LINGUISTICALLY BASED REQUIREMENTS ENGINEERING – THE NIBA-PROJECT

Günther Fliedl[1], Christian Kop[2], Heinrich C. Mayr[2], Willi Mayerthaler[1], Christian Winkler[1]

1. Introduction

The analysis of requirement specifications in the context of information systems development is of growing importance in the field of computational linguistics.

Usually, the development of an information system (or some part of it) starts with an requirements elicitation, collection and analysis phase that results in a set of natural language requirements specifications. These then serve as a source for the phase of conceptual design where a semantic model of the given universe of discourse is established using modeling and representation concepts like UML [1] or former approaches like OMT [6] or even the classical Entity-Relationship Model (or one of its numerous extensions). These approaches mainly are thought for enhancing the validation of the collected requirements as well as for providing a more formal basis for the subsequent system design and realization steps. Indeed, they form a considerable help for the designer, however they are no aid in collecting and cataloguing information. Moreover, these approaches require a level of abstraction end-users (i.e., the 'requirement suppliers') often are not willing or not able to cope with.

To overcome these disadvantages the Klagenfurt Conceptual Predesign Model (KCPM) [3] was developed to harmonize the developer's and the end-user's view of a given universe of discourse (UoD), i.e., to provide an interface for their mutual understanding. The intention is to support the idea of collecting and cataloguing UoD informations in an non-textual form immediately after a linguistic analysis of the input texts but without formalizing them corresesonding to conventional conceptual modeling. Linguistic analysis is done using the model of NTMS, which stands for

[1] Institute for Linguistics and Computational Linguistics, University of Klagenfurt
[2] Institute for Business Informatics and Information Systems, University of Klagenfurt

‘Natürlichkeitstheoretische Morphosyntax’ (Naturalness Theoretical MorphoSyntax). NTMS is a generative orientated grammar model that uses categorial specific phrase structures in the description of grammatical phenomena. In what follows we outline the main KCPM concepts first and present the steps of the linguistic analysis next.

2. KCPM

In order to provide for a language that enhances the communication between developers and end-users we restricted ourselves to a small number of modeling concepts (notions), namely *thing type, connection type*, *operation-type*, *event type* and *constraint*.

The notions thing type and connection type are to cover static UoD aspects: Consider *thing type* as a generalization of the well-known conceptual notions *entity-type resp. class* and *attribute resp. value type*. This, thing type covers general concepts like natural or juridical persons (e.g. customer, employee), material or immaterial objects (e.g. product, vehicle, account, order), features (e.g. customer name, article number, color, weight).

As a consequence of not distinguishing between classes and value types, relationships between those are not distinguished as well (e.g. into associations and attributes): KCPM proposes the concept *connection type* for all kinds of relationships including the abstractions aggregation and generalization. Think, e.g., of sentences like

(1) *A customer can buy more than one product.* (*Ein Kunde kann mehrere Produkte kaufen.*)
(2) *A product is sold to exactly one customer.* (*Ein Produkt wird an genau einen Kunden verkauft.*)
(3) *A customer has up to 4 phone numbers. (Ein Kunde hat bis zu 4 Telefonnummern.)*
(4) *A phone number belongs to one customer. (Eine Telefonnummer gehört zu einem 1 Kunden.)*

which relate several thing types, here *customer* with *product* and *customer* with *phone number*. Note that this approach corresponds to the NIAM Object-Role Model [5].

In order to capture dynamic UoD aspects, we provide the concepts *operation type* and *event type*. An *operation type* describes the activities the instances of a thing type are able to perform, the resources needed for performing an activity as well as the duration of activity execution and the period within which an activity has to be performed. Operations may be aggregated to complex ones by composition and sequencing. An operation type (or some aggregate) Pre- and Postconditions for the execution of one or more (aggregated) operations together with that operations form an *event type*.

All other UoD aspects that are not to be captured by the before-mentioned KCPM concepts are called *constraints*.

KCPM does not enforce a specific representation (notation), however, we propose to use glossaries thus underlining the idea of using a KCPM-Schema as a kind of notepad during requirements collection. This schema then also may be understood as a dictionary for the given application domain.

To sum up, our approach is called ‘conceptual’ since it uses semantic modeling concepts; it is called ‘predesign’ since it forms a very early modeling step within the IS design process. Of course, using KCPM is very time consuming if there are neither tools supporting the collection and classification of linguistic objects nor mechanisms for automatizing the transformation of a KCPM-Schema into a conceptual one (e.g. a UML model). Therefore, within the NIBA-Project[3], we aim at providing for such tools and the related theory.

3. Linguistic support for the extraction of KCPM notions

Knowledge processing as required in conceptual predesign needs a highly efficient computational linguistic model for the analysis of language. The aim of the NTMS approach is to comply with those requirements with regard to praxis oriented computational linguistic work. In addition, NTMS provides for the following features:

- lexicon entries are classified by features and organized correspondingly to application domains,
- semantic rules,
- grammatical values (e.g. subject function) and sentence context (e.g. argument structure) are determining parameters; these are made explicit according to naturalness theory.

[3] NIBA is a german acronym for natural language requirement analysis. The project is sponsored by the Klaus Tschira Stiftung Heidelberg.

Computational grammar is considered as the *missing link* between the UoD in question and the schemes applied/defined using KCPM. In other words, the semantically enriched syntactic structures can be translated directly into the KCPM modeling concepts, the input being any German sentence related to the chosen segment of reality. The analysis contains the following main steps:

1. the words of a given text are compiled into a lexicon which implies automatic categorial, semantic and contextual specification of words,
2. word- and morphosyntactically interpreted microstructures are assigned to wordgroups and words, e.g., the automatic splitting of the german verbal noun *Übersetzer,* [2]
3. analysis of (sentence)syntax according to the specific X'-mechanism of NTMS [4]. This generates binary branching trees or labeled bracketings with enriched semantic features for phrases.
4. bracketing-output is listed and compared; the analysed sentences are listed and sorted with respect to their textual information and position in the text produced by the end-user.

The bracketing-output contains the semantic features of the analysis step and may be used for the interpretation of the sentence itself and for the extraction of KCPM concepts. Generally speaking, in this step we extract thing types from noun phrases (without the need to distinguish whether the noun phrase is a class or an attribute). From verb-phrases we extract connection types or operations respectively. We also extract connection-types in syntactic relations within noun-phrases. The semantic roles (a specific generated semantic feature) a noun phrase has in the verb-phrase indicates it's specific function in an operation and a connection type.

4. Conclusion

The NTMS as a computational grammar is considered as the 'missing link' between an UoD and the schemes resulting from the use of KCPM. We gave a short sketch on how semantically enriched syntactic structures might be translated into KCPM modeling concepts, the input being any German sentence related to the chosen segments of reality. For this purpose, the process of requirements translation has been divided

- into the NTMS-based semanto-syntactic analysis of natural language requirements specifications and
- the extraction of discourse semantics into a UoD glossary. We propose that these two subtasks are part of the first step of conceptual modeling which we call conceptual predesign.

The next steps of our work aim at the extraction of dynamic UoD aspects and their mapping to KCPM concepts for dynamics and the before-mentioned transformation of KCPM schemes into UML models.

5. References

[1] Booch, G.; Rumbaugh, J.; Jacobson, I.: The Unified Modeling Language User Guide, Addison Wesley, 1999.

[2] Fliedl, G.: Natürlichkeitstheoretische Morphosyntax, Aspekte der Theorie und Implementierung", Habilitationsschrift, Gunter Narr Verlag Tübingen, 1999.

[3] Kop, C.; Mayr, H.C.: "Conceptual Predesign – Bridging the Gap between Requierements analysis and conceptual design". To appear in: Proc. ICRE'98, International Conference on Requirements Analysis, Colorado Springs, April 1998.

[4] Mayerthaler, W.; Fliedl, G.; Winkler, Ch.: Lexikon der Natürlichkeitstheoretischen Syntax und Morphosyntax" Stauffenburg Verlag, Tübingen, 1998.

[5] Nijssen, G. M.; Halpin, T. A.: Conceptual Schema and Relational Database Design – A fact oriented approach" Prentice Hall Publ. Comp., 1989.

[6] Rumbaugh, J.; Blaha, M.; Premelani, W.; Eddy, F.; Lorensen, W.: Object Oriented Modeling and Design, Englewood Cliffs, NJ, Prentice Hall, 1991.