An Approach to Building Domain Models Interactively

Helmut Horacek

Universität des Saarlandes
FB 14 Informatik
Postfach 1150, D-66041 Saarbrücken, Germany
email: horacek@cs.uni-sb.de

Abstract. Building formal domain models on the basis of natural language specifications is an ambitious task that has a high application potential. However, existing methods are limited by several kinds of conceptual shortcomings, one of them being the strong ontological correspondence between specified natural language expressions and their domain model counterparts. Aiming at an increased flexibility in this respect, we propose an interactive exploration of domain model variants in which modifications are initiated by user requests. Ingredients of our approach are an intermediate representation which covers specificities of language and domain model representations, and operations that introduce stepwise modifications of domain model variants. Our method constitutes a first step towards an interactive, incremental, and interest-driven approach to building domain models that are less tightly related to the specified natural language expressions.

1 Introduction

Building formal domain models on the basis of natural language specifications is an ambitious task that has a high application potential. So far, this aim has been pursued for database models and for some aspects of requirements engineering. However, existing methods are limited in several respects, one of them being the strong ontological correspondence between specified natural language expressions and their domain model counterparts.

Aiming at an increased flexibility in this respect, we propose an interactive exploration of domain model variants in which modifications are initiated by user requests. Ingredients of our approach are an intermediate representation which covers specificities of language and domain representations, and operations for stepwise modifications of domain model variants. In the current state of work, these models may differ from given natural language specifications in certain degrees of explicitness and granularity, and the model building constructs are restricted to those for relational databases; nevertheless, we believe that our techniques can be extended to other variations and representation elements. Our approach constitutes a first step towards interactive, incremental, and interest-driven methods for building domain models that are less tightly related to the specified natural language expressions.

This paper is organized as follows. We review previous work on building domain models from natural language specifications and work in natural language processing that can handle some sorts of variations in natural language expressions. Then we introduce our techniques for capturing domain model variations that increasingly abstract from natural language formulations. We follow by describing operations that induce modifications triggered by user requests. We conclude with a short outlook for extensions.

2 Motivation

Building domain models, primarily database models, on the basis of natural language specifications has been pursued for many years, including automated database design [SG93], semi-automated generation of entity relationship models [GSD99] and linguistically supported tools for requirements engineering [Bu96]. The general approach is that semantic analysis yields logical forms or equivalent structures built out of every sentence in the natural language specifications, and that these structures are subsequently transduced into constructs in the target modeling language. This process may be supported by linguistic and domain knowledge of varying depth and accuracy, ranging from elaborate linguistic theories and large-scale knowledge sources [BR96] to domain-specific preferences for interpreting ambiguous verbs [MG00]. Despite the considerable effort in using linguistic knowledge beneficially, existing systems have a characteristic property which is one of the reasons for their limited application potential: The elements in the domain models built widely reflect the formulations of the natural language specifications, and they are composed by simply adjoining representations built from individual sentences without considering contextuality and intended use. This property causes fundamental deficits for building domain models:

- Particularities of the purpose of the target system are not reflected – be it storing specific entities and relations abstracted from some segment of the real world, as in databases, or different states of a system, the current and the intended state, and the hereby role of a program to be built, as in requirements engineering.
- Natural languages cannot express everything in exactly the way needed for some formal purpose; sometimes, suitable natural language expressions are more detailed than the corresponding formal constructs, but these expressions can also appear in a form that is less explicit than an appropriate form needed for the formal system.
- Even for cases where expressing the intended functionality in an exact way in the target language is possible, relying on accurate natural language descriptions would assume humans to be capable of expressing themselves perfectly.

These observations illustrate that building domain models is more a design issue than a translation task. Hence, an automated approach should be organized as an incremental, refinement-based process rather than as a one-shot activity, thereby exploring potential alternatives in accordance with methodologies for building entity-realtionship diagrams by hand.

3 Related Work

The task of building a database model from natural language specifications bears some similarities to the task of interfacing a database with natural language. In each case, portions of a database structure are referred to by natural language expressions. In the interfacing task, however, these expressions are mapped onto a segment of a specific database model, while the task of building such a model on the basis of natural language specifications in some sense presupposes a mapping onto all possible database models and choosing the most appropriate one. According to the currently dominating strategy, this choice is in some

sense circumvented by building a model that corresponds most directly to the formulation in the given text. As opposed to this strategy, the interfacing task forces a system to bridge differences between natural language formulations and the data structure referred to by them, which existing systems are not able to do perfectly. Some ambitious systems are TEAM [Gr87], DATENBANK-DIALOG [Tr87] and DELPHI [BIS91, St93], the latter reflecting empirical evidence about database inquiries in natural language [Ma92].

One issue addressed by all these systems is the occurrence of nominals that are forced to take a meaning that differs from the immediately apparent one, but is somehow related to it. This phenomenon is commonly known as metonymy and has been treated intensively in the computational linguistics literature (e.g., [Nu95]). Typically, some sort of term expansion is done by which the implicitly expressed relation is made explicit. Thus, this phenomenon constitutes one case where reference to a domain model requires more details than those provided by the corresponding natural language expressions. As opposed to that, there are compact domain model components that conflate several facts whose description requires the combination of some phrases. An example for that are the properties of being a medical doctor and being on board of a ship, which are condensed into a logical attribute in a military database (see [Gr87]). Apparently, this compactification can only be justified by the perspective adopted for meeting the interest underlying that specific database.

In order to solve similar problems in the context of machine translation, one principled approach is to design an intermediate representation onto which source language expressions are mapped and from which appropriate expressions in the target language are built. The intermediate representation is 'language-neutral' in the sense of Dorr [Do93], and it needs to cover all distinctions needed for at least one of the languages addressed. These are exactly the properties which we consider relevant for a representation structure mediating between natural language specifications and domain models. The main difference is that the repertoire of mapping operations is different, since it does not mediate between several natural languages, but between one natural language and a domain model language.

4 A Method for Handling Domain Model Variations

In order to envision more flexibility in building domain models, we propose to adopt techniques from machine translation as mentioned in the previous section. Adoption for our purposes means that the intermediate representation must be explicit in all aspects covering distinctions in the natural language in which specifications are formulated as well as in all aspects covering distinctions in the domain model variants to be built. Moreover, transitions between elements of the domain model and corresponding elements of the intermediate representation must be captured in the same way as the elements of the natural language representation are treated. Consequently, the repertoire of mapping operations must be modified and enhanced according to the structures appearing in the domain model and their correspondences with intermediate representation elements. This modification is the most difficult part, and shortcomings in the coverage of mapping operations cause limitations in the way a domain model can be designed automatically. We will make use of the mapping operations in [Ho96], which allow bi-directional mappings between natural language and conceptual representations. The mapping operations provide a considerable degree of para-

prasing capabilities. *Mapping schemata* express local correspondences across representation levels. They define equivalences of the information content associated with individual elements of the target representations – natural language and domain model – and corresponding constructs of the intermediate representation, which may consist of a chunk of elements for target representation elements associated with rich semantics.

There are two classes of schemata, *ZOOM* schemata and *SUBSTITUTION* schemata. In each case, the semantics associated with one element of the natural language side (a lexeme, a grammatical function, a feature or a feature value) or the domain model side (an entity, a relation, an attibute, or an attribute value) is expressed by a chain of representation elements on the intermediate representation level, that is, concept and role nodes and their connecting links in a KL-ONE [BS85] representation. The composition of elements on each representation level is handled according to coherence criteria on the respective level. On the language level, these operations are regulated by an interplay of lexical and grammatical constraints. On the intermediate representation level, the relation and entity nodes must appropriately connected by links, with value fillers of relations satisfying class restrictions. Similarly, the domain model level has its specific constraints. For example, attributes in a database model must not bear information other than their values, which must be atomic.

ZOOM schemata serve the purpose of bridging differences in granularity, by relating a lexical predicate to a chunk of elements on the intermediate representation level; alternative correspondences can be established by implementing results from lexical semantics and insights originating from lexicography, in the degree of accuracy needed for the application at hand. There are four subclasses of *ZOOM* schemata: *MICRO*, *STANDARD*, *MIX* and *MACRO ZOOM* schemata. The *MICRO ZOOM* schema, in turn, has two subclasses: *MICRO 1* and *MICRO-2*. A *MICRO-1 ZOOM* schema maps a lexeme or a feature value onto a node in the intermediate representation level (a *concept* or a *role* node). For example, the verb 'owning' may be expressed by an OWNING concept. The complementary *MICRO-2 ZOOM* schema maps an auxiliary verb or a grammatical function onto a *link* connecting a concept and a role node. Unlike the basic *MICRO ZOOM* schema, which describes simple one-to-one mappings, other schemata refer to non-atomic structures on the intermediate representation level. The *STANDARD ZOOM* schema maps a grammatical function onto a role node and its adjacent links. For example, the grammatical functions SUBJECT and OBJECT of the verb 'owning' are mapped onto the roles Agent and Theme, including both links connecting these roles. The *MIX ZOOM* schema is dedicated to handling role nouns, that is, nouns derived from an action. It maps such a noun onto a concept, a role, and the connecting link. For example, the noun 'owner' is mapped onto a conceptual chain consisting of the concept OWNING, the role Agent, and their connecting link. Finally, there is the schema with the largest coverage, the *MACRO ZOOM* schema. It covers a concept node, two role nodes linked to it, and all links attached to the two role nodes. For example, a *MACRO* schema maps the genitive case relating an owner to the object owned by him onto the concept OWNING, the roles Agent and Theme attached to that concept, and all links of the roles. Through composing *ZOOM* schemata, which must yield a connected structure out of individual results without gaps and overlaps, alternative ways of expressing relations in natural language can be mapped onto a uniform intermediate representation. For example, phrasings such as 'A owns B' or 'A is the owner of B', or a possessive genitive relating A to B are mapped onto an OWNING concept with uniform connections to the owner and the thing owned.

The intermediate representation built from these expressions is more detailed than the version with the largest number of words. If paraphrased in all detail, which would be quite unnaturally, this would yield 'There is an ownership relation. It has an agent, which is A, and a theme, which is B'. The compositions of mapping schemata that are instantiated to build this structure from three alternative expressions are illustrated in Fig. 1. For each variant, the relevant fragment of the intermediate representation is depicted, with concept nodes shown as ovals and role nodes as squares. Since the concept OWNING bears the roles Agent and Theme, the connecting links are directed towards their fillers. These fillers, which in turn may bear structures representing the descriptions conveyed by the noun phrases A and B, respectively, are not shown in the Figure. Below each copy of the intermediate representation fragment are the versions of natural language expressions related to it. The dotted arrows that link elements of the intermediate representation to lexical items indicate the coverage of each schema in the intermediate representation, with schema names in between. Genitive cases are mapped context-dependently, by a *MACRO ZOOM* schema (when related to a entity) and by a *STANDARD ZOOM* schema (when related to the owning relation).

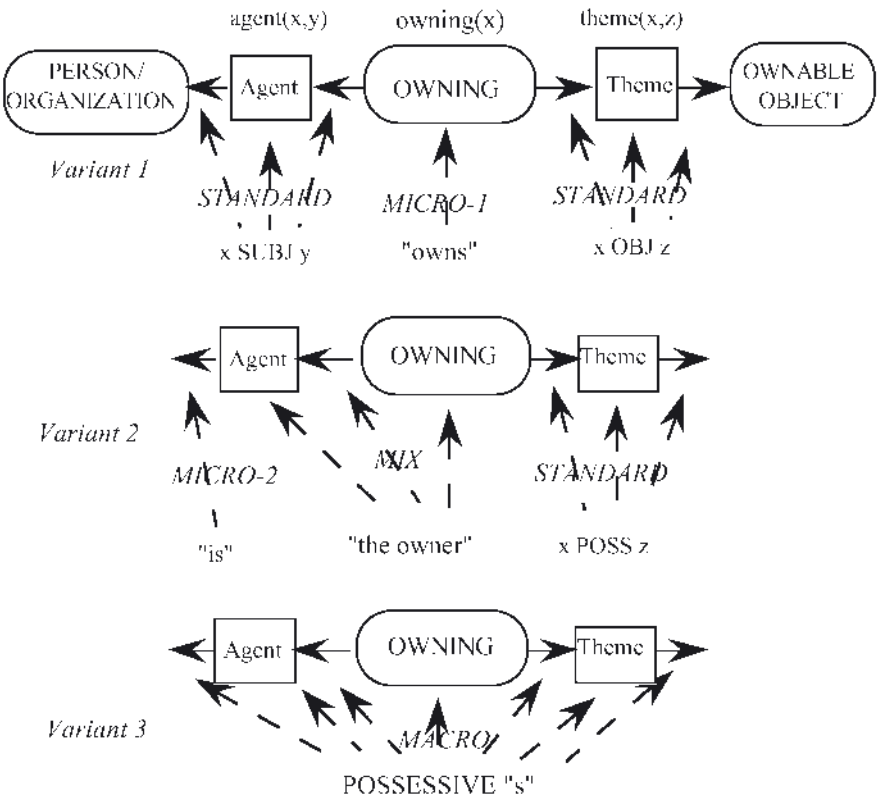

Fig. 1: Compositions of *ZOOM* schemata for mapping alternative natural language expressions

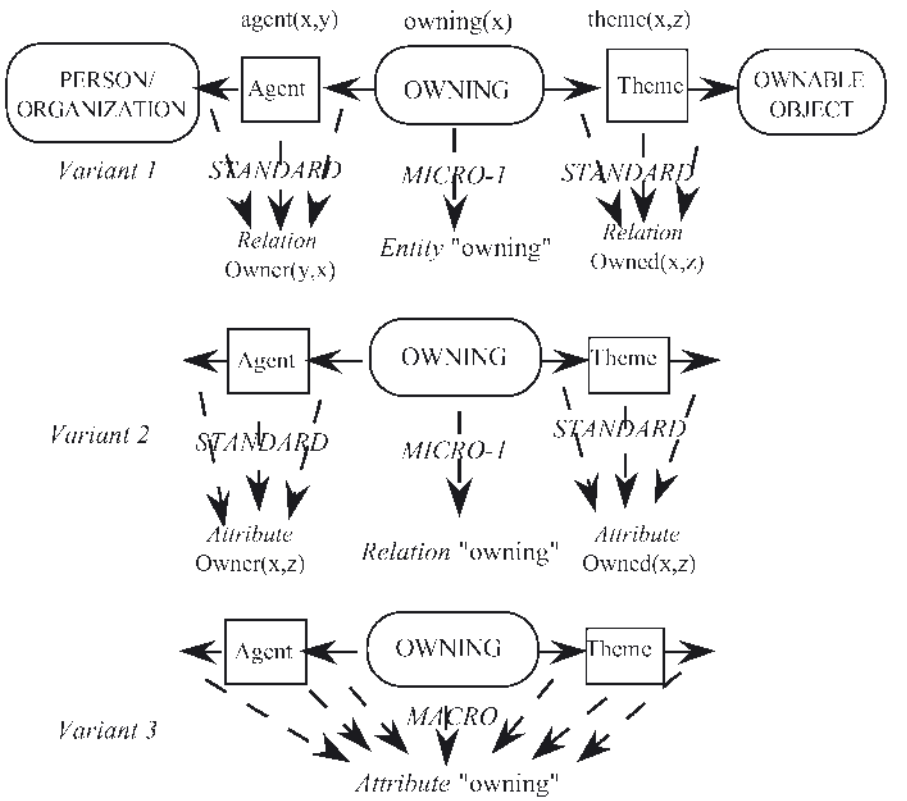

Fig. 2: *ZOOM* schemata used for mapping onto domain model elements

Similarly to the mappings from natural language expressions onto intermediate representation constructs, there exist alternatives on the side of the domain model. An ownership may be merely modeled as an attribute if just a few distinctions, such as the kind of owner (e.g., private or government) are of interest for the application at hand. If more details about both, the owner and the thing owned by him, are of interest, an ownership may be modeled as a relation between these two entities. Finally, an ownership may be modeled as an entity of its own right, if properties attributed to it, such as a time interval expressing its duration, are of relevance for the application. These alternatives are illustrated in Fig. 2, which are structurally identical to two of the variants shown in Fig. 1, except to the direction of mapping. To express ownership as an entity or as a relation, a *MICRO ZOOM* schema is used, and a *MACRO ZOOM* schema handles the attribute variant. The choice between these alternatives is contextually justified in favor of the entity variant, if further attributes of ownership are modeled. In addition, Fig. 2 depicts suitable mappings for the adjacent intermediate representation elements (attributes for the relation variant, relations for the entity variant).

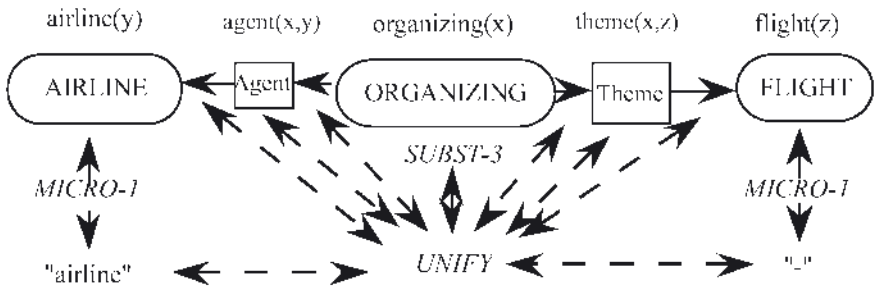

Fig. 3: An example of a *SUBSTITUTION* schema

The other class of schemata, *SUBSTITUTION* schemata, serve the purpose of bridging differences in degrees of explicitness, thereby relating pieces of information expressed implicitly on the lexical level to their corresponding images on the intermediate representation level. Similarly, representations on the domain model side may not be needed in the same degree of explicitness as the representations on intermediate representation level. The functionality of a *SUBSTITUTION* schema is to conflate the images of related objects on the intermediate representation level when mapped onto the natural language domain model levels. The *SUBSTITUTION* schema must cover the connection between the two objects conflated. Moreover, one of the objects must be mapped onto a 'semantically empty' object for the less explicit representation variant. Conversely, when the mapping direction is inverted, a lexical item is represented as a structure consisting of an object expressing its proper semantics, a suitable 'semantically empty' object. Through an appropriate *SUBSTITUTION* schema these two objects are connected by the conceptual part covered by that schema, which leads to the insertion of a chain of elements between the two entities, expressing the relation left implicit on the other representation level. In order for this expansion to work, two conditions must hold: there must be some reason for applying this operation, such as a type incompatibility that can potentially be resolved this way, and there exists a suitable *SUBSTITUTION* schema that bridges this discrepancy. Conversely, that chain is left implicit when mapping from the intermediate representation, unifying the images of the related entities.

One kind of phenomena treated by *SUBSTITUTION* schemata are metonymic expressions, with an example shown in Fig. 3. It refers to the conceptual image of the phrase 'airline' in the context of the assertion 'airlines connect cities with one another'. Since not the airlines themselves, but the flights organized by them connect cities, these two kinds of objects are distinguished in the intermediate representation, linked through an appropriate schema (*SUBST-3* in Fig. 3) which is justified by lexical semantics. The lexicon entries relevant for this example, as expressed in terms of the Generative Lexicon [Pu91], are shown in Fig. 4 – they are motivated by the regular use of companies referring to the events they organize as their usual business. Among the four roles of each lexeme, the TELIC and the AGENTIVE roles are the important ones for our purposes, since they represent the relevant information for term expansion. These are the events that bring something into being (the AGENTIVE role), and that indicate what a thing is made for (the TELIC role) – thereby, *or* stands for an

airline(x)		flight(x)	
CONST	{planes, office, .. }	CONST	{place, source, ... }
FORMAL	organization(x)	FORMAL	location-change(x)
TELIC	organize(e^{τ},y,x)	TELIC	carry(e^{τ},y,x)
AGENTIVE	found(e'^{τ},z,x)	AGENTIVE	organize(e'^{τ},z,x)

Fig. 4: Qualia Structures for nouns for justifying a SUBSTITUTION schema

event variable of type transition (τ). For example, airlines have the purpose of organizing something (that is, flights), which gives rise to building an appropriate schema.

5 Interactive Operations

A repertoire of schemata covering texts in the intended domain of application enables us to map natural language descriptions onto a uniform intermediate representation, as well as to build limited variations of domain model fragments. Since differences in the natural language formulations indicate no preferences for choosing among domain model variants, the intermediate representation is first mapped onto a domain model representation that most closely corresponds to the natural language formulation. Some operations (currenty two) are provided by which a user can express ways to modify the current representation variant:

- An object can be indicated as being not important to be modeled in its own right, or the explicit introduction of an object instead of a simple value can be demanded.
- A relation can be indicated as being represented too implicitly or too explicitly.

The first option is applicable to switch between varying degrees of granularity, alternating different compositions of *ZOOM* schemata. The second option is applicable to switch between varying degrees of explicitness, which means using an alternative to a *SUBSTITUTION* schema or using such a schema instead of the ones that are actually used.

Let us consider a short example. When the natural language specification 'airlines serve food between one city and another' (accomodated and abstracted from the database request 'Which airlines serve diet food from New York to Boston?', as discussed in [St93]) is analysed, it is transduced literally to the intermediate representation level and expressed in the most compact domain model variant (an entity for 'airline' with attributes 'serve', 'from' and 'to'). This domain model excerpt corresponds to the logical form (1) in Figure 5. Since linking source and goal cities directly to the airlines is odd, a reasonable user request would be to make the attribute Serve more explicit. The demand induces two subsequent metonymic extensions. The first one is based on the lexicon entry for 'airline' (see Fig. 4), which causes the insertion of FLIGHTS linked to AIRLINE via an ORGANIZE relation. The second extension is based on the lexicon entry for 'flight' (see Fig. 4) and leads to the insertion of PERSON related to FLIGHT via a CARRY relation. After suitable renaming of variables, it yields expression (3) in Figure 5. However, for a database about flight connections, this version is too explicit, since airline employees need not to be modeled for this purpose.

(1)
```
(SOME x AIRLINE
   (AND (SERVE x DIET-FOOD)
        (SOME y CITY (SOURCE x y))
        (SOME z CITY (GOAL x z))))
```

(2)
```
(SOME x AIRLINE
   (MULTIPLE y FLIGHT
   (AND (ORGANIZE x y)
        (SOME u CITY (SOURCE y u))
        (SOME v CITY (GOAL y v))
        (SERVE y DIET-FOOD))))
```

(3)
```
(SOME x AIRLINE
   (MULTIPLE y FLIGHT
   (AND (ORGANIZE x y)
        (SOME u CITY (SOURCE y u))
        (SOME v CITY (GOAL y v))
        (MULTIPLE z PERSON
        (AND (CARRY y z)
             (SERVE z
                DIET-FOOD))))))
```

Fig. 5: Representation variants for the sentence 'Airlines serve diet food between two cities'

Hence, indicating the relation CARRY as too explicit leads to a contraction operation, which yields expression (2) in Figure 5, the best domain model representation for the given text.

In order to approach relevance for practical application, these mapping techniques need to be extended substantially. This concerns at least the following three aspects:

- Enriching the repertoire of representation elements. In addition to concepts, roles, and the links between them, also generalisation and cardinality information is of relevance.
- Purpose-driven choice among domain model variants. For example, preferring to express an m:n relation by an entity in its own right, which relies on cardinality information.
- Processing strategy. For larger natural language descriptions, it seems to be advisable to complement interactivity by incrementality, that is, building domain model representations is not done in one-shot but in portions, interleaved with exploring the variants.

In that setting, user requests could be processed by providing hints or warnings if the modfications required lead to non-preferred variants or require provision of further specifications.

6 Conclusion

In this paper, we have argued in favor of an interactive exploration of domain model variants in which modifications are initiated by user requests. The main emphasis in our approach lies in decoupling natural language formulations and domain model representations. Ingredients of our method are an intermediate representation which covers specificities of language and domain model representations, and operations that introduce stepwise modifications of domain model variants. These techniques constitute a first step towards interactive, incremental, and interest-driven methods for building domain models that are less tightly related to the specified natural language expressions. The strongest requirements for setting up a system on that basis lies in the availability of lexical semantic representations, which must be built for a domain of interest. Concerning processing efficiency, experience from generation tells us that the composition of several schemata is reasonably fast due to constraint

propagation techniques. In this respect, we are better off than matching based on Meaning-Text Theory [MP87], which provides an even richer repertoire of semantic definitions for relating formulation variations to underlying conceptual representations, but Meaning-Text Theory even more suffers from the need to build elaborate and large dictionaries. In the current state of work, domain models may differ from given natural language specifications in certain degrees of explicitness and granularity, and the model building constructs are restricted to those for relational databases; nevertheless, we believe that our techniques can be extended to other kinds of variations and representation elements.

References

[BIS91] Bobrow, R.; Ingria, R.; Stallard, D. The Mapping Unit Approach to Subcategorization. In Proc. of Speech and Natural Language Workshop. 1991.

[Bu96] Burg, J. Linguistic Instruments in Requirements Engineering. PhD Thesis, Vrije Universiteit Amsterdam, 1996.

[BR96] Burg, J.; van de Riet, R. Color-x: Using Knowledge from Wordnet for Conceptual Modeling. In (Fellbaum, C., ed.), WordNet: An Electronic Reference System and Some of its Applications. MIT Press, Cambridge, MA, 1996.

[BS85] Brachman, R; Schmolze, J. An Overview of the KL-ONE Knowledge Representation System. *Cognitive Science* 9, 1985; pp. 171-216.

[Do93] Dorr, B. Parameterized Machine Translation. *Artificial Intelligence* 6, 1993; pp. 193-223.

[GSD99] Gómez, F.; Segami, C; Delaune, C. A System for the Semi-Automatic Generation of E-R Models from Natural Language Specifications. *Data & Knowledge Engineering* 29, 1999; pp. 56-81.

[Gr87] Grosz, B.; Appelt, D.; Martin, P.; Pereira, F. TEAM: An Experiment in the Design of Transportable Natural-Language Interfaces. *Artificial Intelligence* 32, 1987; pp. 173-243.

[Ho96] Horacek, H. On Expressing Metonymic Relations in Multiple Languages. *Machine Translation* 11, 1996; pp. 109-158.

[Ma92] MACDOW Committee. Multi-Site Data Collection for a Spoken Language Corpus. In Proc. of Speech and Natural Language Workshop, 1992.

[MG00] Martinez, P.; Garcia-Serrano, A. On the Automatization of Database Conceptual Modelling Through Linguistic Engineering. NLDB'2000, Versaillles, 2000.

[MP87] Mel'cuk, I.; Polguère, A. A Formal Lexicon in the Meaning-Text Theory (or How to Do Lexica with Words). *Computational Linguistics* 13, 1987; pp. 261-275.

[Nu95] Nunberg, G. Transfers of Meaning. *Journal of Semantics* 12, Oxford University Press, 1995; pp. 109-132.

[Pu91] Pustejovsky, J. The Generative Lexicon. *Computational Linguistics* 17, 1991; pp. 409-441.

[St93] David Stallard. Two Kinds of Metonymy. In Proc. of the 31st Annual Meeting of the Association for Computational Linguistics, 1993; pp. 87-94.

[SG93] Storey, V.; Goldstein, R. Knowledge-Based Approaches to Database Design. *Management Information Systems Quaterly* 17(1), 1993; pp. 25-46.

[Tr87] Trost, H.; Buchberger, E.; Heinz, W.; Hörtnagl, C.; Matiasek, J. DATENBANK-DIALOG: A German Language Interface for Relational Databases. *Applied Artificial Intelligence* 1, 1987; pp. 181-203.