FROM NATURAL LANGUAGE REQUIREMENTS TO A CONCEPTUAL MODEL

Christian Kop, Günther Fliedl, Heinrich C. Mayr

Alpen-Adria Universität Klagenfurt
Applied Informatics/Application Engineering
Universitätsstasse 65 – 67, 9020 Klagenfurt
(chris | guenther | heinrich)@ifit.uni-klu.ac.at

ABSTRACT

In literature it is described in great detail how class diagrams and ER diagrams or UML class diagrams are derived from natural language sentences. It is normally assumed, that there is a direct correspondence between natural language elements (e.g., words) and conceptual model elements. We do not strictly follow this assumption because of the complexity of natural language with its ambiguities and ellipsis. Hence in this paper a stepwise generation of a conceptual model out of natural language requirements sentences is proposed. According to the ideas of MDA we assume that automatic transformation steps from the source model (in our case natural language) to the target conceptual model (e.g., UML class diagram) make sense. In addition to that we suggest that the designer should play an important part during transformation. It is furthermore proposed to introduce an interlingua which helps to detect defects and provides traceability between sentences and the model elements.

Index Terms – natural language processing, interlingua, conceptual modeling, defect detection

1. INTRODUCTION

In most cases the requirements are presented on two levels: the level of end user needs and the level of developers or requirements engineers models. End user requirements usually are expressed via natural language; requirements handled by engineers are usually expressed through formal, conceptual models. In many cases this diverging way of representing knowledge is the main reason for misunderstandings between users and engineers concerning initial requirements. The discrepancy disables the possibility of validating requirements, which is an important step in the process of requirements engineering.

To handle such problems we proposed an intermediate level for requirements representation, an interlingua connecting the natural language level of the end user and conceptual model level produced by engineers. The approach provides instruments for the representation of intermediate results and the traceability between intermediate results and the original sentences. It supports automated mapping from natural language requirements to interlingua specifications and automated mapping from the interlingua representation to the conceptual models.

The linguistic processing step focuses on the transfer of written textual requirements to an interlingua, the so called Pre-design Model. The "Klagenfurt Conceptual Pre-design Model (KCPM)" [6] provides a glossary and a graphical representation and it is used as a basis for the mapping to the conceptual model (e.g., UML). We propose that the basic notions introduced in this interlingua should correspond to hypothetical basic linguistic categories like nouns, verbs, etc. Thus, the goal of the whole process which is called NIBA ("Natürlichsprachliche Informationsbedarfsanalyse") is to automate the process of producing pre-design models by extracting their entries from the end-user's natural language requirements statements.

To enhance the mapping process a specific framework for annotating natural language descriptions on different layers was developed.

The paper is structured as follows. In the next section the related work is described. The linguistic processing step is introduced in Section 3. Section 4 explains the interpretation step. Section 5 focuses on the interlingua and their possibilities. Section 6 gives an overview of the mapping to the conceptual model. The paper is summarized in Section 7.

2. RELATED WORK

The interpretation of natural language has a long tradition. In earlier approaches heuristics were proposed. Some of these approaches were described in [3] [1] [8] [7]. Chen presented 11 rules to generate conceptual model elements (entity types and relationship types) from structured sentence. Excerpts of these rules can be found in the next listing [3].

- (Rule 1) A common noun in English corresponds to an entity type.
- (Rule 2) A transitive verb in English corresponds to a relationship type in an ER diagram.
- (Rule 3) An adjective in English corresponds to an attribute of an entity in an ER diagram.
- (Rule 4) An adverb in English corresponds to an attribute of a relationship in an ER diagram.
- (Rule 5) If the sentence has the form: „There are … X in Y“ then we can convert it into the equivalent form „Y has … X “.
- (Rule 7) If the sentence has the form „The X of Y is Z“ and if Z is not a proper noun, we may treat X as an attribute of Y.

Abbot [1] used heuristics for the generation of program specifications. Parsing techniques were introduced in [2] and [11]. NL-OOPS [14] uses the LOLITA [15] natural language processing toolkit with an internal knowledge base to generate first cut conceptual models. Meanwhile tagging and chunking is the state of the art for the linguistic step. In [13] an approach is described which uses part of speech tagging and morphological analysis for the generation of conceptual model element candidates. Additionally an ontology (world model) was used to refine the candidates for the project specific conceptual model (discourse model).

3. LINGUISTIC PROCESSING

The system solves the task of Natural Language Processing of English requirements texts by producing chunked and semantically annotated text, which is made ready for the KCPM modeling notions extraction in the interpretation stage of the project. In a first stage it accepts the tagged sentences which are produced by QTag [16]. This output is refined and certain structures are chunked together. Figure 1 in the appendix shows such a chunk tree representing the syntactic structure including phrasal, feature inheriting nodes.

This chunking output was processed by a modular system of linguistic subsystems including the following functions:

- The identification of compound nouns. We suppose that unclear compound boundaries are very often motivated through ambiguity of complex terms, e.g., the implicit structure of compounds or other groups of words.
- The extraction and generation of inflectional word forms.
- Extraction of derivational morphological information.
- The identification of multi-words units and idiomatic expression identification. This is made possible by dynamically extending linguistic knowledge inside the lexicon component.
- Verb subclass identification. The filtered verb classes are based on the NTMS-system (“Natürlichkeitstheoretische Morphosyntax”) [4] included in the NIBA framework.

4. INTERPRETATION

4.1 General guidelines for interpretation

Following the different approaches mentioned in the related work section, the following can be learned for the interpretation of natural language sentences:

- Common (individual) nouns are candidates for classes and attributes.
- An adjective and a noun together are candidates for specialized classes.
- Proper nouns are candidates for instance labels.
- A transitive verb is a candidate for a relationship type.
- The nouns related to the verbs are the involved classes of the relationship type.
- Also prepositions can be candidates for relationship types.

In other words, given a source language (e.g., natural language) and a “meta model” (i.e., the grammar description of the sentence) as well as a target language (e.g., a conceptual model and its meta model), certain instances of the source language can be mapped to instances of the target language. This is achieved by defining equivalences between syntactic structures of the source model and syntactic structures of the target model.

These general rules must be adopted for the certain situation (i.e., the annotated natural language). In our case the NTMS was used for annotating the natural language sentences with syntactic grammar information. Since the NTMS defines N0 as a noun and N3 as a noun phrase, a class can be derived from a noun (N0) or noun phrase (N3) respectively. If we find a verb (V0) together with two noun phrases then a relationship can be derived from such a pattern. Figure 1 in the appendix shows such an example.

Although these and other heuristics are commonly used they cannot really support the interpretation. The next section will explain some difficulties of interpretation.

4.2 Problems of Interpretation

The problems of interpretation arise since the same syntactic structure of a phrase can be interpreted differently. A typical example of this problem is that the combination of an adjective and a noun can be seen as a specialization of that noun. It is also possible that the adjective together with the noun is the needed concept. Another problem: It is not always possible to distinguish between a class and an attribute just by analyzing one single sentence. In literature [11] the subject-predicate-object structure with the predicate "has" (e.g., X has Y) is interpreted as follows. The subject X is a class and the object Y is an attribute. However in [9] it was shown that the verb "has" is very ambiguous.

Since mainly syntactic structures are analyzed and mapped to elements of the conceptual model there is no guarantee that all the extracted elements are relevant for the target model. There is no guarantee that the model assembled only with the extracted elements will be complete or consistent. Even worse if an arbitrary text is taken for analyzing and interpretation there is no guarantee that the intention of the customer fits with the intentions of the designer.

3.4 Solution

As one possible solution it is necessary to give the designer the freedom to select those extracted model elements which seem to be necessary for the target model. Furthermore it is necessary to introduce an interlingua. This interlingua presents the designer the result of the extraction process and the designer can maintain and refine the results. Hence the model presented in the interlingua does not represent the final result or final conceptual model. It represents a intermediate result that must be discussed, refined and improved. A tool was implemented with which the designer can select necessary model elements and manage the elements in the model of the interlingua. This also includes a tool feature for the mapping from the interlingua to the conceptual model.

5. INTERLINGUA

5.1 Overview

According to the underlying paradigm of how a stakeholder perceives the "world", two types of conceptual modeling approaches can be distinguished:

- Entity type and object oriented approaches.
- Fact oriented approaches.

In the first paradigm the "world" is seen as a world of objects which have properties. Therefore a clear distinction is made between object and object types respectively and their properties. Representatives of this paradigm are the classical ER approach and UML. Fact oriented approaches on the other hand see the "world" as a world of facts. Facts describe objects and their roles within a relationship. No distinction is made between objects and their properties. Every concept is treated equally in a first step. Representatives of this kind of paradigm are NIAM [7] and its successor ORM [5]. Both approaches have pros and cons. Object oriented approaches look very compact. In a typical object oriented class diagram attributes are embedded in the class representation. No additional connections between classes and attributes are necessary which would expand the diagram. On the other hand, many revisions must be made if such a diagram is used too early in the design phase. Due to information that is collected, classes might become attributes and attributes might become classes. According to [5] this is a reason why fact oriented approaches are better suited to be used as an interlingua.

Since the interlingua is placed before the conceptual model during an early phase of design the fact oriented paradigm was preferred. Nevertheless there must also be the necessity to provide an easy transformation from the interlingua to a conceptual model like UML since it is actually the standard for conceptual modeling. Hence the interlingua for conceptual modeling of structural aspects of an information system consists of the following basic notions:

- Thing type: Any notion which is important in a certain universe of discourse is treated as a thing type. Since attributes are not defined also notions like person name, course id etc. are seen as thing types.
- Connection type: Connection types relate thing types to each other. Special connection types like generalization or aggregation can be defined.

The aim of the interlingua is also to be a support for all kinds of stakeholders (designers and end users). Therefore a graphical and glossary based representation was used for the collection of requirements (see Figure 3 in the appendix for the graphical representation – the glossary representation is hidden).

5.2 Defect detection support

Beside the purpose to provide a communication platform between stakeholders, the interlingua can also support the detection of structural inconsistencies and incompleteness. The simplest one can be detected if the designer takes a look at the cardinality definitions of the connection types. As it can be easily seen, all of these cardinality descriptions have a "?..?". This means that cardinalities could not be extracted from the textual description.
Another possibility is to count the number of connection types of a thing type. This is described in detail in [12]. With this strategy, centered thing types

can be detected (see Figure 4 in the appendix). The more connection types a thing type has, the more centered or important it is. Such centered thing types appear with a bigger rectangular and in another color (e.g., green) than other thing types which seem to be less important. However, this must not necessarily reflect the end users intention. Therefore this strategy is used to confront the end user with the result and to discuss the result with him. For instance if the end user wonders why certain thing types like course and professor are not so important (they appear in white color and the rectangular is not so big as the rectangular for assistant or employee) then this can be the hint for a defect in the original specification.

If a mapping preview is made, then orphan classes [10] can be detected. The Figure 5 shows such a case for the university example. In this case thing types like university, faculty, department, assistant, employee, professor, budget, ut8 and ut3 were detected to be class candidates. All the thing types which appear in white color are currently candidates for attributes. Once again this is not the final result but a starting point for communication, discussion and refinement. As can be seen in Figure 5, professor, budget, faculty and university do not have any related attributes. Hence the mapping preview gives also hints for defects.

5.3 Traceability

Sentences from which thing types and connection types can be extracted are also stored as "Sources" in the interlingua model. If a thing type was extracted from the sentence, then a relation between the thing type and the sentence exists. The same holds for connection types.

6. MAPPING TO THE CONCEPTUAL MODEL

In order to guarantee the mapping to a conceptual model rules are applied. These rules can be classified into

- Laws vs. proposals.
- Direct vs. indirect rules.

Laws are much stricter than proposals. If a mapping rule is a law than a mapping to a certain target concept (e.g., class) cannot be ignored otherwise the syntax of the conceptual target model will be incorrect. Proposals on the other hand only give hints. The syntax of the target model will not be wrong if these hints are ignored.

An indirect rule not only uses the semantic relationship to decide about the mapping but also information about previous mappings. For example, if a concept X is already mapped to an attribute and a concept Y is related to that attribute X then an indirect rule for Y detects a mapping possibility (Y will become a class).

This mapping approach also applies meta-rules to resolve conflicting situations between the rules. An example of a meta rule is: "Laws overrule proposals".

7. CONCLUSION AND FUTURE WORK

In this paper an overview of a mapping process from natural language descriptions to a conceptual model was given. It was also described that such a process is not straight forward. Instead the designer must handle problems. As one possible solution the interlingua (KCPM) was introduced. This model gives the designer an overview of the output of natural language processing and provides him with some help to improve it. Without generating the UML target model, he is able to revise it. Different presentation techniques (e.g., graphical view and glossary view) make it possible to communicate with the end user.

In future, it is planned to find more possibilities to detect defects. These defect detection strategies should then be applied on the notions which were extracted from English or from German requirements sentences.

8. REFERENCES

[1] R.J. Abbot, "Program Design by Informal English Descriptions," Communication of the ACM, Vol. 26 No. 11, pp. 882 – 894, 1983.

[2] E. Buchholz, H. Cyriaks, A. Düsterhöft, H. Mehlan, B. Thalheim, B.. "Applying a Natural Language Dialogue Tool for Designing Databases,". International Workshop on Applications of Natural Language to Databases (NLDB'95), pp. 119 – 133, 1995.

[3] P. Chen "English Sentence Structure and Entity Relationship Diagrams," International Journal of Information Siences, Vol. 29., pp. 127-149, 1983

[4] G. Fliedl, Natürlichkeitstheoretische Morpho-syntax – Aspekte der Theorie und Implementierung, Gunter Narr Verlag Tübingen, 1999.

[5] T. Halpin, "UML Data Models from an ORM Perspective Part 1," Journal of Conceptual Modeling 1998.

[6] H.C. Mayr, Ch. Kop, "A User Centered Approach to Requirements Modeling, Proceedings Modelierung," Lecture Notes in Informatics LNI, p-12, GI-Edition, pp. 75-86, 2002.

[7] G.M. Nijssen, T.A Halpin, Conceptual Schema and Relational Database Design – A fact oriented approach. Prentice Hall Publishing Company, 1989.

[8] M. Saeki, H. Horai, H. Enomoto, "Software Development from Natural Language Specification," Proceedings of the 11th International Conference on Software Engineering, pp. 64 – 73, 1989.

[9] V.C. Storey, "Understanding Semantic Relationships," VLDB Journal, Vol. 2, pp. 455 – 488., 1993.

[10] B. Tauzovich, "An Expert System for Conceptual Data Modeling," Proceedings of the 8th International Conference on Entity Relationship Approach, North Holland Publ. Company, pp. 205 – 220, 1989.

[11] A.M. Tjoa, A.M.; L. Berger, "Transformation of Requirement Specification Expressed in Natural Language into an EER Model, " Proceedings of the 12th International Conference on Entity Realtionship Approach, Springer Verlag, New York, pp. 127-149, 1993.

[12] Ch. Kop, "Visualizing Concetual Schemas with their Sources and Progress," International Journal on Advances in Software, Vol. 2. u. 3., pp. 245 – 258, 2009.

[13] H. M. Harmain, R. Gaizauskas, "CM-Builder: An Automated NL-based Case Tool," 15th IEEE International Confernce on Automated Software Engineering (ASE'00), pp. 45 – 54, 2000.

[14] L. Mich, J. Mylopoulos, N. Zeni, "Improving the Quality of Conceptual Models with NLP Tools: An Experiment," Technical Report DIT-02-0047, Dept. of Information and Communication Technology, Univ. of Trento, 2002.

[15] R. Garigliano, R. Morgan, M. Smith, "The LOLITA System as a Contents Scanning Tool," Proceedings of the 13th International Conference Artificial Intelligence, Expert Systems, and Natural Language Processing, 1993.

[16] D. Tufis, O. Mason, "Tagging Romanian Texts: a Case Study for QTAG, a Language Independent Probabilistic Tagger," Proceedings of the First International Conference on Language Resources & Evaluation (LREC), Granada (Spain), p.589-596, 1998.

APPENDIX

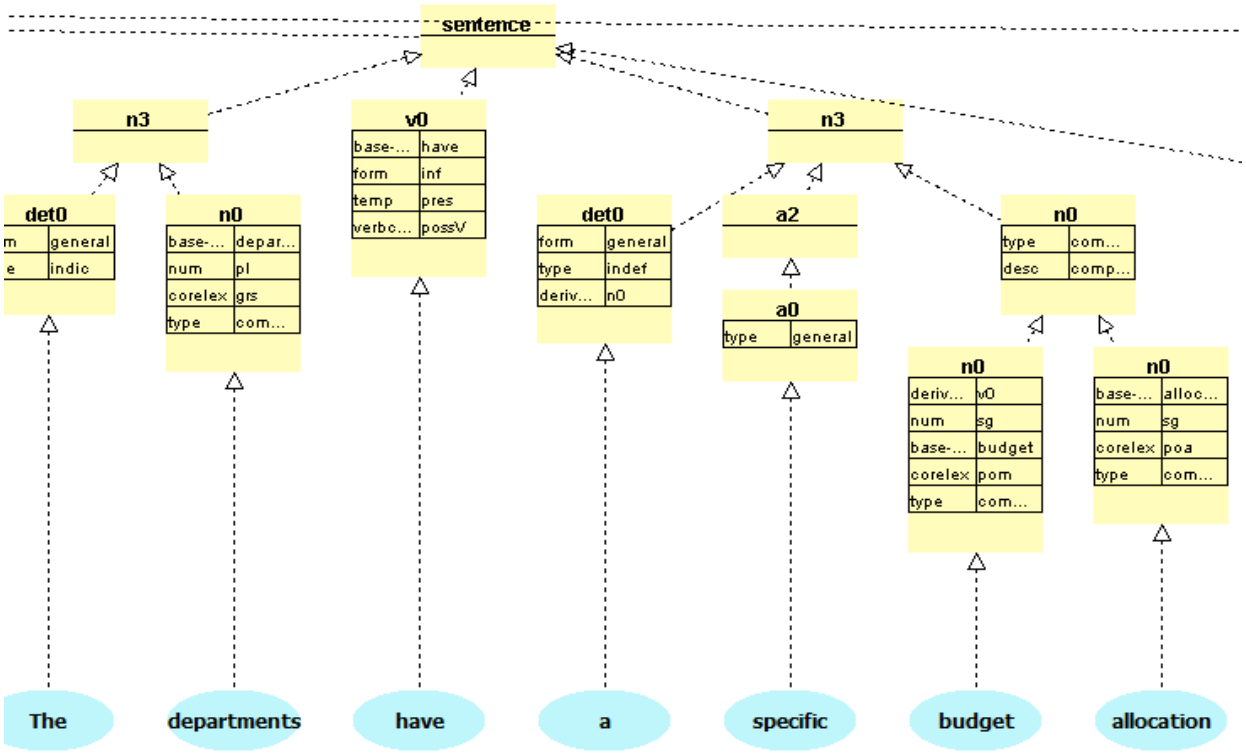

Fig. 1. Tagged sentence with chunk tree

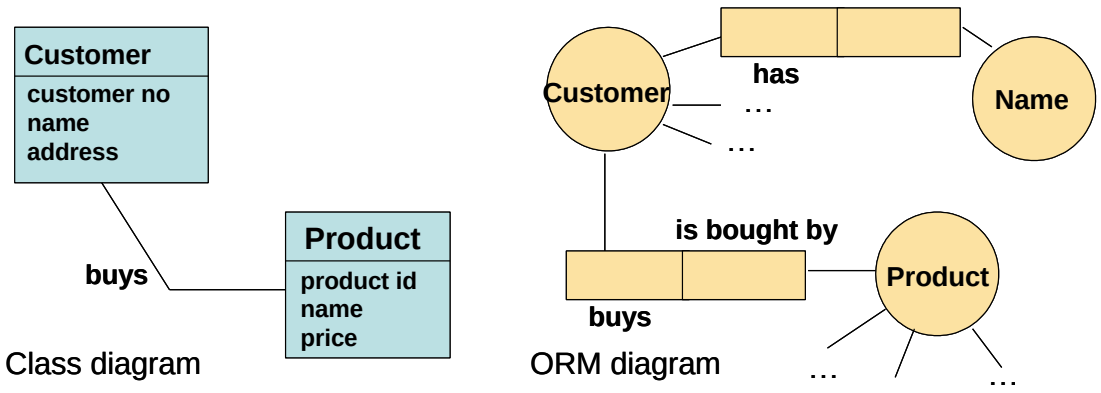

Fig. 2. Class diagram versus ORM diagram

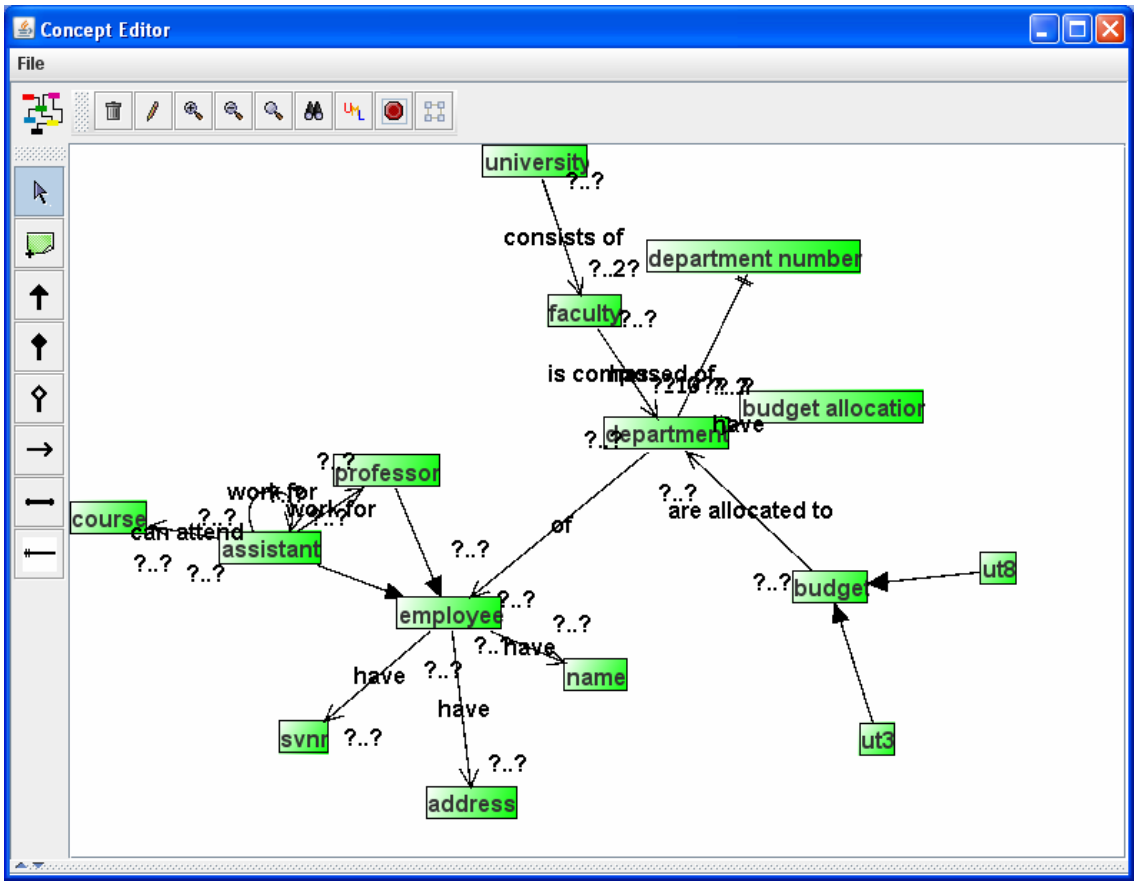

Fig. 3. Graphical representation of the interlingua (university example)

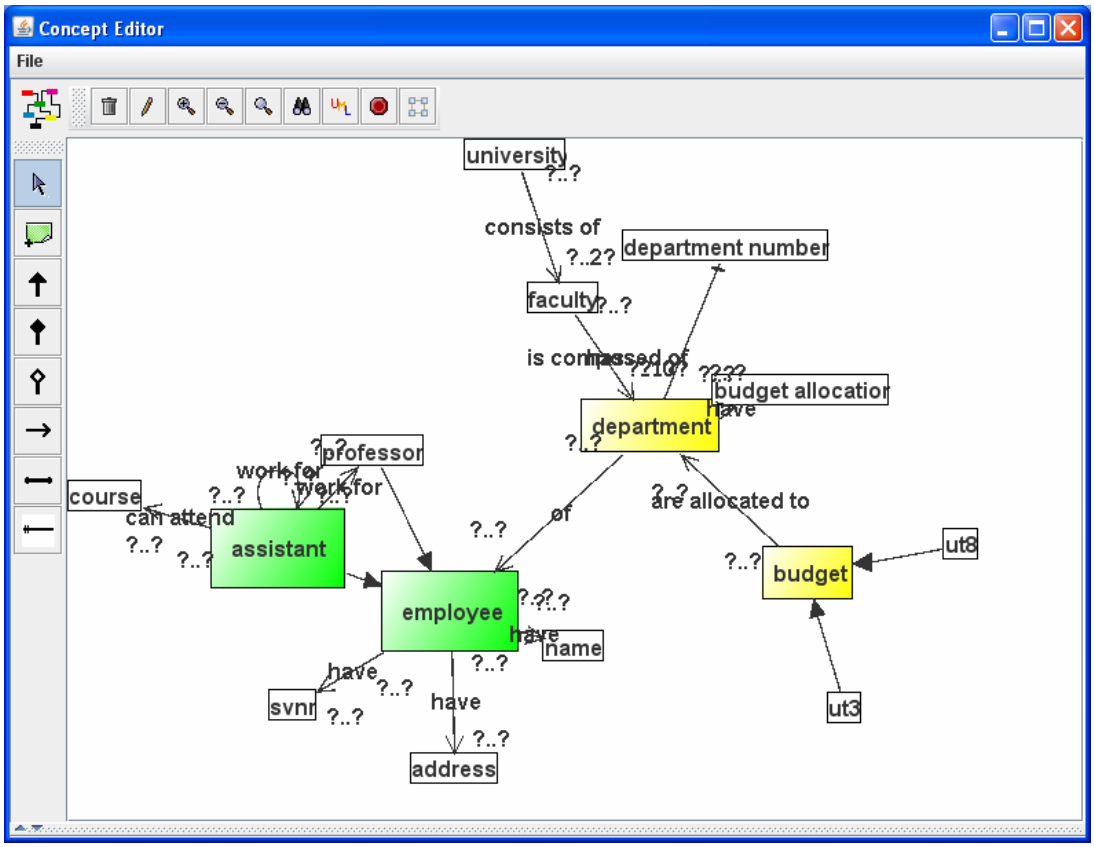

Fig. 4. Visualization of centered thing types

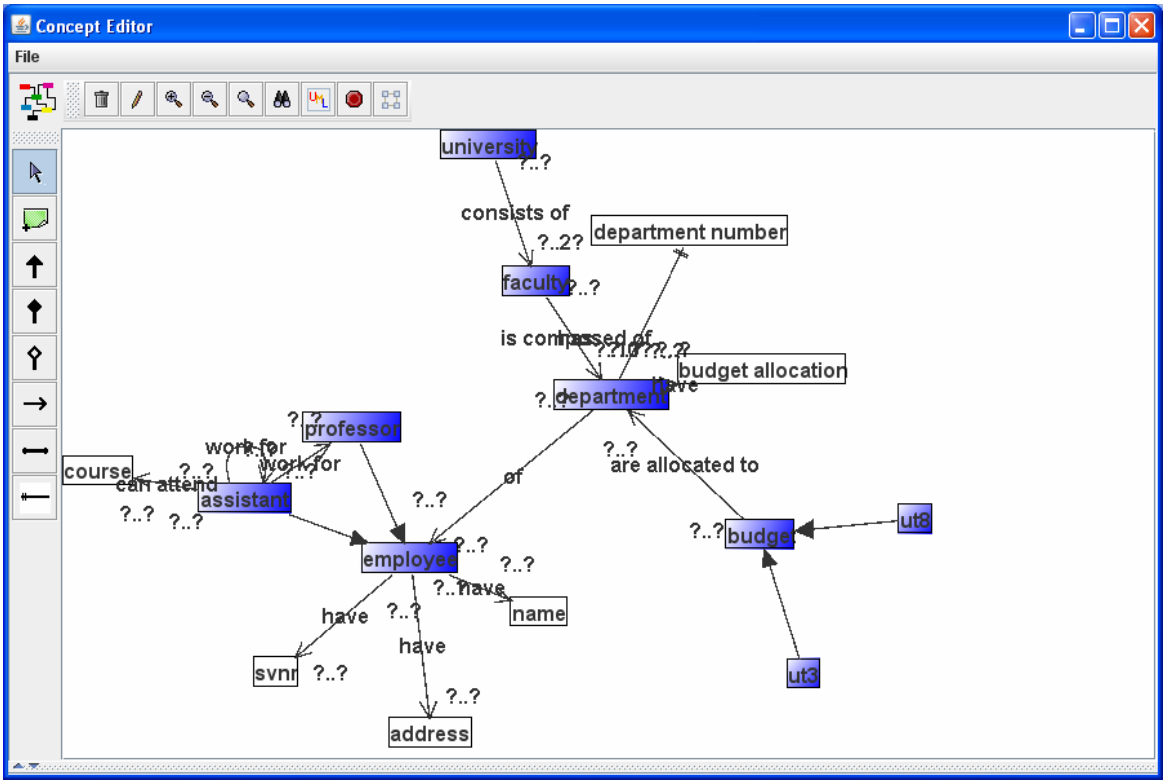

Fig. 5. Mapping preview