, a sum of 4-gram vectors around appearance place circumference of each of 57 concepts was calculated. In order to construct a context scope from some 4-grams, it consists of putting together 10 4-grams before the 4-gram and 10 4-grams after the 4-grams independently of length of a sentence. For each of 57 concepts, the sum of context vectors in all the appearance places of the concept in xCBL was calculated, and the vector representations of the concepts were obtained. The set of these vectors is used as WordSpace to extract concept pairs with context similarity. Having calculated the similarity from the inner product for concept pairs which is all the combination of 57 concepts, 40 concept pairs were extracted.

2) Finding Associations between Input Terms

DODDLE II extracted 39 pairs of terms from text corpus using the above-mentioned association rule algorithm. There are 13 pairs out of them in a set of similar concept pairs extracted using WordSpace. Then, DODDLE II constructed concept specification templates from two sets of concept pairs extracted by WordSpace and Associated Rule algorithm. However, the user didn't have enough time to modify them and didn't finish modifying them.

3) Evaluation of Results of NTRL module

The user evaluated the following two sets of concept pairs: one is extracted by WordSpace (WS) and the other is extracted by Association Rule algorithm (AR). Fig. 5 shows two different sets of concept pairs from WS and AR. It also shows portion of extracted concept pairs that were accepted by the user. Table 4 shows the details of evaluation by the user, computing precision only. Because the user didn't define concept definition in advance, we can not compute recall. Looking at the field of precision in Table 4, the precision from WS is higher than others. Most of concept pairs which have relationships were extracted by WS. The percentage is about 77% (30/39). But there are some concept pairs which were not extracted by WS. Therefore taking the join of WS and AR is the best method to support a user to construct non-taxonomic relationships.

TABLE 4
EVALUATION BY THE USER WITH XCBL DEFINITION

	WordSpace (WS)	Association Rules (AR)	The Join of WS and AR
# Extracted concept pairs	40	39	66
# Accepted concept pairs	30	20	39
# Rejected concept pairs	10	19	27
Precision	0.75(30/40)	0.51(20/30)	0.59(39/66)

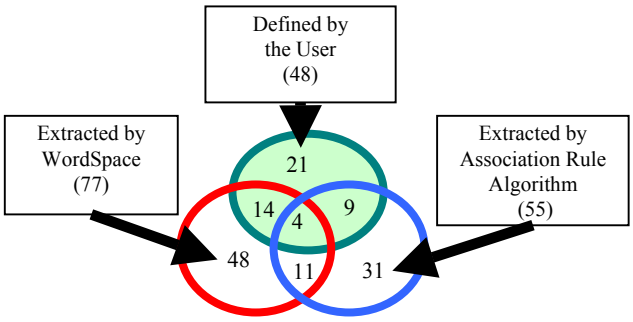

Fig. 5. Two Different Sets of Concept Pairs from WS and AR and Concept Sets have Relationships

D. *Results and Evaluation of the Case Study*

In regards to support in constructing taxonomic relationships, the precision and recall are less than 0.3 in the case study. Generally, 70 % or more support comes from TRA module. About more than half portion of the final domain ontology results in the information extracted from WordNet. Because the two strategies just imply the part where concept drift may come up, the part generated by them has about 30 % hit rate. So one out of three indications based on the two strategies work well in order to manage concept drift. Since the two strategies use matched and trimmed results, based on structural information of an MRD only, the hit rates are not so bad. In order to manage concept drift smartly, we may need to use more semantic information that is not easy to come up in advance in the strategies, and we also may need to use domain specific text corpus and other information resource to improve supporting a user in constructing taxonomic relationships.

In regards to construction of non-taxonomic relationships, the precision in the case study with xCBL is good. Generating non-taxonomic relationships of concepts is harder than modifying and deleting them. Therefore, DODDLE II supports the user in constructing non-taxonomic relationships.

After analyzing results of the case study, we have the following problems:

- Determination of a Threshold: Threshold of the context similarity changes in effective value with domain. It is hard to set up the most effective value in advance.
- Specification of a Concept Relation: Concept specification templates have only concept pairs based on the context similarity, it still requires high cost to specify relationships between them. It is needed to support specification of concept relationships on this system in the future work.
- Ambiguity of Multiple Terminologies: For example, the term “transmission” is used in two meanings, “transmission (of goods)” and “transmission (of communication)”, in the xCBL document. However, DODDLE II considers these terms as the same and creates WordSpace as it is. Therefore constructed vector expression may not be exact. In order to extract more useful concept pairs, semantic specialization of a multi-sense word is necessary, and it should be considered that the 4-grams with same appearance and different meaning are different 4-grams.

IV. Conclusions and Future Work

In this paper, we discussed how to construct domain ontologies using an existing MRD and text corpus. In order to acquire taxonomic relationships, two strategies have been proposed: matched result analysis and trimmed result analysis. Furthermore, to learn non-taxonomic relationships, concept pairs may be related to concept definition, extracted on the basis of the co-occurrence information in text corpus, and a domain ontology is developed by the modification and specification of concept relations with concept specification templates. It serves as the guideline for narrowing down huge space of concept pairs to construct domain ontologies.

It is almost craft-work to construct domain ontologies, and still difficult to obtain the high support rate on the system. DODDLE II mainly supports for construction of a concept hierarchy with taxonomic relationships and extraction of concept pairs with non-taxonomic relationships. However, a support for specification concept relationship is indispensable.

As a future work, we are trying to find out the way to extend DODDLE II into DODDLE-R (DODDLE RDF model extension). In the recent stream of ontology engineering towards the Semantic Web, the relation between meta-models of Web resources represented in RDF (Resource Description Framework) [7] and RDFS (RDF Vocabulary Description Language) [8] (as a kind of ontology for particular Web resources) are gathering more attention than before.

Fig. 8 shows the general procedure of DODDLE-R. In

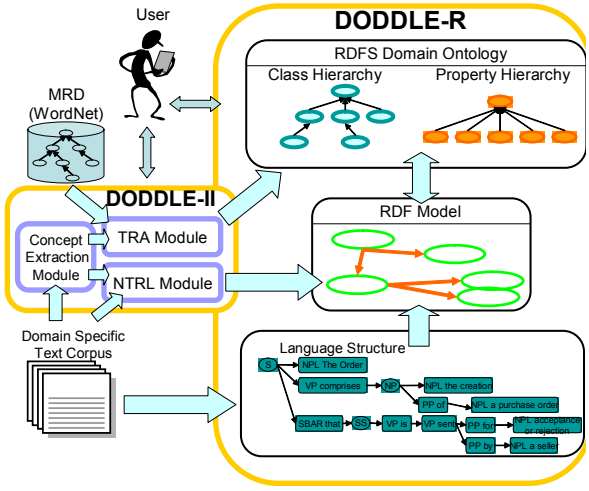

Fig. 6. General Procedure of DODDLE-R

addition to DODDLE-II, DODDLE-R generates natural language structures from text corpus. Then, based on the structures and non-taxonomic relationships produced by NTRL, the prototype of RDF model is built up. Also taxonomic relationships are constructed by using TRA and they become the basis of RDFS class hierarchy. After that, to build up and improve the RDF model and RDFS class hierarchy based on the prototypes as mentioned above, it is necessary to manage their relation. To do that, and also to improve the interaction process with users, the combination with MR3 [9], a state-of-the-art

RDF(S) management tool, must be essential. Furthermore, the strategy to manage the semantic and granularity gaps between RDFS class hierarchy, RDF model and natural language structures would be the key issue of this research work.

ACKNOWLEDGEMENT

This work was supported by Masaki Kurematsu (Iwate Prefectual University, Japan), Naomi Nakaya and Takamasa Iwade (former students of Shizuoka University, Japan).

REFERENCES

[1] Y. Ding and S.Foo, "Ontology Research and Development, Part 1 – A Review of Ontology", Journal of Information Science, Vol.28, No2, 123 – 136 (2002)

[2] Rieko Sekiuchi, Chizuru Aoki, Masaki Kurematsu and Takahira Yamaguchi, "DODDLE: A Domain Ontology Rapid Development Environment", PRICAI98, 1998

[3] Marti A. Hearst, Hirsch Schutze, "Customizing a Lexicon to Better Suit a Computational Task", in Corpus Processing for Lexical Acquisition edited by Branimir Boguraev & James Pustejovsky, 77–96

[4] C.Fellbaum ed, "WordNet", The MIT Press, 1998. See also URL: http://www.cogsci.princeton.edu/~wn/

[5] Rakesh Agrawal, Ramakrishnan Srikant, "Fast algorithms for mining association rules," Proc. of VLDB Conference, 487–499 (1994)

[6] xCBL.org, http://www.xcbl.org/xcbl40/documentation/listofdocuments.html

[7] Resource Description Framework (RDF) , http://www.w3.org/RDF/

[8] RDF Vocabulary Description Language 1.0 RDF Schema, http://www.w3.org/TR/rdf-schema/

[9] Noriaki Izumi, Takeshi Morita, Naoki Fukuta and Takahira Yamaguchi, "RDF-based Meta-Model Management Environment", Sanken (ISIR) International Symposium, 2003