


# An Integrated Model of Semantic and Conceptual Interpretation from Dependency Structures

Udo Hahn Martin Romacker

Text Understanding Lab,  Group,  
Freiburg University, D-79085 Freiburg, Germany  
<http://www.coling.uni-freiburg.de/>

## Abstract

We propose a two-layered model for computing semantic and conceptual interpretations from dependency structures. Abstract interpretation schemata generate semantic interpretations of ‘minimal’ dependency subgraphs, while production rules whose specification is rooted in ontological categories derive a canonical conceptual interpretation from semantic interpretation structures. Configurational descriptions of dependency graphs increase the linguistic generality of interpretation schemata, while interfacing schemata and productions to lexical and conceptual class hierarchies reduces the amount and complexity of semantic specifications.

## 1 Introduction

The syntax/semantics interface has always been a matter of concern for constituency-based feature grammar theories (cf., e.g., Creary and Pollard (1985), Moore (1989), Dalrymple (1992), Wedekind and Kaplan (1993)). Within the dependency grammar community, far less attention has been paid to this topic. As a consequence, there is no consensus how syntactic dependency structures might be adequately transformed into semantic interpretations (cf., Hajicova (1987), Milward (1992), Lombardo et al. (1998) for alternative proposals).

In this paper, we introduce a two-layered interpretation model. In a first pass, dependency graph structures which result from incremental parsing are immediately submitted to a *semantic interpretation* process. Such a process is triggered by general schemata whenever a semantically interpretable subgraph of a syntactic dependency graph becomes available (cf. Section 3). As a result, lexical items and the dependency relations holding between them are directly mapped to associated conceptual entities and relations at the level of semantic representation (cf. Sections 4 and 5). In a subsequent step, the (quasi-inferential) implications of the knowledge representation structures emerging from the semantic interpretation step are accounted for by a process we here refer to as *conceptual interpretation*. The corresponding operations relate to the concep-

tual representation level only and are triggered by a variety of production rules rooted in ontological categories in order to generate a canonical conceptual representation of the parsed sentence (cf. Section 6). This second level of interpretation is usually not taken into consideration by computational models of semantic interpretation, neither constituency-based nor dependency-based ones, although it turns out to be crucial for natural language *understanding*.

## 2 Grammar and Concept Knowledge

*Grammatical knowledge* for syntactic analysis is based on a fully lexicalized dependency grammar (Hahn et al., 1994). Our preference for dependency structures is motivated, among other things, by the observation that the correspondence of dependency relations (holding between lexical items) to conceptual relations (holding between the concepts they denote) is much closer than for constituency-based grammars (Hajicova, 1987). Hence, a dependency-based approach eases inherently the description of the regularities underlying semantic interpretation.

In this lexicalized dependency framework, lexeme specifications form the leaf nodes of a lexicon DAG, which are further abstracted in terms of lexeme class specifications at different levels of generality (cf. Figure 1). This leads to a lexeme class hierarchy, which consists of lexeme class names  $\mathcal{W} := \{\text{VERBAL, VERBINTRANS, NOMINAL, NOUN, ...}\}$  and a subsumption relation  $isa_{\mathcal{W}} = \{(\text{VERBINTRANS,}$

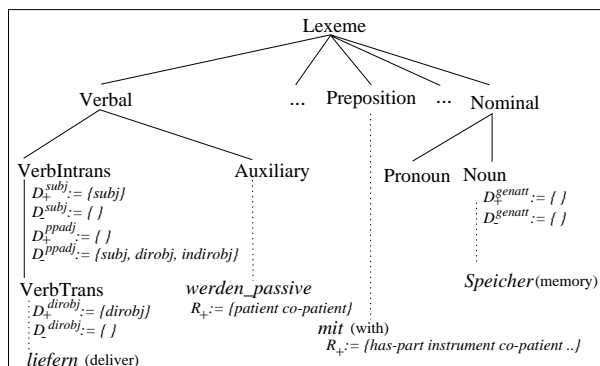


Figure 1: Fragment of the Lexeme Class Hierarchy

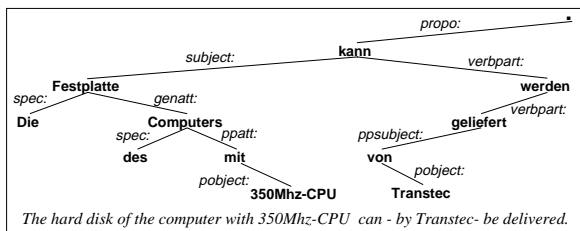


Figure 2: A Sample Dependency Graph

VERBAL), (NOUN, NOMINAL), ...}  $\subset \mathcal{W} \times \mathcal{W}$ . Inheritance of grammar knowledge is based on the idea that constraints are attached to the most general lexeme classes to which they apply, leaving room for more and more specific (possibly, even idiosyncratic) specifications when one descends this hierarchy.

A dependency grammar captures binary constraints between a syntactic head (e.g., a noun) and one of its possible modifiers (e.g., a determiner or an adjective). In order to establish a dependency relation  $\delta \in \mathcal{D} := \{\text{specifier}, \text{subject}, \text{dirobject}, \dots\}$  between a head and a modifier, lexeme-class-specific constraints on word order, compatibility of morphosyntactic features and semantic integrity must be fulfilled. Figure 2 depicts a dependency graph in which word nodes are given in bold face and dependency relations are indicated by labeled edges.

*Conceptual knowledge* of the underlying domain is expressed in terms of a KL-ONE-like knowledge representation language (Woods and Schmolze, 1992). The domain ontology consists of a set of concept names  $\mathcal{F} := \{\text{COMPANY}, \text{HARD-DISK}, \dots\}$  and a subsumption relation  $isa_{\mathcal{F}} = \{(\text{HARD-DISK}, \text{STORAGE-DEVICE}), (\text{TRANSTEC}, \text{COMPANY}), \dots\} \subset \mathcal{F} \times \mathcal{F}$ . The set of relation names  $\mathcal{R} := \{\text{HAS-PART}, \text{DELIVER-AGENT}, \dots\}$  denotes conceptual relations which are also organized in a subsumption hierarchy  $isa_{\mathcal{R}} = \{(\text{HAS-HARD-DISK}, \text{HAS-PHYSICAL-PART}), (\text{HAS-PHYSICAL-PART}, \text{HAS-PART}), \dots\}$ .<sup>1</sup> Examples of emerging concept and relation hierarchies are depicted in Figure 3 (right box).

In our approach, the representation languages for semantics and domain knowledge coincide (for arguments supporting this view, cf. Allen (1993)). Linking lexical items and conceptual entities proceeds as follows: Upon entering the parsing process, each lexical item  $w$  that has a conceptual correlate  $C$  in the domain knowledge base,  $w.C \in \mathcal{F}$  (mostly verbs, nouns and adjectives), gets immediately instantiated in the knowledge base, such that for any instance  $I_w$ , initially,<sup>2</sup>  $type(I_w) = w.C$  holds (e.g.,  $w = \text{“Festplatte”}$ ,  $I_w = \text{HARD-DISK.2}$ ,  $w.C = type(\text{HARD-DISK.2}) = \text{HARD-DISK}$ ). If several conceptual correlates exist, either due to homonymy or polysemy,

<sup>1</sup>All subsumption relations,  $isa_{\mathcal{W}}$ ,  $isa_{\mathcal{F}}$ , and  $isa_{\mathcal{R}}$ , are considered to be transitive and reflexive.

<sup>2</sup>For instance, anaphora might necessitate changes of this initial reference assignment, cf. Strube and Hahn (1999).

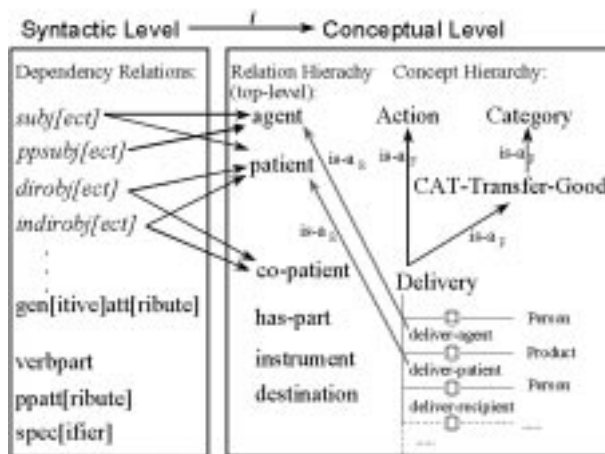


Figure 3: Relating Grammatical (left box) and Conceptual Knowledge (right box)

each lexical ambiguity is processed independently within separate context partitions of the knowledge base (Romacker and Hahn, 2000a).

### 3 Interpretable Subgraphs

In the parse tree from Figure 2, we can distinguish lexical nodes that have a conceptual correlate (e.g., “Festplatte” relating to HARD-DISK, “geliefert” relating to DELIVERY) from others that do not have such a correlate (e.g., “mit” (with), “von” (by)). Semantic interpretation capitalizes on this distinction in order to find adequate conceptual relations between the corresponding concept instances:

**Direct Linkage.** If two word nodes with conceptual correlates are linked by a *single* dependency relation, a *direct* linkage is given. Such a subgraph can immediately be interpreted in terms of a conceptual relation licensed by the corresponding dependency relation. This is illustrated in Figure 2 by the direct linkage between “Festplatte” (hard disk) and “Computers” via the *gen[itive]att[ribute]* relation, which gets mapped to the HARD-DISK-OF role linking the corresponding conceptual correlates, *viz.* HARD-DISK.2 and COMPUTER-SYSTEM.4, respectively (see Figure 4). This interpretation uses only knowledge about the conceptual correlates and the linking dependency relation.

**Indirect Linkage.** If two word nodes with conceptual correlates are linked via a *series* of depen-

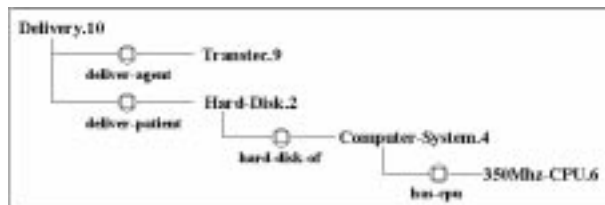


Figure 4: Semantic Interpretation of the Dependency Graph from Figure 2

dependency relations and none of the intervening nodes have a conceptual correlate, an *indirect* linkage is given. For such a “minimal” subgraph, semantic interpretation is made dependent on lexical information from the intervening nodes, as well as knowledge about the conceptual correlates and dependency relations. Figure 2 illustrates such a configuration by the linkage between “Computers” and “350Mhz-CPU” via the intervening node “mit” (*with*) and the *ppatt*[*ribute*] and *pobject* relations, the result of which is a conceptual linkage between COMPUTER-SYSTEM.4 and 350MHZ-CPU.6 via the relation HAS-CPU in Figure 4.

In order to increase the generality and to preserve the simplicity of semantic interpretation we introduce a generalization of the notion of dependency relation such that it incorporates direct as well as indirect linkage: Two content words (nouns, adjectives, adverbs or full verbs) stand in a *mediated syntactic relation*, if one can pass from one word to the other along the connecting edges of the dependency graph without traversing word nodes other than prepositions, modal or auxiliary verbs (i.e., elements of closed word classes). In Figure 2, e.g., the tuples (“Festplatte”, “Computers”) or (“Computers”, “350Mhz-CPU”) stand in mediated syntactic relations, whereas, e.g., the tuple (“Festplatte”, “Transtec”) does not, since the connecting path contains “geliefert” (*delivered*), a content word.

This leads to the following definition: Let  $w$  and  $w'$  be two content words in a sentence  $S$ . In addition, let  $w_2, \dots, w_{n-1} \in S$  ( $n \geq 2$ ) be prepositions, auxiliary or modal verbs, and  $w_1 := w$  and  $w_n := w'$ . Then we say that  $w$  and  $w'$  stand in a *mediated syntactic relation*, iff there exists an index  $l \in \{1, \dots, n\}$  so that the following two conditions hold:

1.  $w_i$  is modifier of  $w_{i+1}$  for  $i \in \{1, \dots, l-1\}$ ;
2.  $w_l$  is head of  $w_{i+1}$  for  $i \in \{l, \dots, n-1\}$ .

We call a subgraph identified by such a series  $w_1, \dots, w_n$  a *semantically interpretable subgraph* of the dependency graph of  $S$ . The definition of a mediated syntactic relation encompasses the notion of a direct linkage ( $n := 2$ , so that an empty set of intervening nodes emerges). The special cases  $l := 1$  and  $l := n$  yield an ascending and descending series of head-modifier relations, respectively.

#### 4 Semantic Interpretation Model

The model of semantic interpretation we propose comprises two constraint layers. First, *static* constraints for semantic interpretation derived from directly mapping dependency relations to conceptual roles, and, second, a search of the knowledge base which *dynamically* takes these static constraints into account. The translation from the syntactic to the semantic level is achieved in a strictly compositional

way by incrementally combining the conceptual representations of semantically interpretable subgraphs until the entire dependency graph is processed.

**Static Constraints.** Interpretation procedures operating on semantically interpretable subgraphs may inherit restrictions from the type of dependency relations or from the lexical material they incorporate. Constraint knowledge from the grammar level comes in two varieties, *viz.* via a positive list,  $D_+^{lexval}$ , and a negative list,  $D_-^{lexval}$ , of dependency relations, from which admitted as well as excluded conceptual relations,  $R_+$  and  $R_-$ , respectively, are derived by a simple static symbol mapping.

Knowledge about  $D_+^{lexval}$  and  $D_-^{lexval}$  is part of the valency specifications. It is encoded at the level of lexeme classes  $\mathcal{W}$ , such that  $lexval \in \mathcal{W} \times \mathcal{D}$ . By way of property inheritance this knowledge is passed on to all subsumed lexical classes and instances. For instance (cf. Figure 1), the lexeme class of intransitive verbs,  $VERBINTRANS \in \mathcal{W}$ , defines for its subject valency  $D_+^{(verbintrans, subject)} := \{subject\}$  and  $D_-^{(verbintrans, subject)} := \emptyset$ , whereas for prepositional adjuncts we require  $D_+^{(verbintrans, ppadj)} := \emptyset$  and  $D_-^{(verbintrans, ppadj)} := \{subject, diobject, indirobject\}$ . All these constraints are inherited by the lexeme class VERBTRANS. We then distinguish three basic cases how corresponding constraints may affect semantic interpretation processes:

1. Knowledge available from syntax *determines* the semantic interpretation, if  $D_+^{lexval} \neq \emptyset$  and  $D_-^{lexval} = \emptyset$  (e.g., the subject of a verb).
2. Knowledge available from syntax *restricts* the semantic interpretation, if  $D_+^{lexval} = \emptyset$  and  $D_-^{lexval} \neq \emptyset$  (e.g., for prepositional adjuncts).
3. If  $D_+^{lexval} = \emptyset$  and  $D_-^{lexval} = \emptyset$ , no syntactic constraints apply and semantic interpretation proceeds *entirely concept-driven*, i.e., it relies on domain knowledge only (e.g., for genitives).<sup>3</sup>

In order to transfer syntactic constraints to the conceptual level, we define  $i: \mathcal{D} \rightarrow 2^{\mathcal{R}}$ , a mapping from dependency relations onto sets of conceptual relations. Some of these mappings are already depicted in Figure 3 (e.g.,  $i(subject) := \{AGENT, PATIENT\}$ ). For dependency relations  $\delta \in \mathcal{D}$  that cannot be linked a priori to a conceptual relation (e.g., *gen*[*itive*]*att*[*ribute*]), we require  $i(\delta) := \emptyset$ .

The conceptual restrictions,  $R_+$  and  $R_-$ , must be computed from  $D_+^{lexval}$  and  $D_-^{lexval}$ , respectively, by applying the interpretation function  $i$  to each element of the corresponding sets. This leads us to  $R_+ := \{y \mid x \in D_+^{lexval} \wedge y \in i(x)\}$  and  $R_- := \{y \mid x \in D_-^{lexval} \wedge y \in i(x)\}$ .

<sup>3</sup>We have currently no empirical evidence for the fourth possible case, where  $D_+^{lexval} \neq \emptyset$  and  $D_-^{lexval} \neq \emptyset$ .

**Dynamic Constraint Processing.** Semantic interpretation implies a search in the knowledge base which takes the constraints into account that derive from a particular dependency parse tree. Two sorts of knowledge then have to be combined — first, a pair of concepts for which a connecting relation path has to be determined; second, conceptual constraints on permitted and excluded conceptual relations when connected relations are being computed. The first constraint type incorporates the content words linked by the semantically interpretable subgraph, the latter accounts for the particular dependency relation(s) holding between them. Schema (1) describes the most general mapping from the conceptual correlates,  $h.C_{from}$  and  $m.C_{to}$ , in  $\mathcal{F}$  of the two syntactically linked lexical items,  $h$  and  $m$ , respectively, to connected relation paths  $R_{con}$ .

$$si : \left\{ \begin{array}{l} \mathcal{F} \times 2^{\mathcal{R}} \times 2^{\mathcal{R}} \times \mathcal{F} \rightarrow 2^{R_{con}} \\ (C_{from}, R_+, R_-, C_{to}) \mapsto \widetilde{R_{con}} \end{array} \right. \quad (1)$$

A connected relation path  $rel_{con} \in R_{con}$  is defined by:

$$rel_{con}((r_1, \dots, r_n)) : \Leftrightarrow \forall i \in \{1, \dots, n-1\} : isa_{\mathcal{F}}(type(range(r_i)), type(domain(r_{i+1})))$$

A relation path is called *connected*, if for all its  $n$  constituent, noncomposite relations  $r_i$  the concept type of the domain of the relation  $r_{i+1}$  subsumes the concept type of the range of the relation  $r_i$ .

To compute a semantic interpretation,  $si$  triggers a search through the knowledge base and identifies all connected relation paths from  $C_{from}$  to  $C_{to}$ . Due to potential conceptual ambiguities in interpreting syntactic relations, more than one such path may exist (hence, we map to the power set of  $R_{con}$ ). In order to constrain connectivity,  $si$  takes into consideration all conceptual relations  $R_+ \subset \mathcal{R}$  a priori permitted for semantic interpretation, as well as all relations  $R_- \subset \mathcal{R}$  a priori excluded. Both of them reflect the constraints set up by particular dependency relations or non-content words figuring as lexical relators of content words. Thus,  $rel \in \widetilde{R_{con}}$  holds, if  $rel$  is a connected relation path from  $C_{from}$  to  $C_{to}$ , obeying the restrictions imposed by  $R_+$  and  $R_-$ .

If the function  $si$  returns the empty set (i.e., no valid interpretation can be computed), no dependency relation will be established. Otherwise, for all resulting relation paths  $REL_i \in \widetilde{R_{con}}$  an assertional axiom of the form  $(h.C_{from} REL_i m.C_{to})$  is added to the knowledge base, where  $REL_i$  denotes the  $i^{th}$  reading. If  $i > 1$ , conceptual ambiguities occur, resolution strategies for which are described in Romacker and Hahn (2000a).

To match a concept definition  $C$  against the constraints imposed by  $R_+$  and  $R_-$ , we define the function  $get-roles(C) =: CR$ , where  $CR$  denotes the set of conceptual roles associated with  $C$ , which are then

used as starting points for the path search. For ease and generality of specification,  $R_+$  and  $R_-$  consist of the most general conceptual relations only. Hence, the concrete conceptual roles  $CR$  and the general ones in  $R_+$  and  $R_-$  may not always be compatible. So prior to semantic interpretation, we expand  $R_+$  and  $R_-$  into their transitive closures, incorporating all their subrelations in the relation hierarchy. Thus,  $R_+^* := \{ r^* \in \mathcal{R} \mid \exists r \in R_+ : r^* isa_{\mathcal{R}} r \}$ .  $R_-^*$  is correspondingly defined.  $R_+$  restricts the search to relations contained in  $CR \cap R_+^*$ , iff  $R_+$  is not empty (otherwise, all elements of  $CR$  are allowed), whereas  $R_-$  allows only for relations in  $CR \setminus R_-^*$ .

## 5 A Sample Semantic Interpretation

Whenever a semantically interpretable subgraph is complete, semantic interpretation gets started immediately. As an example, we will consider a case of indirect linkage, as illustrated by the occurrence of auxiliary and modal verbs within a passive clause.

When interpreting indirect syntactic relations, information not only about content word nodes but also about intervening noncontent word nodes becomes available. This way, further static constraints are imposed on  $R_+$  (and  $R_-$ ) in terms of a list  $R_{lex} \subset \mathcal{R}$  of permitted conceptual relations. This information is always specified at the lexeme level. Since  $R_{lex}$  relates to closed-class items only, the required number of specifications is easy to survey.

In our example (cf. Figure 2), the content words “Festplatte” (*hard disk*) and “geliefert” (*delivered*) are linked by a mediating modal verb (“kann” (*can*)) and a passive auxiliary (“werden” (*be passive*)). The semantic interpretation schema for passive auxiliaries (2) addresses the concept type of the instance for their syntactic *subject*,  $C_{subj} = type(I_{subj}) = \text{HARD-DISK}$ , and that for their *verbpart*,  $C_{verbpart} = type(I_{verbpart}) = \text{DELIVERY}$ . The relation between these two, however, is determined by  $R_{passaux} := \{\text{PATIENT}, \text{CO-PATIENT}\}$ , constraint knowledge which resides in the lexeme specification for “werden” as passive auxiliary (cf. Figure 1).

$$si_{aux} : (C_{verbpart}, R_{passaux}, \emptyset, C_{subj}) \mapsto \widetilde{R_{con}} \quad (2)$$

With  $si_{aux}(\text{DELIVERY}, \{\text{PATIENT}, \text{CO-PATIENT}\}, \emptyset, \text{HARD-DISK})$ , we get the conceptual relation DELIVER-PATIENT (cf. Figure 3), since HARD-DISK is subsumed by PRODUCT and, thus, a legal filler of DELIVER-PATIENT  $\in R_{passaux}^*$ .

## 6 Conceptual Interpretation

*Conceptual interpretation* uses a production rule system (Yen et al., 1991) which accounts for characteristic patterns of assertions that result from the semantic interpretation process. While the outcome of semantic interpretation (cf. Figure 4) still adheres

to the surface form of the parsed sentence, conceptual interpretation abstracts away from these surface phenomena and creates a ‘normalized’, canonical conceptual representation of the input, as needed, e.g., for uniformly querying the knowledge base.

As an example of such inferences consider Figure 5, with the DELIVERS relation linking TRANSFER.9, a hardware supplier, and HARD-DISK.2. By computing a conceptual relation representing the underlying ACTION TRANSFER.9 and HARD-DISK.2 are integrated in a normalized concept graph. Note that the corresponding lexical items, “*Transtec*” and “*Festplatte*” (*hard disk*), are not linked via a mediated syntactic relation in Figure 2. Hence, we may clearly discern semantic interpretation, which operates on *single* semantically interpretable subgraphs only, from conceptual interpretation, where the inference-based interpretation of relationships among *different* subgraphs comes into play.

An independent level for conceptual interpretation also became a necessity due to analytic considerations. Often the local constraints for conceptual roles of ACTION, STATE, or EVENT concepts cannot be formulated restrictive enough for the semantic interpretation process. For example, the conceptual correlate of the verb “*possess*” does not impose any restriction on its PATIENT role (linked to the *subject* dependency relation in a semantically interpretable subgraph). Rather, restrictions apply to properly *relating* the filler of the PATIENT slot with that of the CO-PATIENT slot (*dirobject* at the dependency level). Conceptual interpretation rules are a means to further constrain these ‘context-sensitive’ aspects of the interpretation process.

Since verbs play a prominent role in dependency grammars, the production rule system for conceptual interpretation is based upon the conceptual correlates of verbs (henceforth *verb concepts*) in the knowledge base. Different views are defined for verb concepts by using three abstraction dimensions.

First, verb concepts are classified, according to the set of thematic roles they supply, as ACTION, STATE or PROCESS. DELIVERY, e.g., is assigned to ACTION, since both AGENT and PATIENT form part of the concept definition (cf. Figure 3, right box).

The second level of abstraction consists of categorizations which reflect a common core meaning. The upmost conceptual node in this hierarchy is CATEGORY. DELIVERY, e.g., is considered as a concept which represents the ACTION of transferring a GOOD

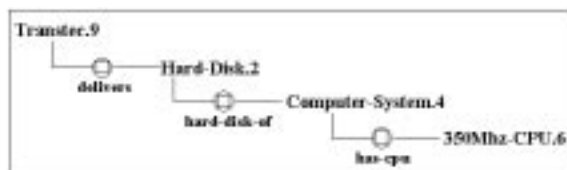


Figure 5: A Sample Conceptual Interpretation of the Dependency Graph from Figure 2

to a customer. All verb concepts belonging to this category are subsumed by the corresponding concept CAT-TRANSFER-GOOD. (We here make use of multiple inheritance mechanisms.)

Finally, every verb concept is linked to some VERB-MODEL. DELIVERY or any other verb concept of the CAT-TRANSFER-GOOD category is a constituent phase of the BUY-AND-SELL-MODEL. To generalize appropriately from individual verbs, verb categories were extracted from our text corpora that further refine a large-scale taxonomy for German verbs (Ballmer and Brennenstuhl, 1986). In this work, a total of about 20,000 verbs were subsumed by 700 categories to reflect a semantic generalization in terms of a hierarchy of verb categories.

The production rules for conceptual interpretation operate on this categorial hierarchy. Every verb concept in the hierarchy is a subconcept of exactly one category in the knowledge base. Whenever the preconditions of an interpretation rule are fulfilled, a conceptual interpretation is computed.

Conceptual and semantic interpretation depend on each other, since the basic interpretation schema (cf. expression (1) in Section 4) is supplied with actual parameters from production rules. We therefore may define another specialization of the basic interpretation schema for conceptual interpretation  $si_{conc}$ . In particular, path searches are triggered that are restricted by a positive list rendered by the applicable production rule.

For our sample sentence (cf. Figures 2 and 4), the conceptual correlate for the verb “*delivers*” (DELIVERY) is a subconcept of ACTION. Additionally, DELIVERY is a subconcept of the category CAT-TRANSFER-GOOD (cf. Figure 3). The corresponding conceptual interpretation rule is given in Figure 6. Whenever an instance of the category CAT-TRANSFER-GOOD is encountered and both its AGENT and PATIENT roles are filled, relation paths are computed from the types of the two instances involved,  $a$  and  $p$ , respectively. For each relation found by the search algorithm ( $REL$  in Figure 6), a corresponding assertion is added to the knowledge base ( $TELL$  in Figure 6). In the example, the interpretation schema is instantiated with the 4-tuple (COMPANY, {TRANSFERS-GOOD}, {}, HARD-DISK) resulting in the computation of {DELIVERS} as the proper relation link (cf. Figure 5), since it is a subrelation of TRANSFERS-GOOD.

<p>EXISTS <math>v, a, p</math>:</p> <p><math>v : \text{CAT-TRANSFER-GOOD} \sqcap</math></p> <p><math>v \text{ AGENT } a \sqcap v \text{ PATIENT } p \implies</math></p> <p>IF <math>si_{conc}(type(a), \{\text{TRANSFERS-GOOD}\}, \{\}, type(p)) \neq \emptyset</math></p> <p>THEN</p> <p><math>REL := si_{conc}(type(a), \{\text{TRANSFERS-GOOD}\}, \{\}, type(p))</math></p> <p><math>TELL a \text{ REL } p \text{ FORALL } REL \in REL</math></p>
--

Figure 6: Sample Conceptual Interpretation Rule

## 7 Evaluation

We evaluated this approach to semantic interpretation on a random selection of 54 texts (comprising 18,500 words) from two text corpora, *viz.* consumer product test reports and medical finding reports. For evaluation purposes, we concentrated on the interpretation of genitives (as an instance of direct linkage) and on the interpretation of periphrastic verbal complexes, i.e., passive, temporal and modal constructions (as instances of indirect linkage).

The underlying ontology consists of an upper generic part (containing about 1,500 concepts and relations) and domain-specific extensions relating to information technology (IT) and (parts of) anatomical medicine (MED). Each of these two domain models adds about 1,400 concepts and relations to the upper model. Corresponding lexeme entries in the lexicon provide linkages to the entire ontology.

We considered a total of 247 genitives in the sample. Recall was higher for medical texts (57%) than for IT documents (31%), though, in general, rather low. However, precision peaked at 97% and 94% for medical and IT texts, respectively. The number of syntactic constructions with modal verbs or auxiliaries amount to 292 examples. Compared to genitives, we obtained a slightly more favorable recall for both domains — 66% for MED, 40% for IT —, while precision dropped slightly to 95% and 85% for medical and IT documents, respectively.<sup>4</sup>

As with any such evaluation, idiosyncrasies of the coverage of the knowledge bases are inevitably tied with the results and, thus, put limits on too far-reaching generalizations. However, our data reflect the intention to submit a knowledge-intensive text understander to a realistic, i.e., conceptually unconstrained and therefore “unfriendly” test environment. Judged from the figures of our recall data, there is no doubt, whatsoever, that conceptual coverage of the domain constitutes *the* bottleneck for any knowledge-based approach to NLP.<sup>5</sup> Sublanguage differences are also mirrored systematically in these data, since medical texts adhere more closely to well-established concept taxonomies and writing standards than magazine articles in the IT domain, whose rhetorical styles vary to a larger degree.

## 8 Related Work

The standard way of deriving a semantic interpretation for constituency-based grammars is to assign each syntactic rule one or more semantic interpretation rules (e.g., van Eijck and Moore (1992)), and to

---

<sup>4</sup>A more detailed presentation of this evaluation study is given in Romacker and Hahn (2000b).

<sup>5</sup>For the medical domain at least, we are currently actively pursuing research on the semiautomatic creation of large-scale ontologies from weak knowledge sources, *viz.* medical terminologies; cf. Schulz and Hahn (2000).

determine the meaning of the syntactic head from its constituents. This approach has also been adopted in the few explicit attempts at incorporating semantic interpretation into a dependency grammar framework (Milward, 1992; Lombardo et al., 1998). There are no constraints on how to design and organize this rule set despite those that are implied by the choice of the semantic theory. In particular, abstraction mechanisms (going beyond the level of sortal taxonomies for semantic labels, cf., e.g., Creary and Pollard (1985)), such as property inheritance, defaults, are lacking. Accordingly, the number of rules increases rapidly and easily reaches orders of several hundreds in a real-world setting (Bean et al., 1998). As an alternative, we provide a small set of *generic* semantic interpretation schemata (by the order of 10) and conceptual interpretation rules (by the order of 30 for 200 verb concepts) instead of assigning *specific* interpretation rules to each grammar item (in our case, single lexemes), and incorporate inheritance-based abstraction in the use of these schemata during the interpretation process in the knowledge base. We clearly want to point out that while this rule system covers a wide variety of standard syntactic constructions (such as genitives, prepositional phrases, various tense and modal forms), it currently does not account for quantificational issues (like scope ambiguities) for which entirely logic-based approach (Charniak and Goldman, 1988; Moore, 1989; Pereira and Pollack, 1991) provide quite sophisticated solutions.

Sondheimer et al. (1984) and Hirst (1988) treat semantic interpretation as a direct mapping from syntactic to conceptual representations. They also share with us the representation of domain knowledge using KL-ONE-style terminological languages, and, hence, they make heavy use of property inheritance (or typing) mechanisms. The main difference to our approach lies in the status of the semantic rules. Sondheimer et al. (1984) attach single interpretation rules to each *role* (*filler*) and, hence, have to provide utterly detailed specifications reflecting the idiosyncrasies of each semantically relevant (role) attachment. Property inheritance comes only into play when the selection of alternative semantic rules is constrained to the one(s) inherited from the most specific case frame. In a similar way, Hirst (1988) uses strong typing at the conceptual *object* level only, while we use it simultaneously at the grammar and the domain knowledge level for the processing of semantic schemata.


## 9 Conclusions

We introduced an approach to the design of compact, yet highly expressive semantic interpretation schemata. They derive their power from two sources. First, the organization of grammar and domain

knowledge, as well as semantic interpretation mechanisms, are based on inheritance principles. Second, interpretation schemata abstract from particular linguistic phenomena (specific lexical items, lexeme classes or dependency relations) in terms of general configuration patterns in dependency graphs.

Underlying these design decisions is a strict separation of linguistic and conceptual knowledge. A clearly defined interface is provided which allows these specifications to make reference to fine-grained hierarchical knowledge, no matter whether it is of grammatical or conceptual origin. The interface is divided into two levels. One makes use of static, high-level constraints supplied by the mapping of syntactic to conceptual roles or supplied as the meaning of closed word classes. The other uses these constraints in a dynamic search through a knowledge base, that is parametrized by few and simple schemata. Finally, at the level of conceptual interpretation inferences emerging from semantic representations are computed by a set of productions which make reference to a verbcategorial hierarchy.

Also since the number of schemata at the semantic description layer remains rather small, their execution is easy to trace and thus supports the maintenance of large-scale NLP systems. The high abstraction level provided by inheritance-based semantic specifications allows easy porting across different application domains. Our experience rests on reusing the set of semantic schemata once developed for the information technology domain in the medical domain *without* further changes.

**Acknowledgments.** We want to thank the members of the  group for close cooperation. M. Romacker was supported by a grant from DFG (Ha 2097/5-1).

## References

- J. Allen. 1993. Natural language, knowledge representation, and logical form. In M. Bates and R. Weischedel, editors, *Challenges in Natural Language Processing*, pages 146–175. Cambridge University Press.
- T. Ballmer and W. Brennenstuhl. 1986. *Deutsche Verben. Eine sprachanalytische Untersuchung des deutschen Verbwortschatzes*. Tübingen: G. Narr.
- C. Bean, T. Rindfleisch, and C. Sneiderman. 1998. Automatic semantic interpretation of anatomic spatial relationships in clinical text. In *Proc. 1998 AMIA Annual Fall Symposium*, pages 897–901.
- E. Charniak and R. Goldman. 1988. A logic for semantic interpretation. In *Proc. of the 26th Annual Meeting of the ACL*, pages 87–94.
- L. Creary and C. Pollard. 1985. A computational semantics for natural language. In *Proc. of the 23rd Annual Meeting of the ACL*, pages 172–179.
- M. Dalrymple. 1992. Categorical semantics for LFG. In *COLING'92 – Proceedings of the 15th [sic! 14th] International Conference*, pages 212–218.
- U. Hahn, S. Schacht, and N. Bröker. 1994. Concurrent, object-oriented natural language parsing: the PARSETALK model. *International Journal of Human-Computer Studies*, 41(1/2):179–222.
- E. Hajicova. 1987. Linguistic meaning as related to syntax and to semantic interpretation. In M. Nagao, editor, *Language and Artificial Intelligence*, pages 327–351. North-Holland.
- G. Hirst. 1988. Semantic interpretation and ambiguity. *Artificial Intelligence*, 34(2):131–177.
- V. Lombardo, L. Lesmo, L. Ferraris, and C. Seidenari. 1998. Incremental interpretation and lexicalized grammar. In *CogSci'98 – Proceedings of the 20th Annual Conference*, pages 621–626.
- D. Milward. 1992. Dynamics, dependency grammar and incremental interpretation. In *COLING'92 – Proceedings of the 15th [sic! 14th] International Conference*, pages 1095–1099.
- R. Moore. 1989. Unification-based semantic interpretation. In *Proceedings of the 27th Annual Meeting of the ACL*, pages 33–41.
- F. Pereira and M. Pollack. 1991. Incremental interpretation. *Artificial Intelligence*, 50(1):37–82.
- M. Romacker and U. Hahn. 2000a. Coping with different types of ambiguity using a uniform context handling mechanism. In *Applications of Natural Language to Information Systems. Proceedings of the 5th NLDB Conference*.
- M. Romacker and U. Hahn. 2000b. An empirical assessment of semantic interpretation. In *Proc. of the 6th Applied Natural Language Processing Conference & 1st Conference of the North American Chapter of the ACL*, pages 327–334.
- S. Schulz and U. Hahn. 2000. Knowledge engineering by large-scale knowledge reuse: experience from the medical domain. In *KR'2000 – Proc. 7th International Conference*, pages 601–610.
- N. Sondheimer, R. Weischedel, and R. Bobrow. 1984. Semantic interpretation using KL-ONE. In *COLING'84 – Proc. 10th Intl. Conference & 22nd Annual Meeting of the ACL*, pages 101–107.
- M. Strube and U. Hahn. 1999. Functional centering: grounding referential coherence in information structure. *Computational Linguistics*, 25(3):309–344.
- J. van Eijck and R. Moore. 1992. Semantic rules for English. In H. Alshawi, editor, *The Core Language Engine*, pages 83–115. MIT Press.
- J. Wedekind and R. Kaplan. 1993. [Type-driven semantic interpretation of f-structures << J, W >, < R, K >>]. In *EACL'93 – Proc. 6th Conf. European Chapter of the ACL*, pages 404–411.
- W. Woods and J. Schmolze. 1992. The KL-ONE family. *Computers & Mathematics with Applications*, 23(2/5):133–177.
- J. Yen, R. Neches, and R. MacGregor. 1991. CLASP: integrating term subsumption systems and production systems. *IEEE Transactions on Knowledge and Data Engineering*, 3(1):25–32.