SCIENTIFIC PAPERS, UNIVERSITY OF LATVIA, 2009. Vol. 751
COMPUTER SCIENCE AND INFORMATION TECHNOLOGIES 40–51 P.

Automatic Verification of the Conceptual Model and Its Documentation

Algirdas Laukaitis[1] and **Olegas Vasilecas**[1, 2]
[1] Vilnius Gediminas Technical University, Sauletekio al. 11, LT-10223 Vilnius-40, Lithuania
algirdas.laukaitis@fm.vgtu.lt, olegas@fm.vgtu.lt
[2] Klaipeda University, Herkaus Manto 84, LT-92294 Klaipeda, *olegas.vasilecas@ik.ku.lt*

By using background knowledge of general and specific domains, and by processing a new natural language corpus, experts are able to produce a conceptual model for some specific domain. In this paper, we present a model that tries to capture some aspects of this conceptual modeling process. This model is functionally organized into two information processing streams: one reflects the process of formal concept lattice generation from the domain conceptual model; the other reflects the process of formal concept lattice generation from the domain documentation. It is expected that similarity between those concept lattices reflects similarity between documentation and the conceptual model. In addition to this process of formal verification, a set of natural language processing artifacts are created. Those artifacts then can be used for the development of information system natural language interfaces. To demonstrate it, an experiment of concept identification from natural language queries is provided at the end of this paper.

Keywords: information systems engineering, formal concept analysis, IS document self-organization, natural language processing.

1 Introduction

Software engineers spend hours in defining information systems (IS) requirements and finding common ground of understanding. The overwhelming majority of IS requirements are written in a natural language supplemented with a conceptual model and other semi-formal UML diagrams. In the form of semantic indexes, the bridge between documents and the conceptual model can be useful for more effective communication and model management. Therefore, integration of the natural language processing (NLP) into information system documentation process is an important factor for meeting the challenges of modern software engineering methods. Reusing natural language IS requirement specifications and compiling them into formal statements has been a prolonged challenge [2], [17]. Kevin Ryan claimed that NLP is not mature enough to be used in requirement engineering [16] and our research justifies that as well. Nevertheless, we hope that the current paper will suggest some promising findings towards this challenging task.

In this paper, by combining the symbolic and connectionist paradigms, we present our efforts to overcome difficulties and problems of the natural language usage in all stages of IS development. The self-organizing map (SOM) [13] is proposed as a tool to analyze the documents and communication utterance, and Formal Concept Analysis (FCA) [5] is suggested as a method to reinterpret SOM topology and to verify the comprehensibility and soundness of the information system documentation and model. All presented ideas and methodological inference have been tested with the IBM Information FrameWork (IFW) [9], which is a comprehensive set of banking-specific business models from the IBM corporation. For our research, we have chosen the set of models under the name *Banking Data Warehouse.* We define the following problems: 1) *How can we formally verify the IS documentation if we have at least several sentence description for each business information system component?* 2) *What is the architectural solution of the system where the designers, modelers, requirement engineers can verify new pieces of textual documentation and automatically generate hierarchical prototypes of the information system model?* 3) *What components from the new modeling system can be taken and reused as plugins in the natural language interfaces (i.e. database querying [1])? It must be proved on an experimental basis that those components can compete with the existing natural language systems.*

The solutions to the stated problems organize the rest of the paper as follows: first, we present the general framework of an automated model generation system from the IS documentation and utterances by engineers. Next, we present the IBM's IFW solution and the model which we used in our experiments. We present FCA as the formal technique to analyze the IS model on the *object:attribute* sets. In Section 4, we present the architectural solution of the natural language processing (NLP) system which was built from open source, state-of-the-art NLP components. Then we present an idea of the conceptual model vector space. The motivation for introducing this stage to the modeling process is that it helps us deal with the modeling documentation and its topological structures numerically. Then the SOM of the conceptual model is introduced in Chapter 5. Finally, to prove the soundness of the proposed method, we provide a numerical experiment in which the ability of the system to identify concepts from user utterance is tested. The IBM Voice Toolkit for WebSphere [10] (an approach based on statistical machine learning) solution is compared with the system suggested in this paper.

2 The General Framework of the Solution

Conceptual models offer an abstract view on certain characteristics of the domain under consideration. They are used for different purposes – such as a communication instrument between users and developers, for managing and understanding the complexity within the application domain, etc. The presence of tools and methodology that supports integration of the requirement documents and communication utterance into conceptual model development is crucial for a successful development of the IS architectural framework.

In this paper, we suggest the use of SOM to classify IS documentation and IS utterance on a supervised and unsupervised basis. SOM has been extensively studied in the field of textual analysis. Such projects as WEBSOM [11] [14] have shown that the SOM algorithm can organize very large text collections and that SOM is suitable for visualization and intuitive exploration of the document collection. The experiments with the Reuters corpus (a popular benchmark for text classification) were investigated in [8]; there was evidence that SOM can outperform other alternatives.

Nevertheless, in the field of IS modeling the connectionist paradigm has been met with some skepticism. The reason is that IS architects and modelers want to give the credibility on how clusters received from document processing are related and explain the semantic meaning of the underlying topology of documents. To overcome this problem, we suggest that FCA can give more on that account by formally analyzing the set of objects and their attributes. On the other hand, when directly applied to the large data set of textual information, FCA is of little meaning for the presentation of the overwhelming lattice. Those arguments motivate integration of FCA and other text clustering techniques. In that sense, our work bears some resemblance with the work of Hotho et al. [7]. They used BiSec-kk-Means algorithm for text clustering and then FCA was applied to explain relationships between clusters. Authors of the paper have shown the usability of such approach in explaining the relationships between clusters of the Reuters-21578 text collection.

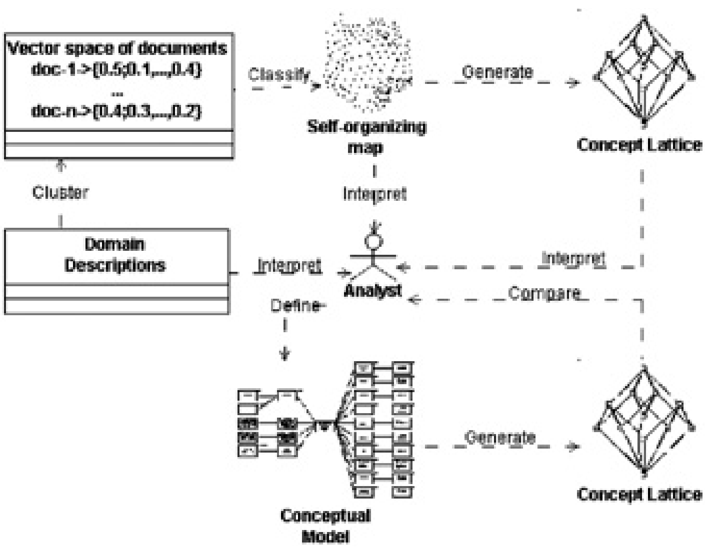

Fig. 1. Process of integration: conceptual modeling, textual description cluster detection and interpretation by use of FCA

Our approach differs in two important respects. First, our goal is not text clustering. Our goal is automated generation of the ontology from textual documents if there is no knowledge base produced by human experts. In case the knowledge base has already been developed, we seek for the method that formally measures the comprehensibility of the knowledge base topology and in case of new documents and concepts, automatically integrates them into the knowledge base.

The overall process of automatically clustering concept descriptions and then deriving concept hierarchies from SOM is presented in Figure 1. First, the corpus is created from the model concept descriptions; it is called domain descriptions in

Figure 1. Then vector space of the corpus is created using natural language processing framework, domain ontology, and WordNet ontology [15]. SOM is built and used for cluster analysis. Next, with conceptual context and concept lattice (CL) improvements are made in the understanding of cluster relationships. Simultaneously, CL is created directly from the conceptual model. The analyst can compare the lattice received from IS documentation and the lattice generated from the conceptual model. If both lattices are similar, we can say that the quality of IS documentation is acceptable.

3 Business Knowledge Bases and Formal Concept Analysis

The problem with data-centric enterprise-wide models is that they are difficult to understand. Their abstract and generic concepts are unfamiliar to both business people and IS professionals, and remote from their local organizational contexts [4]. Natural language processing and understanding techniques can be used to solve the mentioned problems. Nevertheless, before applying the NLP techniques in IS engineering, we must have some formal method to deal with the sets of *{classes, object, and attributes}* which are the products of systems of natural language processing. In this section we introduce FCA as the method for automatical building of a hierarchical structure of concepts (or classes) from the *{object:attribute}* set.

In Figure 2 (left side) we can see an excerpt from the IBM IFW financial services data model (FSDM) [9], which is a domain-specific model based on the ideas of the experts of the IBM financial service solutions center. The IBM financial services data model consists of a high level strategic classification of domain classes integrated with particular business solutions (e.g., Credit Risk Analysis) and logical and physical data entity-relationship (ER) models.

CL of the extract from the model have been produced by FCA with Galicia software [19] and are shown on the right side of Figure 2. As we can see, it is consistent with the original model. It replicates the underlying structure of the conceptual model originally produced by a human expert team; in addition, it suggests one formal concept that aggregates *Arrangement* and *Resource Item:* the two top concepts of the original model.

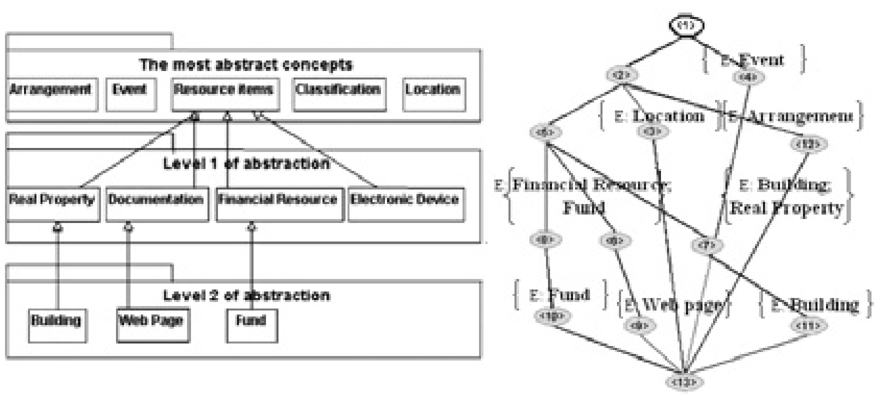

Fig. 2. Left side: a small extract from the financial services conceptual model. Right side: CL from this conceptual model. (We see that FCA depicts the structure of the conceptual model.)

FCA is used to represent underlying data in the hierarchical form of the concepts. The most adapted form in the FCA analysis for data representation is CL. Due to its comprehensive form in visualising the underlying hierarchical structure of the data and rigorous mathematical formalism, FCA has developed into a fully-fledged theory for data analysis since its introduction in the 1980s [5]. FCA has been successfully applied in many areas, but our interest in this paper is the ability to use it in the area of IS modeling. In defining the concepts and attributes, FCA is similar to the database theory and object orientated system design. Due to this fact, FCA has been often applied in class diagram design in IS [5].

For the introduction to the area of FCA, we can return to Figure 2. The conceptual model extract in the figure has 12 objects. Let us name them the set G. Let M be the set of attributes that characterise the set of objects, i.e., an attribute is included into the set M if it is an attribute for at least one object from the set G. In our example we have 137 attributes (the whole model has more than 1000 objects and more than 4000 attributes). We identify the index I as a binary relationship between two sets G and M, i.e., $I \subseteq G \times M$. In our example the index I will mark that, eg., an attribute “Interest Rate” belongs to an object “Arrangement” and that it does not belong to an object “Event”.

In order to be able to start FCA algorithms, we define a triple $\mathbb{K} := (G, M, I)$ which is called a formal context. Further, we define subsets $A \subseteq G$ and $B \subseteq M$ as follows:

$$A' := \{m \in M | (g,m) \in I \text{ for all } g \in G\};$$

$$B' := \{g \in G | (g,m) \in I \text{ for all } m \in B\}.$$

Then a formal concept of a formal context (G, M, I) is defined as a pair (A, B) with $A \subseteq G$, $B \subseteq M$, $A' = B$ and $B' = A$. The sets A and B are called extend and intend of the formal concept (A, B). The set of all formal concepts $\mathfrak{B}(\mathbb{K})$ of a context (G, M, I) together with the partial order $(A_1, B_1) \leqslant (A_2, B_2) :\Leftrightarrow A_1 \subseteq A_2$ is called the concept lattice of the context (G, M, I) .

In Figure 2, the FCA algorithm *Incremental Lattice Builder* generated 11 formal concepts. In the lattice diagram, the name of an object g is attached to the circle and represents the smallest concept with g in its extent. The name of an attribute m is always attached to the circle representing the largest concept with m in its intent. In the lattice diagram an object g has an attribute m if and only if there is an ascending path from the circle labeled by g to the circle labeled by m. The extent of the formal concept includes all objects whose labels are below in the hierarchy, and the intent includes all attributes attached to the concepts above. For example, the concept 7 has {*Building; Real Property*} as extend (the label *E:* in the diagram), and {*Postal Address; Environmental Problem Type;Owner;... etc*} as intent (due to the huge number of attributes they are not shown in the figure).

4 Vector Space Representation of the Conceptual Model

The vector space model (VSM) for document transformation into vectors is a well-known representation approach that transforms a document into a weight vector in automatic text clustering and classification. The method is based on the bag-of-words approach, which ignores the order of words in a sentence and uses basic occurrence information [18].

On the other hand, the vector space model's dimensionality is based on the total number of words in the data set and it brings difficulties for the large data sets. The document corpus of the conceptual model described above included 3587 words. The process of dimensionality reduction and noise filtering is depicted in Figure 4. All presented processes are described in detail below.

1) *Transform conceptual model.* As the first step we transform conceptual model to the Web Ontology Language (OWL) structure. The motivation behind this step is that the OWL is one of the most used standard in describing the knowledge base and we already use it in Semantic Web applications. Additional motivation for using OWL is the availability of the knowledge base development tools such as Protégé – OWL editor [12] that supports OWL standard.

2) *Extract triplet.* The triplet consists of concept name, the most abstract parent concept name – class label for a particular document, and description of the concept. To be more specific, the following steps were performed: first, we selected only concepts (entities) from 'C' level of the conceptual model and then selected the textual description of each entity. We received 1256 documents in the corpus, each document describing one concept. Each document in the corpus has been labeled with its original concept name and its top parent concept name. For example, the concept "Employee" has the following entry in the corpus: { *Concept-Employee; Parent-Individual; Top parent concept – Involved Party ; Description – An Employee is an Individual who is currently, potentially or previously employed by an Organization, commonly the Financial Institution itself...* }. We had to add textual descriptions to 254 concepts. It was done because we wanted to measure additional documentation impact on the classification accuracy of concepts. The descriptions were taken from web dictionaries. 198 concepts were removed due to short textual descriptions and our inability to supplement them from the web dictionaries. After these steps, we obtain our final corpus for the evaluation. It consists of the 1058 documents, distributed over 9 top parent concepts (*involved party, products, arrangement, event, location, resource items, condition, classification, business*).

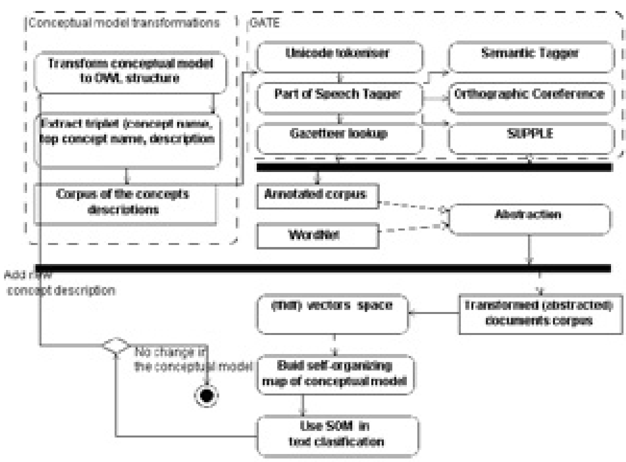

Fig. 3. The processes of dimensionality reduction and the conceptual model SOM design

3) GATE – Natural Language Processing Engine is a well-established infrastructure for customization and development of NLP components [3]. It is a robust and scalable infrastructure for NLP and it allows users to use various modules of NLP as plugins. We briefly describe modules used in our research for building vector spaces of concepts. The Unicode tokeniser splits the text into simple tokens. The tagger produces a part-of-speech tag as an annotation for each word or symbol. The gazetteer further reduces dimensionality of the document corpus prior to classification. The semantic tagger provides finite state transduction over annotations based on regular expressions. It produced an additional set of named entities, and we replaced each named entity with the class label. Orthographic Coreference module adds identity relations between named entities found by the semantic tagger. SUPPLE is a bottom-up parser that constructs syntax trees and logical forms of English sentences. We used it only to remove tokens not annotated by this module. All modules within GATE produced annotations – pairs of nodes pointing to positions inside the document content, and a set of attributes-values, encoding linguistic information.

4) *Abstraction.* The basic idea of the abstraction process is to replace terms by more abstract concepts as defined in a given thesaurus in order to capture similarities at various levels of generalization. For this purpose we used WordNet [15] and annotated GATE corpus as the background knowledge base. WordNet consists of the so-called synsets, together with a hypernym/hyponym hierarchy [6]. To modify the word vector representations, all nouns have been replaced by the corresponding concept of WordNet (‘synset’). Some words have several semantic classes (‘synsets’) and in that case we used a disambiguation method provided by WordNet – the most common meaning for a word in English was our choice. The words replaced by the GATE named entities annotation scheme were not included in the WordNet processing.

5) *Vector space.* In our experiments we used vector space of the terms vectors weighted by *tfidf* (term frequency inverse document frequency) [18], which is defined as follows:

$$tfidf\ (c,t) = tf\ (c,t) \times \log\ \frac{|C|}{|C_t|}\,.$$

where $tf\ (c,t)$ is the frequency of term t in concept description c, and C is the total number of terms, and C_t is the number of concept descriptions containing this term. $tfidf\ (c,t)$ weighs the frequency of a term in a concept description with a factor that discounts its importance when it appears in almost all concept descriptions.

5 Self-Organizing Map of the IS Conceptual Model

Neurally inspired systems, also known as the connectionist approach, replace the use of symbols in problem solving by using simple arithmetic units through the process of adaptation. The winner-take-all algorithms, also known as the self-organizing network, select the single node in a layer of nodes that responds most strongly to the input pattern. In the past decade, SOM have been extensively studied in the area of text clustering. The ideas and results presented here are of general purpose and could be applied in knowledge development by means of the connectionist paradigm in general.

SOM consists of a regular grid of map units. Each output unit i is represented by A prototype vector, $m_t = [m_{i1}...m_{id}]$, where d is input vector dimension. Input units take the input in terms of a feature vector and propagate the input onto the output units. The number of neurons and topological structure of the grid determines the accuracy and generalization capabilities of SOM.

During learning the unit with the highest activation, i.e. the best matching unit regarding a randomly selected input vector is adapted in a way that it will exhibit even higher activation regarding this input in the future. Additionally, the units in the neighborhood of the best matching unit are also adapted to exhibit higher activation regarding the given input.

As a result of training SOM with the text corpora of the IBM IFW financial warehouse conceptual model, we obtain a map which is shown in Figure 4. SOM has been trained for 100,000 learning iterations with learning rate initially set to 0.5. The learning rate decreased gradually to 0 during the learning iterations.

Table 1

Classification accuracy (CA) and average quantization error (AQE) of THE conceptual model SOM

	No hypernym	WordNet synset replacements	One level up hypernym replacements	Two levels up hypernym replacements	Three levels up hypernym replacements
CA	29.57	29.56	41.53	39.27	26.44
ACQ	4.83	4.81	4.56	4.83	4.28

It was expected that if the conceptual model vector space has some clusters that resemble the conceptual model itself, the model will be easier to understand compared with the model of a more random structure. On a closer look at the map, we can find regions containing semantically related concepts. For example, the upper right side of the final map represents a cluster of concepts “Arrangement” and the lower right side “Resource items”. Such map can be used as an interface to the underlying conceptual model. To obtain information from the collection of documents, the users may formulate queries describing their information needs in terms of the features of the required concept.

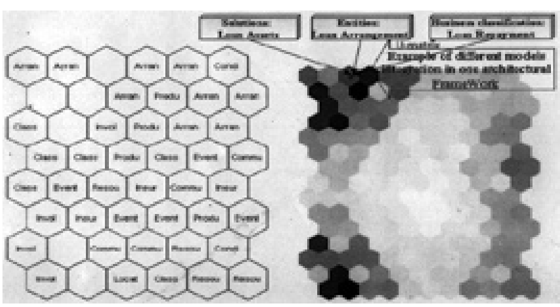

Fig. 4. SOM for the conceptual model. Labels: *invol, accou, locat, arran, event, produ, resou, condi* represent concepts: involved party, accounting, location, event, product, resource, condition.

Figure 5 shows the concept lattice computed from SOM shown in Figure 4. We obtain a list of 23 formal concepts. Each of them groups several neurons from SOM. We can find the grouping similarity of the neurons that are located in the neighborhood of each other. On the other hand, some concepts group neurons that are at some distance form each other. The basic idea of this step is that we receive a closed loop in the business knowledge engineering by an artificial intelligent agent. The agent classifies all IS textual information with the help of the SOM technique and then, using FCA, it builds hierarchical knowledge bases. For the details on how to apply FCA in cluster analysis (SOM in our case), we refer to the paper [7]. The paper describes an algorithm which has been used in our research.

The impact of the abstraction and natural language processing on the performance of the information system model can be checked with classification accuracy (CA) measure. It simply counts the minority of concepts at any grid point and presents the count as a classification error. For example, after the training each map unit has a label assigned by the highest number of concepts (Figure 4). In Figure 4, the neuron on the upper left side mapped 4 concepts with the label *Arrangement* and 2 with label *Event.* Thus, classification accuracy for this neuron is 66%. Another metric to measure classification accuracy is average quantization error AQE. It is defined as the average distance between every input vector and its best matching unit:

$$AQE = \frac{1}{N}\sum_{i=1}^{N} |x_i - b_i|$$

where N is the total number of input patterns, x_i is the vector of each pattern and b_i is best matching unit (BMU) for each pattern x_i. Findings of the influence of terms abstraction and natural language processing are shown in Table 1.

We can see that the hypernym level one is optimal compared with more abstract concepts. The phenomena can be explained by the fact that different meanings of the term, if too abstract, will be treated as the same and, as a result, semantics of discretionary power will be lost.

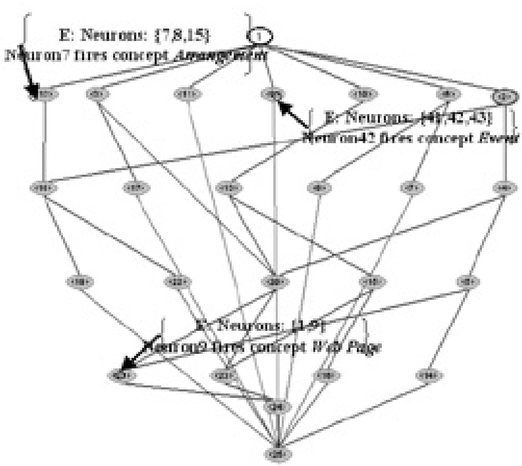

Fig. 5. Concept lattice received from the SOM presented in Figure 4.

6 The Experiment

In the previous sections, we have shown how to build a hierarchical conceptual model from IS documentation and how to verify formally the business information system model. In addition, as mentioned in the introduction, one of the objectives of this research project was to find the techniques and tools of IS modeling that give an opportunity to reuse IS model components as the final products in IS natural language interfaces. We argue that such component can be SOM of the IS conceptual model.

Reusing SOM in IS interfaces is quite simple. Each time the sentence is presented to the system, we have one activated neuron which is associated with one concept from the conceptual model. Additionally, we have the set of formal concepts associated with the activated neuron. Both the label of the activated neuron and the set of formal concepts can be used by formal language generation engines (i.e., structured query language (SQL) sentence generator for querying databases). Then the following hypothesis is formulated in this section: *SOM received from IS documentation can compete with the state-of-the-art concept identification solutions currently available in the market.*

The following experiment was conducted to test this hypothesis. IBM WebSphere Voice Server NLU toolbox, which is part of the IBM WebSphere software platform, was chosen as the solution competitive to the one suggested in this paper. From IBM presentation [10] it appeared that the system is primarily intended to support database interfaces in the telecommunication market. It was a challenging task to test it on a more complex system, i.e., a full enterprise conceptual model for the financial market.

SOM of the conceptual model and CL was used as an alternative to the IBM WebSphere Voice Server NLU solution. We adopted the black box approach for both solutions: put the training data, compile, and test the system response for the new data set. The data set of 1058 pairs *textual description:concept name* mentioned above was constructed to train the IBM NLU model. The same set was used to get SOM of the business model.

Then a group consisting of 9 students was instructed about the database model. They had the task to present for the system 20 questions about information related to the concept “Involved Party”. For example, one of the questions was *“How many customers we have in our system?”* We scored the answers from the system as correct if it identified the correct concept “Involved Party”.

Table 2

Concept identification comparison between IBM NLU toolbox and SOM of the database conceptual model

	CN = 9	CN = 50	CN = 200	CN = 400	CN = 500
IBM NLU	36.82	17.26	14.82	11.15	8.22
SOM	46.73	30.70	27.11	20.53	18.83
No additional descriptions	38.24	18.43	15.72	12.77	9.52

In the beginning only 9 top concepts were considered, i.e. all 1058 documents were labeled with the most abstract concept names from the conceptual model. For example, documents that described concepts “Loan” and “Deposit” were labeled with the concept name “Arrangement” because concepts “Loan” and “Deposit” are subtypes of the concept “Arrangement”.

Next, we increased the number of concept names that we put into the model up to 50. For example, documents that described concepts “Loan” and “Deposit” were labeled with “Loan” and “Deposit” names. Then we increased the number of concept names up to 200, 400 and, finally, 500. Table 2 shows the results of the experiment. Columns show the number of concepts. The row named *IBM NLU* represents results for the IBM WebSphere Voice Server NLU toolbox. The row named *SOM* represents results for the SOM of the conceptual model that has been constructed with the method described in this paper. The row named *No additional descriptions* represents results for SOM of the conceptual model without the 254 additional documents mentioned above. To detect the classification error, the proportion of correctly identified concepts was determined.

As we can see, the performance of the IBM system was similar to the SOM response. The behavior of the IBM system is difficult to explain because it is close system and there was no description of algorithms used. For all cases i.e. IBM, SOM and SOM without additional descriptions the performance decreased when the number of concepts increased. The solution that can increase accuracy of concepts identification is suggested by comparing results in the third and second row of Table 2. We see that the 254 descriptions we added to the system significantly improved the response of the system.

7 Conclusion

Conceptual models and other forms of knowledge bases can be viewed as products that have emerged from human natural language processing. Self-organization is the key property of human mental activity, and the present research focuses on what self-organization properties can be found in the knowledge base documentation. It has been suggested to build a conceptual model vector space and its SOM by comparing the concept lattice received from a manually constructed conceptual model and the concept lattice received from SOM of the conceptual model. We argued that if both concept lattices resemble each other, we can assert that IS documentation quality is acceptable.

The presented architectural solution for the software developers can be labor-intensive. The payoff of such approach is an ability to generate formal language statements directly from IS documentation and IS user utterance. We have shown that with SOM and FCA we can indicate inadequate concept descriptions and improve the process of knowledge base development. The presented methodology can serve as the tool for maintaining and improving enterprise-wide knowledge bases.

There have been many research projects concerning questions of semantic parsing, i.e., the automatic generation of the formal language from the natural language. But those projects have been concerned only with semantic parsing as a separate stage not integrated into the process of software development. The solution presented in this paper allows us to integrate IS design and analysis stages with the stage of semantic parsing. In this paper, we demonstrated that we can label documents and user questions with concept names of the conceptual model. In the future we hope to extend those results by generating SQL sentences and then querying databases. The present research has shown that if we want to build a comprehensible model, we must give more attention to describing concepts by the natural language.

References

1. Androutsopoulos I., Ritchie G. D., Thanisch P. Time, Tense and Aspect in Natural Language Database Interfaces. *Natural Language Engineering*, 4, 1998, 229–276.
2. Burg J. F. M., Riet R. P. Enhancing CASE Environments by Using Linguistics. *International Journal of Software Engineering and Knowledge Engineering* 8(4), 1998, 435–448.
3. Cunningham H. GATE, a General Architecture for Text Engineering. *Computers and the Humanities*, 36, 2002, 223–254.
4. Darke P., Shanks G. Understanding Corporate Data Models. *Information and Management* 35, 1999, 19–30.
5. Ganter B., Wille R. *Formal Concept Analysis: Mathematical Foundations*. Springer, Berlin-Heidelberg, 1999.
6. Hofmann T. Probabilistic latent semantic indexing. In: *Research and Development in Information Retrieval*, 1999, 50–57.
7. Hotho A., Staab S., Stumme G. Explaining text clustering results using semantic structures. In: *Principles of Data Mining and Knowledge Discovery, 7th European Conference, PKDD 2003*, Croatia. LNCS. Springer 2003, 22–26.
8. Hung C., Wermter S., Smith P. *Hybrid Neural Document Clustering Using Guided Self-organisation and WordNet.* Issue of IEEE Intelligent Systems, 2004, 68–77.
9. IBM. IBM Banking Data Warehouse General Information Manual. Available from on the IBM corporate site *http://www.ibm.com* (accessed on July 2007).
10. IBM Voice Toolkit V5.1 for WebSphere Studio. *http://www-306.ibm.com/software/* (accessed on July 2007).
11. Kaski S., Honkela T., Lagus K., Kohonen T. WEBSOM self-organizing maps of document collections. *Neurocomputing*, 21, 1998, 101–117.
12. Knublauch H., Fergerson R., Noy N. F. The Protege-OWL plugin: an open development environment for semantic web applications. Third International Semantic Web Conference. ISWC2004, Lecture Notes in Computer Science, 3298. Springer-Verlag: Heidelberg 2004, 229–243.
13. Kohonen T. Self-Organizing Maps, Springer-Verlag, 2001.
14. Lagus K., Honkela T., Kaski S., Kohonen T. WEBSOM for textual data mining. *Artificial Intelligence Review*, 13 (5/6) 1999, 345–364.
15. Miller G. A. WordNet: A Dictionary Browser, Proc. of 1st Int’l Conf. “Information in Data”, 1985, 25–28.
16. Ryan K. The role of natural language in requirements engineering. *Proceedings of IEEE International Symposium on Requirements Engineering*, IEEE Computer Society Press, 1993, 240–242.
17. Rolland C., Proix C. A Natural Language Approach to Requirements Engineering. 4th International CAiSE Conference, Manchester UK, 1992, 257–277.
18. Salton G. *Automatic Text Processing: The Transformation, Analysis and Retrieval of Information by Computer*. Addison-Wesley, 1989.
19. Valtchev P., Grosser D., Roume C., Rouane H. M. GALICIA: an open platform for lattices. In: A. de Moor, B. Ganter, editors, *Using Conceptual Structures: Contributions to 11th Intl. Conference on Conceptual Structures 2003*, 241–254.