Interaction between Lexical Base and Ontology with Formal Concept Analysis

Sujian Li*, Qin Lu, Wenjie Li**, Ruifeng Xu****

Institute of Computational Linguistics, Peking University, Beijing, China

Dept. of Computing, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong

E-mail: lisujian@pku.edu.cn, csluqin@comp.polyu.edu.hk, cswjli@comp.polyu.edu.hk, csrfxu@comp.polyu.edu.hk

Abstract

An ontology describes conceptual knowledge in a specific domain. A lexical base collects a repository of words and gives independent definition of concepts. In this paper, we propose to use FCA as a tool to help constructing an ontology through an existing lexical base. We mainly address two issues. The first issue is how to select attributes to visualize the relations between lexical terms. The second issue is how to revise lexical definitions through analysing the relations in the ontology. Thus the focus is on the effect of interaction between a lexical base and an ontology for the purpose of good ontology construction. Finally, experiments have been conducted to verify our ideas.

1. Introduction

According to Gruber [1995]'s definition, an ontology can be seen as a catalog of the types of things that are assumed to exist in a domain of interest from the perspective of a person who uses a language for the purpose of talking about the domain. How to construct an ontology is a non-trivial task. An ontology is usually constructed either manually or semi-automatically. It is ideal if we can automatically construct an ontology according to the available data resources compiled by experts. A lexical base which contains lexical terms and explanatory notes can serve as such a resource. The lexical base collects a repository of words and can be seen as a machine-readable dictionary, which provides an index into human knowledge. However, a dictionary often gives independent definition of concepts without explicitly indicating their relationships with other concepts. Priss[2004] has pointed out that the construction of detailed lexica requires precise ontological information, and vice versa.

Formal Concept Analysis (FCA) is a mathematical approach to data analysis based on the lattice theory which can provide a natural representation of hierarchies and classifications [Stumme 2002]. FCA can be used to model a lexical knowledge base and construct it as an ontology with clear structures. FCA has been chosen especially for the automatic construction of ontologies [Cimiano 2004]. Through the use of FCA, a lexical base can also be revised through analyzing the relations in the ontology. The improved lexical base can in turn further help the construction of ontology. In this paper, we will present the interaction between a lexical base and a domain ontology using FCA technique. The key to this project is that with the proper tool, not only lexical base can help domain experts to build an ontology, the tool with its visualization capability can also help to revise and fine tune the lexical base.

The rest of this paper is organized as follows. Section 2 introduces the basic concepts. Section 3 describes the framework for the interaction between ontology construction and the improvement of a related lexical base using FCA. Section 4 describes how attributes are selected in FCA construction, and section 5 discusses how relationships among concepts are built. Section 6 explains the experiments and presents the evaluation results. Section 7 gives concluding remarks and future directions.

2. Concepts

2.1. Ontology Definition

In this work, we define ***ontology*** as follows.

Definition 1: An ***ontology***, denoted by ***O***, is defined by a quadruplet, *O = (L, D, C, R),* where *L* is a specific language, *D* is a specific domain, *C* is the set of concepts and *R* is the set of relations between concepts.

Normally, in ontology construction, both *L* and *D* are already known because any construction method would be applied to a specific language, *L*, in a specific domain, *D*. Ontology construction methods aim at how to obtain the set of concept *C* and how to build the set of relationships *R* among concepts. The ontology in this paper refers to a formal ontology as described in [Sowa 2000], which is specified by a collection of names for formal concepts and relation types organized in *partial ordering*.

2.2. FCA Overview

FCA takes two sets of data, one is called the *object set* and the other is called the *attribute set*[1], as well as a binary relationship between the data of the two sets, to form a *formal context*, according to which a so-called *formal concept lattice* is further constructed with a concept inclusion ordering. The definitions of *formal context* and *formal concept* in FCA are given in [Ganter 1999].

In FCA, a formal concept, as a node, has more attributes describing it than its super class concept. The more attributes a formal concept has, the fewer objects it owns. Thus, a formal concept in the lattice having more

[1] Here, the terms *object* and *attribute* are used as the short forms of formal object and formal attribute, respectively.

attributes and less objects than a super class concept. The whole formal concept lattice satisfies the ***partial ordering relationship***.

It is easy to see that the FCA model can be used to represent an ontology where the formal concepts in FCA correspond to the concept set *C* for a specific language *L* and a specific domain *D*. However, the issue is what should be the set of attributes used to describe these concepts and we also need to know how to acquire the mapping between the partial ordering relationship in FCA and *R*.

2.3. Lexical Base

A lexical base provides a set of terms with detailed explanation, which is formalized as follows:

$$LB = \{ < t_i, e_i > \mid 1 \le i \le n\},$$

where t_i is a term and e_i is the corresponding verbal explanation or definition which describes the characteristics of t_i. To construct the formal context of FCA, all the t_i can serve as formal objects, and attributes can be extracted from e_i.

For the Chinese language, we adopt the HowNet [Dong 2000] Lexicon as the seed to construct our lexical base in which terms are described formally by several sememes (the smallest semantic unit), which can serve as attributes in FCA. For example, "解码 (decode)" is a term[2] defined by a set of sememes "computer|电脑", "translate|翻译" and "software|软件".

3. Design Framework

Figure 1 shows the interaction framework of ontology construction and the lexical base HowNet lexicon. We take HowNet as the starting point to construct an ontology step by step. First, we use information gain as the means to choose the sememes with discriminative power, as the set of attributes. We then construct a concept context using the FCA model to obtain a draft ontology. Through the analysis of the relations obtained from this draft ontology, we can analyze HowNet in terms of the appropriateness and the granularity of the sememes used.

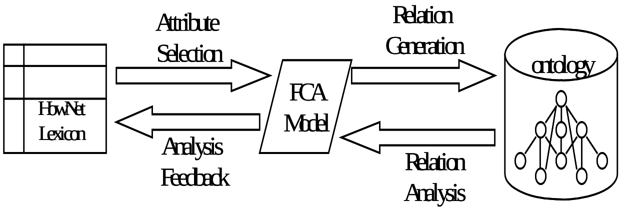


Figure 1: System Framework

The Java API of an open source software called ConExp[3] is used to generate the concept oriented views with FCA technique. ConExp is a visualization and operating tool of FCA[Serhiy 2000]. It provides convenient function to edit attributes and objects, which is the basis for automatic display of a concept lattice. It also provides attribute reduction function to remove the reducible attributes which do not change the structure of the concept lattice. With this tool, we can ignore the concrete algorithm of FCA and focus on the selection of attributes and objects to construct an ontology and improvement of the lexical base.

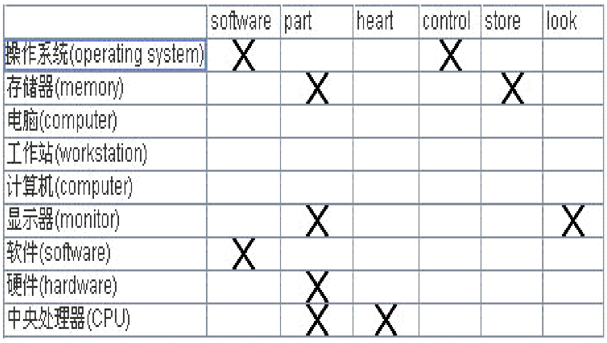
4. Attribute Selection & Ontology Construction

For experimental purpose, we choose Information Technology as our domain for ontology construction with a given manually selected set of IT terms as objects in FCA. Then it is our task to explore the selection of attributes for ontology construction.

HowNet has built a set of descriptive sememes, which are used to define terms and thus can serve as possible attributes from which we must pick the appropriate ones. These selected sememes should have discriminative power to differentiate concepts represented by these terms. The discriminating power is calculated through Information Gain (IG) for each sememe s_i according to the following formula:

$$\begin{aligned} &IG(s_i) = E(S\text{-}\{s_i\})\text{-}E(S) \\ &E(S) = -\Sigma(P_j \log P_j) \end{aligned} \qquad (1)$$

where *E(S)* is the entropy of all terms according to the sememe set *S* of HowNet. $E(S\text{-}\{s_i\})$ is the entropy after s_i is deleted from *S*. P_j is the probability of each class of terms which are split by the sememe set *S* using FCA. All the sememes in HowNet whose information gain is less than a threshold *T*, is filled out. The remaining sememes forms the set of attributes. The threshold *T* is a parameter determined experimentally.



	software	part	heart	control	store	look
操作系统(operating system)	X			X		
存储器(memory)		X			X	
电脑(computer)						
工作站(workstation)						
计算机(computer)						
显示器(monitor)		X				X
软件(software)	X					
硬件(hardware)		X				
中央处理器(CPU)		X	X			

Figure 2a. An example of concept context

Then the relationship between a term t_i and a sememe s_j is represented by a binary membership value. If t_i is defined by s_j, the membership value $\mu(t_i,s_j)$ is 1, 0 otherwise. Here is an example of the formal context for 9 selected IT terms as listed in **Figure 2a**. Then the corresponding

[2] The hypothesis in HowNet is that a term only has one meaning in a specific domain. Hereafter, a term is used to represent a concept which is an object in FCA. A concept is different from a formal concept as a formal concept is only a node in FCA which can contain several terms and their associated attributes. In other words, a formal concept in FCA can represents several terms(and thus concepts) having the equivalent relationship..

[3] http://sourceforge.net/projects/conexp

concept lattice is produced, as shown in **Figure 2b**. A term with fewer descriptive sememes always serves as a super class concept. Thus, the concept lattice visually displays the subsumption relationships between terms. The relations between different terms are represented in the form of a triple <R_k, t_i, t_j>, where t_i, t_j are two terms and R_k indicates the relation between t_i and t_j. Here we only define two kinds of relations: *equivalent* or *isa*. In figure 2, we get the results such as <电脑(computer), 计算机(computer), *equivalent* >, <硬件(hardware), 电脑 (computer), *isa* > and so on.

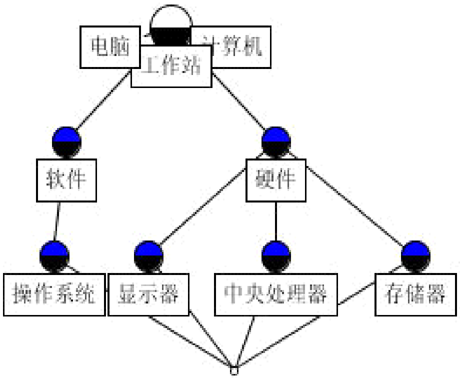


Figure 2b. The Corresponding Concept Lattice of the 9 IT terms in **Figure 2a**

5. Relation Analysis & Lexicon Improvement

The relations reflect the understanding of concepts by domain experts. In compiling a lexicon manually, it is important to provide domain experts with the appropriate tools and timely feedback for fine-tuning the definitions of terms through a visualized interface. In this section, we manually analyze the relations, mainly the inappropriate ones, and project them to the context lattice so that adjustments to terms and their definitions can be easily done. The following analysis and lexical tuning have been conducted.

1) Merging of equivalent terms: Nodes in the concept lattice should represent different concepts. For example, “computer” and “workstation” have the same definition in the lexicon, which means that they are equivalent, and thus, they should be merged as one node. However, if they need to be conceptually differentiated, revision to the definitions or attributes should be done.
2) Modification of inappropriate links: For *isa* relations, we also mainly analyze the inappropriate ones. Two relevant terms of one relation are mapped to one node and its superclass node respectively. For example, term “database” is defined by sememes “software, control, store” and forms an *isa* relation with term “operation system” defined by sememes “software, control”. The *isa* relation is inappropriate and can be modified by defining terms more accurately, e.g. “operation system” is redefined by “software, control, manage”.
3) Filling of terms: For nodes in the lattice which do not have associated terms, it implies that there is a concept, yet association is appropriate and terms are missing. Thus, according to the information that the sememes bear, domain experts can fill in some concrete term if possible, the combination of the sememes as its definition.

For the first two analyses above, with the help of the lattice, it is relatively easy to modify the ineligible relations, comprehensively considering its relations with other terms at the same time. The main operation is to revise the terms with appropriate sememes, including changing, adding or deleting some descriptive sememes. The third analysis means adding new concepts according to existing concepts and its operation includes discovering new terms and defining them. The whole analysis process aims at tuning the sememe set, redefining terms with new sememes, discovering new terms, according to which the lexical base can be improved. Then we can reuse FCA technique to generate relations between terms. The whole process is depicted as in **Figure 1.**

6. Experiments

6.1. Materials

To verify our idea, we have conducted a small-scale experiment due to the manpower limit. Firstly, we used HowNet version 2000, from which 57 IT domain terms with their definitions are chosen as seeds. By calculating information gain using formula (1) with threshold value equal to zero, 38 sememes are selected as formal attributes. Secondly, from a Chinese thesaurus [Wang 1993], we picked out about another 40 IT terms, and define them with the format of HowNet Lexicon according to their liberal explanations. HowNet lexicon is a general resource and there are only a small number of sememes specific for IT domain. In order to define and differentiate concepts in IT domain, domain experts have added twenty five new sememes so that the total number of sememes 63 for the 97 IT terms used in this experiment. Some of the attributes can be written as a combination of other attributes, and has no influence on the structure of the concept lattice, referred to as reducible attributes[Ganter 1999]. The attributes will be eliminated before producing the concept lattice. For example, in these 97 IT terms, whatever term has the sememe “Place” it always has “ProperName”. Thus, “ProperName” can be seen having no discriminative power and is thus removed. It is noted that the attributes are reduced in producing a concept lattice for a specific set of terms although they may have discriminative power for a larger set of terms.

After attribute reduction, there are 52 sememes left. **Figure 3a** shows the hierarchical relations of these 97 IT terms using ConExp before lexicon improvement, **Figure 3b** shows the hierarchical relations of these IT terms after lexicon improvement. In these two figures, the circles represent formal concepts. The larger a circle is, the more terms it owns. The texts in the rectangles are the terms and the rectangles with shadow are attributes. A node with only a point means that the formal concept does not own any terms but only owns some attributes. Because there are too many nodes and edges in these two figures, it is difficult to see much difference by a first glance. In fact, Figure 3b has fewer terms included in each node, and thus Figure 3b can distinguish better between terms. The following evaluation result for produced relations will prove this.

From Figure 3a, we found 109 *equivalent* relations and 147 *isa* relations. During lexicon improvement, we have analyzed the results and conducted operations on sememes and terms. We have adjusted the sememe set, revised the inappropriate definitions, and filled some new terms, just as the operations mentioned in Section 5. For example, to differentiate between terms “计算机 computer” and “服务器 server”, a new sememe “network” is adopted. Note that the new terms are mainly added to the nodes having only attributes. For example, in Figure 3a the node circled by the dashed line has only an attribute “action” with no terms. Through human analysis, a new term “操作(operation)” is added. Then in Figure 3b, a term “操作(operation)” with attribute “action” is represented by the node also circled by the dashed line. Now there are 105 terms and 53 descriptive sememes, from which we can acquire 71 *equivalent* relations and 154 *isa* relations.

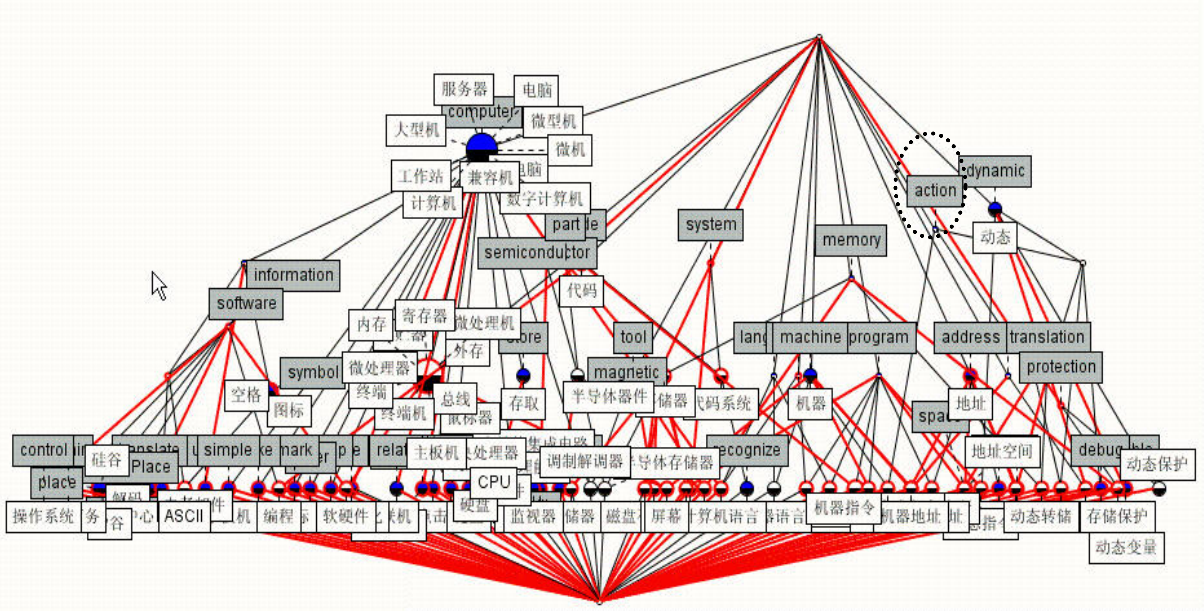


Figure 3a. Hierarchical relations produced before Lexicon Improvement

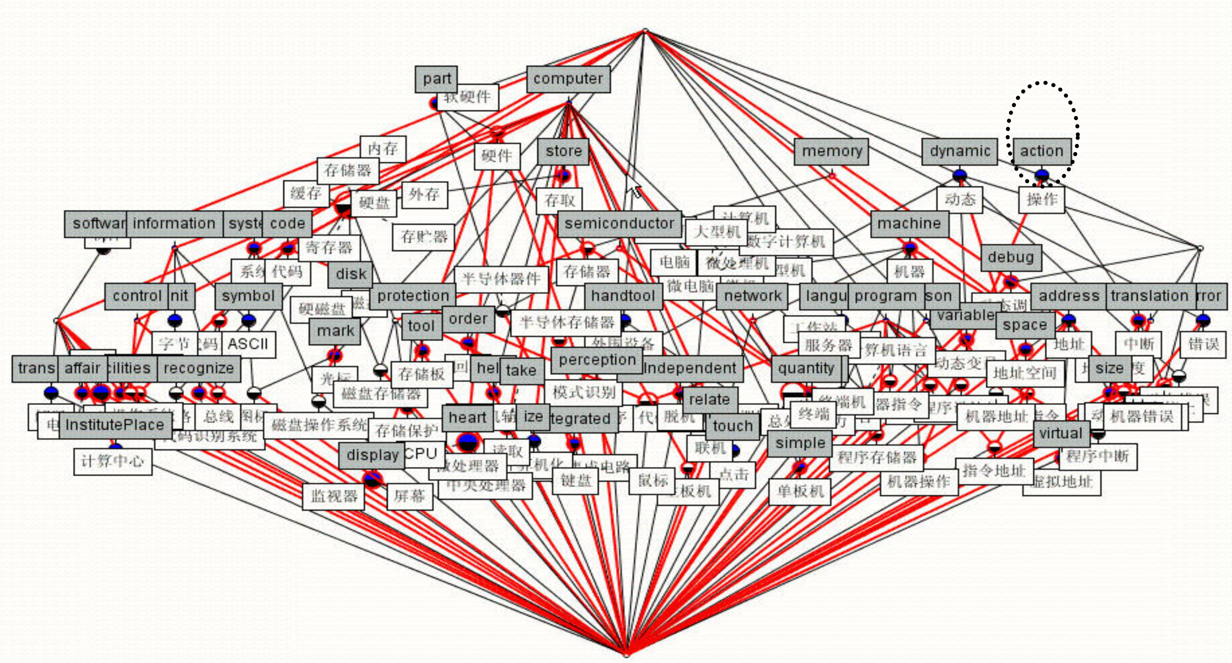


Figure 3b Hierarchical relationships produced after lexicon improvement

6.2. Evaluation

Most researchers evaluate their ontologies in two parts: the lexical part and the taxonomic part [Cimiano 2003, Jiang 2003]. Here our premise is that the domain-specific terms have been correctly recognized. Then our evaluation work mainly focuses on measuring the automatically generated taxonomic relationship between terms. According to the link in each concept lattice, we pick out the equivalent and hierarchical relations in the form of triples $< t_i, t_j, R(t_i, t_j) >$. Five evaluators evaluated the eligibility of these triples. The evaluators all come from the IT fields and are familiar with those terms. For each relation represented by a triple, the evaluators answer [YES] or [NO]. [YES] means that the triple represents an eligible relation and [NO] means it is an unqualified one. With the ratio of answer [YES] we can evaluate which attribute set is more appropriate and whether the lexicon has been improved.

	Equivalent	*Isa*	Average
Before improvement	31.4%	53.1%	43.3%
After improvement	84.4%	59.2%	76.4%

Table 1: Evaluation of Answer "Yes" Ratio

In Table 1, the evaluation results show that experiment before lexicon improvement gets about 43.2% of answers as "YES" on average. After revising the lexical base, the definitions become more reasonable. For the improved lexical base, the result is that we can get 84.4% of answer as "YES" for the *equivalent* relations and 59.2% of answer as "YES" for the *isa* relations respectively. And the average ratio of answer "YES" is 76.4%.

6.3. Discussion

It is a win-win situation to automatically identify the relations between domain-specific concepts by using existing resources and provide the visualized tool for improving the lexical base. Our method has received such effect. However, there are still some problems in the method as stated below.

In principle, each attribute in the attribute set of an ontology should be independent. However, in reality, the selection of independent attributes is very difficult. For example, the sememes "computer" and "software", "action" and "break" are related attributes. We try to avoid selecting the dependent attributes to describe the objects. Sometimes dependent attributes cannot be avoid such as in the case that one attributes is the subtype of another attribute(called its parent attribute). Without the subtype attribute, two different concepts cannot be distinguished. In this case, when a concept is associated with the subtype attribute, it should be made explicit in the concept lattice that the term also owns the parent attribute. This will help to maintain the explicit subsumption relationship for consistency.

From the evaluation, we can see that more correct relations have been identified after lexicon improvement than before improvement. At the same time, after lexicon improvement, the definitions in the lexical base are more detailed and precise. For example, before lexicon improvement, "server (服务器)" has the same definition as "computer(计算机)" which in turn affect their relations. Consequently, they are identified wrongly as having the *equivalent* relation. After lexicon improvement, "server" is redefined and is represented as a kind of "computer(计算机)", and thus the equivalent relationship no longer hold. There are 71 formal concepts before lexicon improvement and 80 formal concepts after the improvement, respectively. That is, redundant concepts are removed and the terms are better differentiated. At the same time, more attributes are adopted. With more attributes, it is easier to define each term separately. However, some sparse attributes may be introduced as a result. We can try to confine the work to use only the predefined sememe set and avoid adding new attributes. Sometimes, after lexicon improvement with existing sememes, there are still some inappropriate definitions, which cause the inappropriate relations in the ontology. Thus to make concepts distinguishable, we have to expand our sememe set. For example, the sememe "network" is added in this work.

In addition, FCA has provided a visualized interface, which makes it convenient to improve the lexical base. However, the relationship between concepts still needs a lot of manual work to verify and adjust the term definitions. For the inappropriate relations, experts redefine each term according to their own knowledge. Such definition is somewhat subjective, which may be inconsistent with knowledge of another expert. That is why in our evaluation, that the average evaluation is only 76.4% correct.

7. Conclusions

In this paper, we have proposed how to use formal concept analysis as an interactive for the construction of an ontology using a lexical knowledge base. Firstly, the focus is how to select attributes to visualize the relations between lexical items. Then, lexical definitions can be revised through analyzing the relations in the ontology.

However, this analysis is still quite labor intensive. We can only select a small number of terms for experiments. In future work, the sememe set needs to be further tuned to increase the scale of our experiment. Methods of automatic or semi-automatic inspection and evaluation of lexical base will also be investigated.

8. Acknowledgements

We give our thanks to the anonymous reviewers for the helpful comments. The work presented in this paper is supported by Research Grants Council of Hong Kong (reference number: CERG PolyU 5190/04E and 5181/03E, Z08K) and the Beijing Natural Science Foundation (Grant no: 4052019).

9. References

Cimiano, P. & Staab, S. & Tane, J., 2003, Automatic Acquisition of Taxonomies from Text: FCA Meets NLP. In Proceedings of the International Workshop on

Adaptive Text Extraction and Mining.

Cimiano, P. & Hotho, A. & Stumme, G. & Tane. J., 2004, Conceptual Knowledge Processing with Formal Concept Analysis and Ontologies. In Proceedings of the 2nd International Conference on Formal Concept Analysis.

Dong, Z. Dong, Q. *HowNet*, http://www.keenage.com

Ganter, B., & Wille, R., 1999, Formal Concept Analysis. Mathematical Foundations. Berlin-Heidelberg-New York: Springer, Berlin-Heidelberg.

Gruber, T.R. 1995, Toward Principles for the Design of Ontologies Used for Knowledge Sharing. International Journal of Human and Computer Studies, 43(5/6):907-928.

Jiang, G. & Ogasawara, K. & et al., 2003, Context-based Ontology Building Support for Clinical Domains Using Formal Concept Analysis. Int J Med Inform. 71(1):71-81.

Quan, T.T. & Hui, S.C. & Cao, T.H., 2004, FOGA: A Fuzzy Ontology Generation Framework for Scholarly Semantic Web, Knowledge Discovery and Ontologies (KDO-2004), Workshop at ECML/PKDD 2004.

Priss, U., 2004, Linguistic Applications of Formal Concept Analysis. In G. Stumme and R. Wille, editors, Formal Concept Analysis - State of the Art. Springer.

Serhiy, A. Y., 2000, System of data analysis "Concept Explorer". (In Russian). Proceedings of the 7th national conference on Arti_cial Intelligence KII-2000, p. 127-134, Russia.

Sowa, J.F., 2000, Knowledge Representation, Logical, Philosophical, and Computational Foundations, Brooks/Cole Thomson Learning.

Stumme, G., 2002, Formal Concept Analysis on its Way from Mathematics to Computer Science. In: U. Priss, D. Corbett, G. Angelova (Eds.): Conceptual Structures: Integration and Interfaces, Proc. ICCS 2002, LNAI 2393, Springer, Heidelberg 2002, 2-19

Wang, G.B., Encyclopedia of Computer Science and Engineering: Classification, Chinese-English, English-Chinese, Jinan: Shangdong Education Publisher, 1993.