The Verbmobil Semantic Database

Johannes Heinecke
Humboldt-Universität zu Berlin

Karsten Worm
Universität des Saarlandes
# Contents

1 Introduction ................................................. 2

2 The Verbmoobil Project ................................. 2

3 Motivation and Goals for the Semantic Database .......... 3

4 Design and Implementation of the Database ............. 6
   4.1 Semantic Classes ..................................... 6
   4.2 The Lexicon Formalism $\mathcal{L}$....................... 6
   4.3 Semantic Classes and their Representation in $\mathcal{L}$ .... 8
   4.4 Representation of Lemmata ........................... 9

5 Application of the Semantic Database ................. 10
   5.1 Creation of the Semantic Lexicon .................... 10
   5.2 Table–based Representation ........................ 11

6 Summary ...................................................... 12
1 Introduction*

The distributed development of the modules of a large natural language processing system at different sites makes interface definitions a vital issue. It becomes even more urgent when several modules with the same intended functionality are developed in parallel and should be indistinguishable with respect to their input–output–behaviour.

Another important issue is the acquisition and maintenance of lexical information which should be stored independently of an application to make it (re)usable for different purposes.

This paper describes the design and use of the Verbmobil Semantic Database which we developed in order to deal with these issues in the area of lexical semantics in Verbmobil.

2 The Verbmobil Project

The Verbmobil project (Wahlster 1993; Bos et al. 1996) aims at the development of a speech–to–speech machine translation system for face–to–face appointment scheduling dialogues.

The application scenario of Verbmobil is that a speaker of German and a speaker of Japanese try to schedule an appointment. They communicate mostly in English, which they understand better than they speak it. If they want to say something they cannot express in English, they can have the Verbmobil system translate from both their native languages to English.

The system is being developed by about 30 partners from academia and industry in Germany, the United States and Japan. A first version, the Demonstrator, was completed in early 1994; for autumn 1996 the release of the Research Prototype is scheduled, which marks the end of the first project phase. A second phase is expected to start in 1997.

Verbmobil employs a semantic transfer approach to translation (Dorna and Emele 1996), i.e. an input utterance is syntactically analyzed, a semantic representation of the content is built up,\(^1\) and this source language semantic representation is mapped to a target language semantic representation by the transfer module. This representation is the input for the target language generation. Additionally, a dialogue processing

---


\(^1\)Syntactic and semantic analysis proceed in parallel in the Research Prototype, while they were two consequent processing steps in the Demonstrator.
module and a semantic evaluation module keep track of the discourse and answer dis-
ambiguation queries. (The relevant part of the system architecture is shown in figure 1.)

3 Motivation and Goals for the Semantic Database

The architecture of Verbmobil makes it necessary for the semantics, transfer, semantic
evaluation and generation modules to agree on the format and contents of the semantic
representations they exchange. E. g. the developers of the transfer module need to
know how the semantics of the different lemmata in the vocabulary is represented in
the structures produced by the syntax–semantics module (SynSem for short), i. e. which
predicates and structures they have to map to the target language. On the other hand,
semantics need to know which readings have to be distinguished by transfer in order to
arrive at correct translations.

This need for information becomes even more urgent when, like in Verbmobil, there
are several SynSem modules (two for German, one for Japanese), which have to pro-
duce compatible output, and the different modules are developed independently and in
parallel by several partners at different sites.2

As a frame for the exchange of semantic representations a common format, the
Verbmobil Interface Term, VIT for short, has been defined (Bos, Egg, and Schiehlen
1996). The VIT is the central data structure used at the interfaces between the language
modules of Verbmobil. A VIT is a ten–place term with slots for an utterance identifier,
a list of labelled semantic predicates, a pointer to the most prominent predicate, sortal,
anaphoric and syntactic information, temporal and aspe ctual properties, scope relations
and prosodic features. Figure 2 shows a VIT for the sentence Wir machen einen Termin
aus (We arrange an appointment).

A VIT is an underspecified representation for a set of discourse representation

---

2In the following, we concentrate on the Semantic Database for German. The Japanese version
follows the same principles.
The Verbmobil Semantic Database

Figure 2: A VIT for *Wir machen einen Termin aus* ("We arrange an appointment").
The Verbmobil Semantic Database

structures (Kamp and Reyle 1993) in which the scope of operators is not fixed yet. In the example shown in figure 2 both the scope of the declarative sentence mood operator, decl/2, and of the quantifier/indefinite, ein_card_qua/5, are left unspecified. They introduce holes, written as h1 and h2, as their scope, which can be plugged by structures subordinated to them by means of less or equal constraints, written as leq/2. Different ways of plugging the holes result in different readings. In addition to the leq/2 constraints determining all possible readings, we supply a default scoping based on syntactic structure in the predicates ccom_plug/2.3

All semantic predicates in the VIT are labelled (their first argument is the label). This allows us to group several predicates together (using the sem_group/2 predicate) and form complex substructures which can occur in the scope of operators.

Apart from the purely semantic information mentioned so far, a VIT contains sortal constraints associated with discourse markers, discourse information about anaphoric elements, syntactic agreement and tense information. Since Verbmobil deals with spoken input, we also represent prosodic information in the VIT.4

What is needed then in addition to the VIT data structure definition is a definition of the VIT's contents, for each lemma in the vocabulary of the system a definition of the semantic predicates and other types of information, e.g. sortal restrictions, it introduces in the slots of the VIT. E.g. for the verb ausmachen in the example above, we need to specify that it introduces a predicate ausmachen(L1,I1) together with argument roles arg1(L1,I1,I2) and arg3(L1,I1,I3) in the semantics slot and sort(L1,ment_communicat_poly) in the sorts slot.

If a source providing this kind of information to the developers of the separate modules is available, the modules which deliver (the two SynSem modules) or process (especially the transfer module) VITs conforming to this definition can be developed in parallel. It would also be desirable to use this information source directly in the construction of the linguistic knowledge bases to guarantee consistency between the output and the specifications.

To meet these goals, we have developed the Verbmobil Semantic Database, which we will describe in the remainder of this paper.

4 Design and Implementation of the Database

The semantic database is organized around a set of abstract semantic classes (Bos, Egg, and Schiehlen 1996), which are used to classify the lemmata in the vocabulary. It is

3For more details on this underspecified approach to semantics, the reader might consult (Bos 1995; Bos et al. 1996).

4The VIT in figure 2 has been generated from typed input and thus contains no real prosodic information.
Table 1: A few examples of semantic classes

<table>
<thead>
<tr>
<th>Class</th>
<th>PredScheme</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>transitive verb</td>
<td>$R(L,I), \ argX(L,I,I1), \ argY(L,I,I2)$</td>
<td>treffen</td>
</tr>
<tr>
<td>common noun</td>
<td>$R(L,I)$</td>
<td>Termin</td>
</tr>
<tr>
<td>det_quant</td>
<td>$R(L,I,H)$</td>
<td>jeder</td>
</tr>
<tr>
<td>demonstrative</td>
<td>demonstrative($L,I,I1$)</td>
<td>dieser</td>
</tr>
<tr>
<td>wh_question</td>
<td>whq($L,I,H$), tloc($L2,I2,I1$), time($L1,I1$)</td>
<td>wann</td>
</tr>
</tbody>
</table>

implemented using the lexicon formalism $\mathcal{L}X^4$.

### 4.1 Semantic Classes

The semantic classes in use are originally based on a morpho–syntactic classification of the words in the vocabulary of the system which has been refined to account for the semantic properties. This has been decided upon, because words of a certain word–class usually have the same semantic properties. In the example given below, it is shown that transitive verbs all need an instance and two arguments with their semantic/thematic roles.

For each semantic class a representation scheme, called the predscheme, has been defined, which specifies the predicates together with their arity and arguments appearing in a VIT for instances of the class.

As an example consider the class transitive verb. A transitive verb is represented as $R(L,I), \ argX(L,I,I1), \ argY(L,I,I2)$. I. e., it introduces some relation $R$ and two thematic roles ($I$ is the event variable, $L$ a label used to refer to the verb’s semantic contribution, and $I1$ and $I2$ are the instances filling the roles). The verb’s relation and the thematic roles it assigns have to be defined for each verb in the database. Cf. table 1 for further examples of semantic classes together with their predschemes.

### 4.2 The Lexicon Formalism $\mathcal{L}X^4$

The semantic database makes use of the lexicon formalism $\mathcal{L}X^4$ developed in the course of the Verbmobil project (Gebhardi and Heinecke 1995a; Gebhardi 1996).

The Lexicon Formalism $\mathcal{L}X^4$ has been used since summer 1994 within Verbmobil’s lexicon group. It is based on feature-structures (permitting disjunction and negation) embedded in an inheritance hierarchy of classes.

In $\mathcal{L}X^4$ the task of constructing a lexicon is split up into four parts:

---

5 $x$ and $y$ stand for the values $\{1, 2, 3\}$, since $arg1$, $arg2$, $arg3$ are the thematic roles used in Verbmobil.
1. Modelling the lexicon (i.e. its linguistic classes),

2. data-acquisition (can be done at the same time by different contributors),

3. definition of the application-interface (data can be compiled into every format needed after being processed by the \textsc{lex}-machine), and

4. efficient storage.

Modelling a lexicon involves defining classes, their appropriate features, and inheritance relations between classes. Examples for defining classes will be given below in section 4.3; appropriateness of features is dealt with in the remainder of this section. For data acquisition, a graphical acquisition tool has been implemented (Heinecke 1996). How the application interface is used in the context of the semantic database will be shown in section 5. Part of the application interface is the \textsc{lex}-TRAFO which outputs the stored information in any format required. A database system for efficient storage has been developed (Kruschwitz and Gebhardi 1996)

Among other formalism constructs, the possible values of a feature can be specified in two ways. If there is no restriction on the value of a feature, it is assigned the most general value keyword (top):

\begin{verbatim}
   predname: top .
\end{verbatim}

Otherwise, the formalism allows to define the appropriateness conditions of a feature, using disjunctions to specify the appropriate values as in the following example (the underlined values are the appropriate ones which can be assigned to the feature sort_of_inst):

\begin{verbatim}
   sort_of_inst: ( abstract \ anything \ communicat.sit \ communicat.result.poly \ person ) .
\end{verbatim}

For constructing morphological lexica, inflection or lexical rules can easily be implemented to generate multiple instances of a single entry (Gebhardi and Heinecke 1995b; Heinecke and Gebhardi 1995).

Database entries, called bases, are instances of a class. Consequently, they assign values to the features they inherit from their class which are not yet fully specified by the class definition. For a verb’s base, e.g., one has to specify its predicate name, thematic roles, the sort of its instance, etc.
4.3 Semantic Classes and their Representation in $\mathcal{L}^4$

The abstract semantic classes of section 4.1 have been modelled in the lexicon formalism $\mathcal{L}^4$ along the following lines.

Firstly, a general superclass $\text{semdb}_c$ is defined from which all classes inherit features for the lemma, the main predicate’s name, the part of speech etc. The individual subclasses corresponding to the abstract semantic classes additionally introduce a specific predscheme for each predicate associated with words of this class and features for sortal information, thematic roles etc.

```
class semdb_c :< top >: % - Main class from which all classes inherit.
syntax_link: top & % - Link to syntactic lexicon.
predname: top & % - Name of the semantic predicate.
lemma: top & % - Lemma of the entry.
pos: top . % - Part of Speech of the occurrences in the corpora.
```

While the abstract semantic classes are not hierarchically organized, their modelling in $\mathcal{L}^4$ makes use of a hierarchy to capture generalizations. For instance, we integrate all properties the verb classes have in common and place them in an abstract verb class $\text{verb}_c$ from which all verb classes, e.g. $\text{transitive}_c$, inherit, cf. figure 3 (classes corresponding to semantic classes are shown in boldface) and below.

```
class verb_c :< semdb_c >: % - All verbal classes inherit this.
sort_of_inst: top . % - Sort of eventuality.
```

```
class transitive_c :< verb_c >: % - Transitive verbs
semclass: transitive_verb & % - Semantic class.
predscheme: 'L,I' & % - PredScheme for the PredName of all transitive verbs.
```
The Verbmobil Semantic Database

predscheme_a1: ‘L,I,I1’ & % - PredScheme for the first 
predscheme_a2: ‘L,I,I2’ & % and the second argument.  
role_a1: (arg1 \ arg2 \ arg3) & % - Thematic roles of the arguments 
role_a2: (arg1 \ arg2 \ arg3) . % of the verb (restricted 
 % to three valid values).

As a second example, consider the following definition for the abstract semantic class common_noun:

class common_noun_c :< semdb_c >: % - Standard nouns  
predscheme: ‘L,I’ & % - PredScheme for standard nouns.  
sort_of_inst: top & % - Sort of instance.  
semclass: common_noun . % - Semantic class.

4.4 Representation of Lemmata

A base for a lemma consists of its classification together with its idiosyncratic properties in terms of feature values; it inherits the feature values which are specified in the definition of the class. Among the idiosyncratic information we have predicate names, sortal restrictions etc. Thus an entry inherits the predscheme from the class, while the concrete predicate name in the predscheme is defined in the entry itself.

base ‘Termin’ :<< common_noun_c >>: % - The entry ‘Termin’ 
% inherits its structure from 
% from the class 
% ‘common_noun_c’. 
pos: ‘NN’ & % - Further individual 
lemma: ‘Termin’ & % specification for 
syntax_link: ‘termin’ & % the current entry. 
predname: ‘termin’ & 

base ‘ausmachen’ :<< transitive_c >>: % - The entry ‘ausmachen’ 
% inherits its structure from 
% the class 
% ‘transitive_c’. 
pos: ‘VVFIN;VVINF’ & % - Further specifications. 
lemma: ‘ausmachen’ & 
syntax_link: ‘ausmachen’ & 
predname: ‘ausmachen’ & 
sort_of_inst: (communicat_sit \ mental_sit) &
role_a1: ‘arg1’ & role_a2: ‘arg3’.

When processing the class definitions and the bases, the $L^4$-machine will calculate all instances from the specifications and expand the base accordingly.

5 Application of the Semantic Database

The Semantic Database is currently being used for creating the semantic lexica of the syntactic–semantic modules of Verbmobil, for producing a table of lemmata with the predicates and other types of information they introduce in a VIT and for checking the correctness of the generated interface terms automatically; it can also be accessed via the World Wide Web.

A similar procedure is used to generate the semantic lexicon etc. for the Japanese syntactic–semantic module of Verbmobil (Mori 1996).

5.1 Creation of the Semantic Lexicon

Consider the compilation of the semantic lexicon from the database for the German SynSem module SynSemS3. To guarantee consistency between the output of the SynSem module and the specifications in the database, the semantic lexicon is generated out of the semantic database.

After the $L^4$–machine has processed the entries and expanded them according to the class definitions, the $L^4$–TRAFO compiles the $L^4$ output into the format required for the semantic lexicon.

\[
\text{sort1-trafo(Base, Class, } \\
\quad \{ \text{predname:Predn,} \\
\quad \text{syntax_link:Sl,} \\
\quad \text{sort_of_inst:Si,} \\
\quad \text{usb_macro:M} \\
\quad \} \Rightarrow \\
\quad \text{fmt}(+sem_lex(Cat,~w) \ short_for~n \ ~w(Cat,~w,(~w)) \ .\ ~n",} \\
\quad \text{[Sl, M, Predn, Si], []}).
\]

\[
\text{trans-trafo(Base, Class, } \\
\quad \{ % - Rule for bivalent verbs.}
\]

---

$^6$SynSemS3 is the syntactic–semantic module developed by Siemens AG (syntax), University of the Saarland and University of Stuttgart (semantics). The other SynSem module developed by IBM Germany makes use of the table output (cf. section 5.2) of the database to create a semantic lexicon.
The Verbmobil Semantic Database

[ predname:Pn,
syntax_link:Sl,
sort_of_inst:Si,
role_a1:R1,
role_a2:R2,
usb_macro:M
] ) =>
fmt("sem_lex(Cat,˜w) short_for˜n ˜w(Cat,˜w,(˜w),[˜w,˜w]).˜n",
[Sl, M, Pn, Si, R1,R2], []).}

The two examples above appear in the semantic lexicon as:

sem_lex(Cat, termin) short_for
common_noun_sem(Cat, termin, (time_sit_poly)) .

sem_lex(Cat, ausmachen) short_for
trans_verb_sem(Cat, ausmachen, (communicat_sit;mental_sit),
[arg1,arg3]) .

The syntactic lexicon contains calls to the macro sem_lex/2 which is expanded in the semantic lexicon as shown above. The mapping from syntactic to semantic lexical entries is achieved via the second argument of sem_lex/2, which originates from the feature syntax_link in the semantic database.7

5.2 Table–based Representation

Apart from compiling out semantic lexica, we generate a table of lemmata together with their semantic representations and additional information out of the database by using a different set of transformation rules for LEX-TRAFO. This table is used by the transfer developers as a basis for writing transfer rules and as an information source for the automatic correctness check on VIT representations.

transitive_trafo(Base, Class, % - Rule for bivalent verbs.
[ lemma:Lm,
pos:Pos,
semclass:Semc,
predname:Pn,
predscheme:Ps,
predscheme_a1:Ps1,
predscheme_a2:Ps2,
role_a1:Ra1,
role_a2:Ra2,

7The first argument of sem_lex/2 ranges over entry nodes of the feature structures of the lexical entry.
The Verbmobil Semantic Database

sort_of_inst:Si,
inst_link:Il,
sort_a1:Sa1, a1_link:Al1,
sort_a2:Sa2, a2_link:Al2
} =>
fmt("˜w ˜w ˜w ˜w,˜w,˜w ˜w ˜w(˜w),˜w(˜w),˜w(˜w) ˜w/˜w,˜w/˜w,˜w/˜w - -˜n",
[ Base, Lm, Pos, Pn,Ra1,Ra2, Semc, Pn,Ps, Ra1,Ps1, Ra2,Ps2,
  Il,Si,Al1,Sa1,Al2,Sa2], []).

default_ps1_inst1(Base, Class, % - Default rule for entries with
  [ lemma:Lm, % one PredScheme and one Sort
    pos:Pos, % (used e.g. by 'common_noun').
    semclass:Semc,
    predname:Pn,
    predscheme:Ps,
    sort_of_inst:Si
  ] =>
fmt("˜w ˜w ˜w ˜w ˜w ˜w(˜w) ˜w - -˜n",
  [ Base, Lm, Pos, Pn, Semc, Pn,Ps, Si], []).

In the table output the two examples above appear as:
Termin Termin NN termin common_noun termin(L,I) I/time_sit_poly - -
ausmachen ausmachen VVFIN;VVINF ausmachen,arg1,arg3 transitive_verb ...
ausmachen(L,I),arg1(L,I,I1),arg3(L,I,I2) I1/communicat_sit;mental_sit - -

In general the concept of TRAFO is trying to map the output of the LEX4-machine
onto the first matching rule in the rule system. Thus only a few class specific rules
are necessary, default rules will cover the entries of the majority of the classes to be
transformed.

6 Summary

We have successfully used the semantic database to deal with about 2000 German
and 150 Japanese lemmata for version 1.0 of the Research Prototype in the way de-
scribed, especially to generate semantic lexica for the German syntax–semantics mod-
ule SynSemS3, and the Japanese one developed by DFKI Saarbrücken and the Univer-
sity of the Saarland.

The use of the semantic database by both the semantics module and the transfer
module guarantees consistency between the representations produced by the semantics
module and the expectations of the transfer module, while both can be developed in
parallel.

Currently the Semantic Database for the German lemmata can be accessed via
WWW at http://www.compling.hu-berlin.de/~vm/semdb.
References


