

# Application of Conceptual Structures in Requirements Modeling

Michael Bogatyrev, Vadim Nuri Ahmetov

Tula State University,  
Lenin ave. 92, 300600 Tula, Russia

[okkambo@mail.ru](mailto:okkambo@mail.ru), [vadim-nuri Ahmetov@yandex.ru](mailto:vadim-nuri Ahmetov@yandex.ru)

**Abstract.** Requirements modeling has been applied in CASE technologies to formalize knowledge needed for constructing models of information systems. The problem is to acquire knowledge from requirements texts and represent it as intermediate requirements model for entity-relationships or object oriented modeling. Proposed approach is based on formalization of entities and their attributes as formal contexts. It is shown that formal contexts created on the set of conceptual graphs extracted from requirements text may serve as data source for requirements models have been applied in real CASE technologies.

**Keywords:** CASE technology, requirements modeling, conceptual graphs, conceptual structures, conceptual requirements model, Sybase PowerDesigner.

## 1 Introduction

In one of early works of John Sowa [1] conceptual graphs were discovered as intermediate models between natural language and database interfaces. Following this idea in this paper conceptual graphs are used as intermediate model between natural language and *requirements models* which have been applied in database CASE technologies.

*Requirements Modeling* has been applied in *Requirements Engineering* [3] to formalize a knowledge needed for constructing models of information systems in CASE technologies. Modern CASE technologies, for example technology of Sybase PowerDesigner [5], realize Requirements Modeling as a real working tool. Here a text of requirements of a project is a data source which contents (words or phrases) beget requirements. Every requirement is an object in requirements model. It has a name and attributes - *type, status, priority, risk*, etc. In the requirements model every requirement is connected with elements of other CASE-models, for example with elements of Entity-Relationship Diagrams (ERD) or UML diagrams. Connection means that when a CASE-model is processed it must be done by meeting demands of requirements. The instrument of Requirements Modeling is actual in big projects with complex textual requirements. It is also important in supporting life cycle of the system to be designed [4].

A challenging problem in Requirements Modeling is the problem of creating requirements model from natural language text of requirements.

Significant numbers of works in the area of Requirements Modeling have been devoted to this problem. The most of them treat it as direct mapping text to CASE-models and a requirement considered as a text. All such works can be divided into two sets: one set of works is devoted to derive a family of Entity Relationship models (plain or extended ERD) from natural language texts ([6] - [8]); another set of works is about object oriented models represented by class diagrams ([9], [12]). These works are based on the assumption that meaning of concepts being extracted from a text can be derived from grammar structures of natural language. Heuristic rules of implementing properties of parts-of-speech and their functions in sentences are applied here. Besides English language, decisions for some other languages including German [8] and Japanese [9] have been presented. Modeling by analyzing contexts in requirements texts is presented in [10]. Some examples of real requirements modeling systems are presented in [16].

In spite of many existing results here including ones oriented on grammars of concrete languages, full automation of CASE-models design from requirements texts is fundamentally impossible. The text of requirements actually contains more or less portion of information needed for creating a CASE-model and textual data could not be mapped exactly to the data of CASE-model.

Therefore the central Requirements Modeling problem needs to be formulated in its natural form – as a problem of creating requirements model from natural language text. This requirements model has to be treated as separate intermediate model between requirements texts and CASE-models.

This paper is based namely on that approach to Requirements Modeling. It is shown that formal contexts created on the set of conceptual graphs extracted from requirements text may serve as data source for requirements models have been applied in real CASE technologies.

## **2. Conceptual Requirements Modeling**

The term *Conceptual Requirements Modeling* is appropriate to denote the fact of applying *Conceptual Structures* in *Requirements Modeling*. Domain of Conceptual Structures combines *conceptual graphs* [2] and Formal Concept Analysis [13] techniques and now can be considered as general approach for modeling many problems in Data Mining and Text Mining areas.

### **2.1 Conceptual Structures as Requirements Model**

Both Entity Relationships and Object Oriented CASE-models use *objects* and *attributes*. Attributes belong to *entities* in ERD and to *objects* in Object Oriented models (OOM).

An *entity* is an object from real world having finite set of attributes. Entity name denotes this set of attributes, for instance,  $Student\{Name, Date\_Birth, \dots\}$ .

One of the crucial principles of Entity Relationship modeling claims that *every entity has only generic attributes* i.e. attributes which characterize only entity itself. Describe this by the following way. Consider the set of data types

$T = \{D_1, D_2, \dots, D_n\}$  consisted of domains  $D_i$ . Every domain is ordered set of data of certain type, for example *character*, *numeric*, *date*, etc. The set of attributes  $A = \{a_i\}$ ,  $a_i \in D_j$  is multiset, which contains examples of domain elements. We

denote every  $i$ -th example of entity as  $e_i = \{A_i\}$ ,  $A_i \subset A$ . All examples  $e_i$  constitute an *entity type*  $E_i = \{e_i\}$ , which attributes are the same for all examples of entities. Then the *generic feature* of attributes is described by condition:

$$\text{for } E_i \neq E_j \quad A_i \cap A_j = \emptyset \quad (1)$$

Very often this demand is not met in practice. For example:  $E_1 = Student\{Name, Date\_Birth, \dots\}$  and  $E_2 = Teacher\{Name, Date\_Birth, \dots\}$  have similar subsets of attributes. To hold the condition (1) in CASE-technology one must rename attributes of entity in the example above. Note that the mapping sets of attributes to entities (or vice versa) could not be rigorously formalized as a function - it is a *relation*. An appropriate way of expressing it is formal context [13].

Consider a formal context  $(E, A, R)$ , where  $E = \bigcup_i E_i$  and  $R$  is a relation which establishes the facts of belonging attributes to entities. Formal context  $(E, A, R)$  may be represented by  $[0, 1]$ -matrix in which units mark correspondence between entities  $E$  and attributes  $A$ . If the set  $A$  is ordered by its subsets  $A = \{A_1, A_2, \dots, A_k\}$  and the condition (1) is hold then the context matrix has block-diagonal structure

$$C = \text{diag}[C_1, C_2, \dots, C_k] \quad (2)$$

as it is shown on Figure 1. Every sub matrix  $C_j$  represents a relation on subsets of entities where entities are grouped into *associated entities* which are associated by closed subsets of attributes. An example of associated entity  $\tilde{E}_1 = Human \{Student, Teacher, Dean\}$  is shown on Figure 1.

		$A$			
		$A_1$	$A_2$	...	$A_k$
Human	$E$				
	Student Teacher Dean	$C_1$			
	$\tilde{E}_2$		$C_2$		
	...		...		
	$\tilde{E}_k$				$C_k$

**Fig. 1.** Formal context for associated entities.

Every associated entity  $\tilde{E}_i$  unites maximum number of entity attributes and not all of them belong to all entities inside an association, so association context matrices  $C_i$  may be sparse. Their attribute subsets  $A_i$  may have some subsets of attributes  $\tilde{A}_i \subseteq A_i$  which belong to all entities in association. If we construct another context  $(\tilde{E}, \tilde{A}, R)$  regard to associated entities and attributes  $\tilde{A} = \{\tilde{A}_i\}$  then this context sub matrices  $\tilde{C}_i$  will be completely filled by units

Sub matrices  $\tilde{C}_i$  may represent formal concepts [13] on the context  $(\tilde{E}, \tilde{A}, R)$ . Formal concept on the context  $(\tilde{E}, \tilde{A}, R)$  is a pair of subsets  $X \subseteq \tilde{E}, Y \subseteq \tilde{A}$  together with pair of mappings  $\varphi: \tilde{E} \rightarrow \tilde{A}, \psi: \tilde{A} \rightarrow \tilde{E}$  realizing so called *Galois connection* [14]. A pair  $(\varphi, \psi)$  is a Galois connection between the partially ordered sets (posets)  $(\tilde{E}, \sqsubseteq), (\tilde{A}, \supseteq)$  if the following conditions hold: for all

$$x \in X, y \in Y \quad x \sqsubseteq \psi(\varphi(x)), \varphi(\psi(y)) \supseteq y. \quad (3)$$

Galois connection is that type of mapping which “synchronously” conserves sets orders or maps sets orders from one poset to another. The set of formal concepts on a context forms a conceptual lattice [13].

Considered conceptual structures – formal context and formal concepts – may serve as an instrument for constructing requirements models. They unite objects and attributes by relations and have important property of completeness: as formal context as formal concepts are complete objects with certain informational content extracted

from the text of requirements. Apparently this content must be represented in CASE-models so the context and formal concepts constitute a kind of requirements. They may be considered as *Conceptual Requirements Model*.

## 2.2 Conceptual graphs acquisition and processing.

To extract objects and their attributes from requirements text, the approaches mentioned in the Introduction section may be applied. Conceptual graphs are appropriate for it due to the following reasons:

- if successfully acquired from text, conceptual graphs represent compact model for discovering objects and their attributes - there may be a set of conceptual relations in a graph which depict connection between objects and their attributes;
- conceptual graphs naturally belong to Formal Concept Analysis paradigm and have been successfully applied for constructing formal contexts [18].

Using conceptual graphs, another problem becomes actual – the problem of acquisition conceptual graphs from texts.

We use our software for conceptual graphs acquisition from natural language texts [15]. The software is based on existing approaches of lexical, morphological and semantic analysis. *Semantic roles labeling* [19] is applied as the main instrument for constructing relations in acquisition algorithm. The algorithm works with our recently developed *controllable grammatical templates*. Using these templates, it is possible to adapt acquisition algorithm as to certain language grammar (Russian grammar in the current version of the system) as to some peculiarities of concrete language. User interface has also tools for recognizing incorrect conceptual graphs. Incorrect conceptual graph is a graph having *isolated concepts* i.e. concepts which are not connected to other concepts by relations.

Conceptual graphs are acquired from subtitles of requirements text and from text sections. It is interesting to find similarities between graphs acquired from subtitles and graphs acquired from text sections since some terms (objects) declared in a subtitle may be mentioned and concretized in a section text. We apply measures of similarity of conceptual graphs which we used in our experiments of conceptual graphs clustering [20].

All acquired correct conceptual graphs are processed to extract objects and their attributes. The way of extraction is based on fixing certain set of conceptual relations presented in derived graph. There are trivial and non trivial patterns of concepts and corresponding relations which may exist in a graph. If standardized text of requirements is a source for graph acquisition then it is possible to create special templates for graph acquisition algorithm.

## 2.3 Creating and processing formal contexts

Conceptual graph represents semantics of only one sentence. Important information about objects and their attributes may be presented in several various sentences. To

collect it we use formal context. A context with associated entities having block-diagonal structure (2) contains the needed information.

Formal context is created as  $[0, 1]$  matrix in which correspondence between objects and their attributes is supported. That correspondence is established after processing conceptual graphs and it is not enough to say that condition (1) is true on the context's structure created automatically on the acquired sets of objects and attributes. So we apply block-diagonal decomposition of context matrix to find its structures similar to shown on Figure 1. Any algorithm of block-diagonal matrix decomposition works so that it is equivalent to some permutations of rows and columns of matrix. As a result initial correspondence between objects and attributes may be disrupted. The sets of objects and attributes in a context are partially ordered so only those permutations which conserve this feature are allowed.

### 3. Conceptual Requirements Modeling System Realization

The approach we propose brings additional functionality to those CASE technologies which work with conceptual and object oriented models. The Sybase PowerDesigner CASE system [5] is one of the few systems where Requirements Modeling is really exploring.

Sybase PowerDesigner CASE technology supports requirements modeling with natural language texts as its input. Figure 2 illustrates the principle of requirements modeling in PowerDesigner [5].

As it is shown on the Figure 2 PowerDesigner processes formatted MS Word textual documents. Requirements model on the Figure 2 is consisted of two elements: requirements which are title and headings of sections and subsections of the document and *traceability matrices* which represent various connections between requirements and between requirements and elements of created CASE-models. Title and headings are treated as elements of requirements model. The text between two headings is treated as the requirements object's comment.

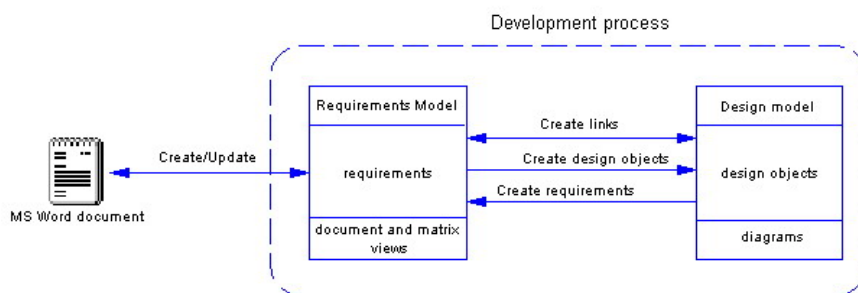


Fig. 2. Principle of requirements modeling in Sybase PowerDesigner

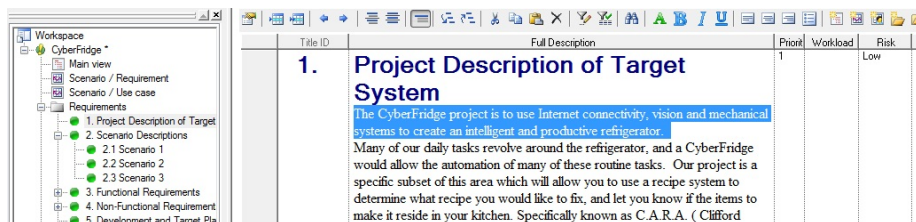
Detecting requirements as title and headings of sections and subsections of a text is the only current function of natural language processing in PowerDesigner. Having such requirements, user manually applies them in constructing CASE-models by

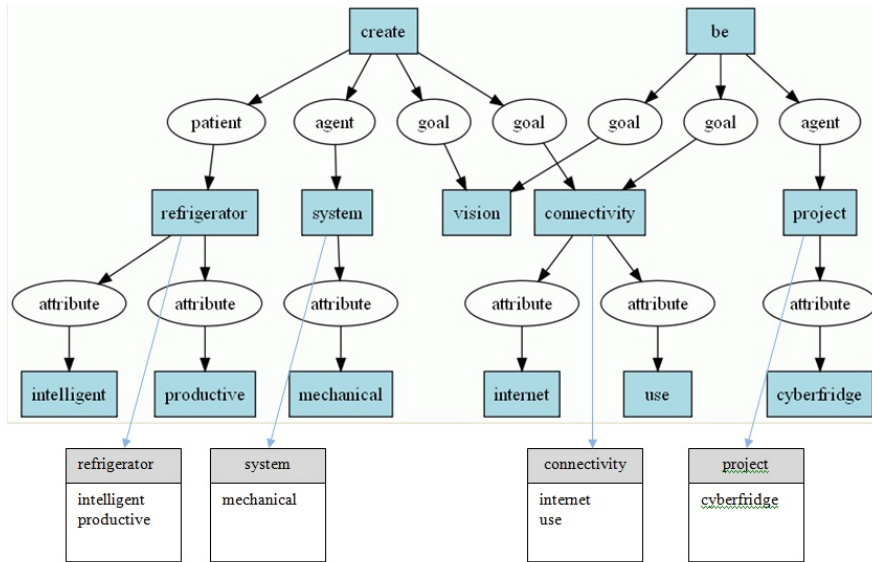
setting entities and relationships, objects and their attributes: type, status, priority, risk. Using traceability matrices user creates tools for checking connections between various objects of models.

Additional functionality to this technology is caused by the fact that requirements text between two headings has been also processed to extract objects and attributes united by formal context. It takes place by the following.

1. Conceptual graphs are acquired from the whole text of requirements. Fixed set of conceptual relations (for example *genitive* and *attribute* relations) in conceptual graphs is applied to select candidate pairs of *objects* - *attributes* to form a context.
2. Context matrix is formed so that only objects with more than one attribute have been included in the matrix. This is the way to select objects significant for constructing CASE-models.
3. Initial sparse context matrix is transformed by using linear algebra methods for block-diagonal decomposition. The problem of keeping order in the sets which form context is actual here. Associated entities may be ordered by hierarchy relation, for example as *Student*, *Teacher*, *Dean* on Figure 1. As usual attributes are less ordered and can be permuted for block-diagonal decomposition.
4. Interaction with user by special interface and visualization is very important since the process of creating CASE-models still remains closely depended on developer's skill. In the current experimental version of the system there is user interface to show every subset of object and its attributes obtained from conceptual graphs. User can correct this set.

Figure 3 illustrates this technology on the fragment of *CyberFridge* project included in PowerDesigner as an example of requirements modelling. On the figure we combined two windows: on the top of the figure there is PowerDesigner interface window showing how requirements are represented, below there is interface window of our system visualizing conceptual graph corresponded to the first sentence highlighted in the top window. Only *attribute* relation was processed here and candidate pairs of *objects* – *attributes* are shown. Later, analyzing other graphs and the context created on the whole set of candidate pairs we keep only *refrigerator* and *cyberfridge* entities as requirements.





**Fig. 3.** The example of visualization of conceptual graph and *objects – attributes* pairs of the sentence of requirements text: “*The CyberFridge project is to use Internet connectivity, vision and mechanical systems to create an intelligent and productive refrigerator*”.

On the standard way of creating requirements model in PowerDesigner user takes only headlines of requirements text (“*Project Description of Target System*” on Figure 3) as requirements objects and treats remaining text as comments. Conceptual Requirements Modeling System extends functionality of requirements modeling in PowerDesigner realizing more complete text processing.

#### 4. Preliminary Results and Future Work

The first version of the system of Conceptual Requirements Modeling was tested on various Russian texts of requirements being structured according to the standard [17]. We also started to process English texts as it is shown on Figure 3.

First results obtained from experiments demonstrate the following.

1. Conceptual graphs are valid for extracting objects and attributes from natural language texts of requirements and can deliver specific new information for CASE-models developer.

2. Formal context serves as a tool for collecting entities and their attributes for Entity Relationship and Object Oriented Modeling and selects objects significant for constructing CASE-models.



The way of developing proposed approach is mostly experimental and its final effectiveness can be confirmed after series of additional experiments and corresponded changes in the algorithm.

Future work is planned in the following directions.

1. Extending the set of relations which is applied to select candidate pairs of *objects - attributes* to form a context. For example the *goal* relation on Figure 3 is also informative as *attribute* relation.
2. Discovering the way of implementing formal concepts on formal context in the approach. Specifically, if formal concepts exist on the context do they form additional objects significant for CASE modeling?
3. Deep integration proposed approach with CASE technology, particularly with Sybase PowerDesigner.

We also plan to expand the set of our controllable grammatical templates by including English language grammar to it.

## References

1. Sowa, J.F.: Conceptual Graphs for a Data Base Interface. IBM Journal of Research and Development 20(4): 336-357 (1976).
2. Sowa, J.F., Knowledge Representation: Logical, Philosophical, and Computational Foundations, Brooks Cole Publishing Co., Pacific Grove, CA, (2000).
3. Young, R.: The Requirements Engineering Handbook. Artech House Publishers (2004)
4. Blanchard B. S., Fabrycky, W. J.: Systems Engineering and Analysis, Fourth Edition. Prentice Hall. (2006)
5. PowerDesigner 15.2 Requirements Modeling. Sybase documentation, DC 00121-01-1520-01. February 2010
6. Chen, P.: English Sentence Structure and Entity-Relationship Diagrams. Information Science, 29(2-3): 127-149 (1983)
7. Hartmann, S., Link, S.: English Sentence Structures and EER Modeling. In Proc. of 4<sup>th</sup> Asia-Pacific Conference on Conceptual Modelling (APCCM2007), 27-35 (2007)
8. Tjoa, A. M., Berger, L.: Transformation of Requirement Specifications Expressed in Natural Language into an EER Model. LNCS, vol. 823, 206-217. Springer-Verlag, Berlin, Heidelberg (1994)
9. Hasegawa, R., Kitamura, M., Kaiya, H. and Saeki, M.: Extracting Conceptual Graphs from Japanese Documents for Software Requirements Modeling. In Proc. Sixth Asia-Pacific Conference on Conceptual Modelling (APCCM 2009), Wellington, New Zealand. CRPIT. 87-96.(2009)
10. Lee, S., Kim, N., Moon, S.: Context-Adaptive Approach for Automated Entity Relationship Modeling. Journal of Information Science and Engineering v. 26, 2229-2247 (2010)
11. Saeki, M., Horai, H., Enomoto, H.. Software Development Process from Natural Language Specification. In Proc. of 11th International Conference on Software Engineering, 64-73. (1989)
12. Overmyer, S. Lavoie, B., Rambow, O.: Conceptual Modeling through Linguistic Analysis Using LIDA. In Proc. of 23rd International Conference on Software Engineering (ICSE'01), 401-410. (2001)

13. Ganter, B., Stumme, G., Wille, R.: Formal Concept Analysis, Foundations and Applications. LNCS. 3626, Springer. (2005)
14. Birkhoff, G.: Lattice Theory. Providence, RI: Amer. Math. Soc. (1967)
15. Bogatyrev, M. .Y, Mitrofanova, O. A., Tuhtin, V. V.: Building Conceptual Graphs for Articles Abstracts in Digital Libraries. - Proceedings of the Conceptual Structures Tool Interoperability Workshop (CS-TIW 2009) at 17<sup>th</sup> International Conference on Conceptual Structures (ICCS'09), 50-57. Moscow. (2009)
16. Omar, N., Paul, H., Paul, M. K.: Heuristics-Based Entity-Relationship Modelling through Natural Language Processing, Proceedings of the Fifteenth Irish Conference on Artificial Intelligence and Cognitive Science, Galway-Mayo Institute of Technology (GMIT), 302-313. Castlebar, Ireland. (2004)
17. Information technology. Set of standards for automated systems. Technical directions for automated system making. (GOST 34.602-89). <http://www.vniiki.ru/document/4144866.aspx>
18. Wille, R.: Conceptual Graphs and Formal Concept Analysis. Proceedings of the Fifth International Conference on Conceptual Structures: Fulfilling Peirce's Dream. 290 - 303. Springer-Verlag, London. (1997)
19. Gildea D., Jurafsky D.: Automatic labeling of semantic roles. Computational Linguistics, 2002, v. 28, 245-288. (2002)
20. Bogatyrev, M. Y., Terekhov, A. P.: Framework for Evolutionary Modeling in Text Mining. - Proceedings of the SENSE'09 - Conceptual Structures for Extracting Natural language Semantics. Workshop at 17<sup>th</sup> International Conference on Conceptual Structures (ICCS'09), 26-37 (2009)