LANGUAGE GENERATION FROM CONCEPTUAL STRUCTURE:

SYNTHESIS OF GERMAN IN A JAPANESE/GERMAN MT PROJECT

J. Laubsch, D. Roesner, K. Hanakata, A. Lesniewski
Projekt SEMSYN, Institut fuer Informatik, Universitaet Stuttgart
Herdweg 51, D-7000 Stuttgart 1, West Germany

ABSTRACT

This paper describes the current state of the SEMSYN project[1], whose goal is to develop a module for generation of German from a semantic representation. The first application of this module is within the framework of a Japanese/German machine translation project. The generation process is organized into three stages that use distinct knowledge sources. The first stage is conceptually oriented and language independent, and exploits case and concept schemata. The second stage employs realization schemata which specify choices to map from meaning structures into German linguistic constructs. The last stage constructs the surface string using knowledge about syntax, morphology, and style. This paper describes the first two stages.

INTRODUCTION

SEMSYN's generation module is developed within a German/Japanese MT project. Fujitsu Research Labs. provide semantic representations that are produced as an interim data structure of their Japanese/English MT system ATLAS/II (Uchida & Sugiyama, 1980). The feasibility of the approach of using a semantic representation as an interlingua in a practical application will be investigated and demonstrated by translating titles of Japanese papers from the field of "Information Technology". This material comes from Japanese documentation data bases and contains in addition to titles also their respective abstracts. Our design of the generation component is not limited to titles, but takes extensibility to abstracts and full texts into account. The envisioned future application of a Japanese/German translation system is to provide natural language access to Japanese documentation data bases.

OVERALL DESIGN OF SEMSYN

Fig. 1 shows the stages of generation. The Japanese text is processed by the analysis part of FUJITSU's ATLAS/II system. Its output is a semantic net which serves as the input for our system.

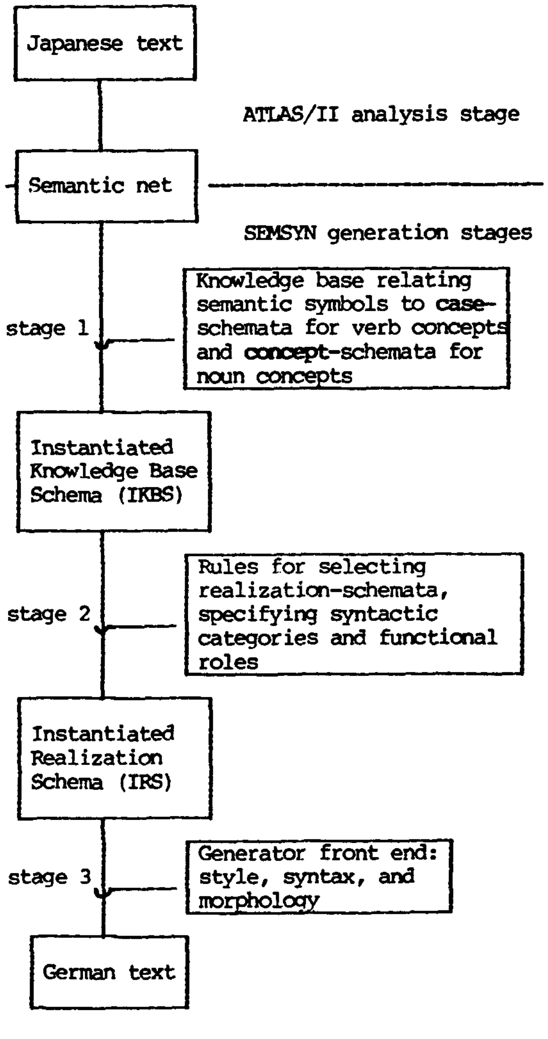

Japanese text
ATLAS/II analysis stage
Semantic net
SEMSYN generation stages
stage 1
Knowledge base relating semantic symbols to case-schemata for verb concepts and concept-schemata for noun concepts
Instantiated Knowledge Base Schema (IKBS)
stage 2
Rules for selecting realization-schemata, specifying syntactic categories and functional roles
Instantiated Realization Schema (IRS)
stage 3
Generator front end: style, syntax, and morphology
German text

Fig. 1 Stages of Generation

[1] SEMSYN is an acronym for semantic synthesis. The project is funded by the "Informationslinguistik" program of the Ministry for Research and Technology (BMFT), FRG, and is carried out in cooperation with FUJITSU Research Laboratories, Japan.

CONCEPTUAL STRUCTURE

ATLAS/II's semantic networks (see Fig.2) are directed graphs with named nodes and labelled arcs. The names of the node are called "semantic symbols" and are associated with Japanese and English dictionary entries. The labelled arcs are used in two ways:

a) Binary arcs either express case relations between connected symbols or combine substructures
b) Unary arcs serve as modifying tags of various kinds (logical junctors, syntactic features, stylistics, ...)

The first stage of generation is conceptually oriented and should be target language independent. We use frame structures in a KRL-like notation. Our representation distinguishes between **case schemata** (used to carry the meaning of actions), and **concept schemata** (used to represent "things" or "qualities"). Each semantic symbol points to such a schema. These schemata have three parts:

(1) **roles**: For action schemata, these are the usual cases of Fillmore (e.g. AGENT, OBJECT, ...); for concept schemata roles describe how the concept may be further specified by other concepts.

(2) **transformation rules**: These are condition-action pairs that specify which schema is to be applied, and how its roles are to be filled from the ATLAS/II net.

(3) **choices** describe possible syntactic patterns for realization.

<u>Examples</u>:

Case schema for the semantic symbol ACHIEVE:

```
(ACHIEVE (superc goal-oriented-act)
  (roles
   (Agent (class animate))
   (Goal)
   (Method (class abstract-object))
   (Instrument (class concrete-object)))
  (transformation-rules ...)
  (choices ...)))
```

The concept schema for SPEAKER is:

```
(SPEAKER (superc animate)
  (roles
   (Performs-act-for (class organization))
   ...)
  (transformation-rules ...)
  (choices ...))).
```

FROM CONCEPTS TO LANGUAGE

In the target language oriented stage 2, the following decisions have to be made:

i) Retrieval of the lexical entry of a German verb and its associated case frame corresponding to the IKBS.
ii) Selection of lexical entries for the other semantic symbols.
iii) Selection of a realization schema (RS), mapping of IKBS roles to RS functional roles, and inferring syntactic features.

In i) a simple retrieval may not suffice. In order to choose the most adequate German verb, it will e.g. be necessary to check the fillers of an IKBS. For example, the semantic symbol REALISE may translate to "realisieren", "implementieren" etc.. If the Instrument role of REALISE were filled with an instance of the PROGRAM concept, we would choose the more adequate word sense "implementieren".

In ii) sometimes similar problems arise. For example, the semantic symbol ACCIDENT may translate to the German equivalent of "accident", "error", "failure" or "bug". The actual choice depends here on the filler of ACCIDENT's semantic role for "where it occurred".

iii) The **choices** aspect of a schema describes different possibilities how an instance may be realized and specifies the conditions for selection. (This idea is due to McDonald (1983) and his MUMBLE system). The factors determining the choice include:

(a) Which roles are filled?
(b) What are their respective fillers?
(c) Which type of text are we going to generate?

For example if the Agent-role of a case frame is unfilled, we may choose either passivation or selection of a German verb which maps the semantic object into the syntactic subject. If neither agent nor object are filled, nominalization is forced.

A realization schema (RS) is a structure which identifies a syntactic category (e.g. CLAUSE, NP) and describes its functional roles (e.g. HEAD, MODIFIER, ...). We employ Winograd's terminology for functional grammar (Winograd, 1983). In general, case schemata will be mapped into CLAUSE-RS and concept schemata are mapped into NP-RS. A CLAUSE-RS has a features description and slots for verb, subject, direct object, and indirect objects. A features description may include information about voice, modality, idiomatic realization, etc.. There are realization schemata for discourse as well as titles. The latter are special cases of the former, forcing nominalized constructions.

REFERENCING AND FOCUSSING

For referencing and other phenomena like focussing, the simple approach of only allowing a schema instance as a filler is not sufficient. We therefore included in our knowledge representation a way to have descriptors as fillers. Such descriptors are references to parts of a schema. In the following example the filler of USE's Object-slot is a reference descriptor to SYNTHESIZE's Object-slot:

```
X = (a USE with
      (Object
        (the Object from
           (a SYNTHESIZE with
              (Object [FUNCTION])
              (Method [DYNAMIC-PROGRAMMING])))
      (Purpose (an ACCESS with
                     (Object [DATA-BASE]))))
```

X could be realized as:
"Using functions, that are synthesized by dynamic programming for data-base access."

In general, descriptors have the form:

```
(the <path> from <IKBS>)
<path> = <slot>...
```

A description can be realized by a relative clause.

The same technique of referring to a sub-structure may as well be used for focussing. For example, embedding X into

(the Purpose from X)

expresses that the focus is on X's Purpose slot, which would yield the realization:

"Database access using functions that are synthesized by dynamic programming."

A WALK WITH SEMSYN

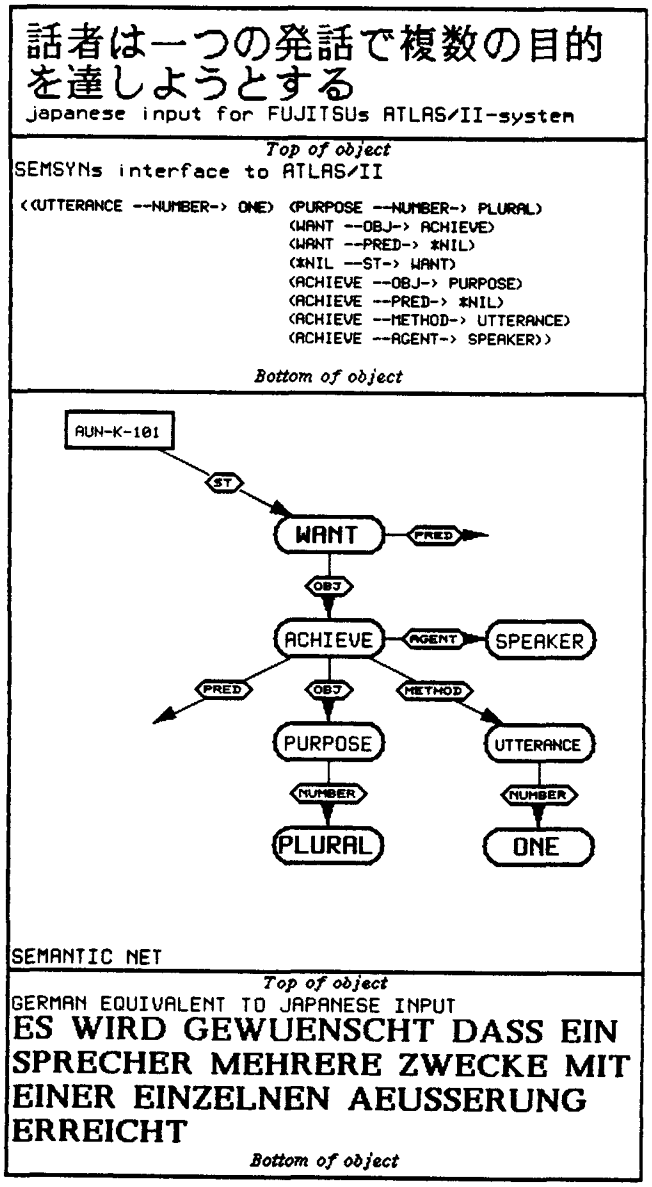
Let us look at the first sentence from an abstract. Figure 2 contains the Japanese input and the semantic net corresponding to ATLAS/II's analysis.

In stage 1, we first examine those semantic symbols which have an attached case schema and instantiate them according to their transformation rules.

In this example the WANT and ACHIEVE nodes (flagged by a PRED arc) are case schemata. Applying their tranformation rules results in the following IKBS:

```
(a WANT with
  (Object
    (an ACHIEVE with
      (Agent [SPEAKER])
      (Object [PURPOSE (Number [PLURAL])])
      (Method [UTTERANCE (Number [SINGLE])]))))
```

In stage 2, we will derive a description of how this structure will be realized as German text.



話者は一つの発話で複数の目的を達しようとする
japanese input for FUJITSUs ATLAS/II-system

Top of object
SEMSYNs interface to ATLAS/II

```
((UTTERANCE --NUMBER-> ONE) (PURPOSE --NUMBER-> PLURAL)
                            (WANT --OBJ-> ACHIEVE)
                            (WANT --PRED-> *NIL)
                            (*NIL --ST-> WANT)
                            (ACHIEVE --OBJ-> PURPOSE)
                            (ACHIEVE --PRED-> *NIL)
                            (ACHIEVE --METHOD-> UTTERANCE)
                            (ACHIEVE --AGENT-> SPEAKER))
```

Bottom of object

SEMANTIC NET

Top of object
GERMAN EQUIVALENT TO JAPANESE INPUT
ES WIRD GEWUENSCHT DASS EIN SPRECHER MEHRERE ZWECKE MIT EINER EINZELNEN AEUSSERUNG ERREICHT
Bottom of object

Figure 2. From Japanese to German

First, consider the outer WANT act. There is no Agent, so we choose to build a clause in passive voice. Next, we observe that WANT's object is itself an act with several filled roles and could be realized as a clause. One of the choices of WANT fits this situation. Its condition is that there is no Agent and the Object will be realized as a clause. Its realization schema is an idiomatic phrase named *Es-Part*:

"Es ist erwuenscht, dass <CLAUSE>"

("It is wanted that <CLAUSE>")

Now consider the embedded <CLAUSE>. An ACHIEVE act can be realized in German as a clause by the following realization schema:

```
(a CLAUSE with
 (Subject <NP-realization of Agent-role>
 (Verb "erreich_"
 (DirObj <NP-realization of Object-role>
 (IndObjs
  (a PP with
    (Prep (One-of ["durch" "mit" "mittels"]))
    (PObj <NP-realization of Method-role>))))
```

This schema is not particular to ACHIEVE. It is shared by other verbs and will therefore be found via general choices which ACHIEVE inherits.

The Agent of ACHIEVE's IKBS maps to the Subject and the Method is realized as an indirect object. Within the scope of the chosen German verb "erreichen" (for "achieve"), a Method role maps into a PP with one of the prepositions "durch", "mit", "mittels" (corresponding to "by means of"). This leads to the following IRS:

```
(a CLAUSE with
   (Features (Voice Passive
             Idiom *Es-Part*)
   (Verb "wuensch_")  ;want
   (DirObj
     (a CLAUSE with
        (Subject (a NP with
                   (Head "Sprecher")));speaker
        (Verb "erreich_")
        (DirObj
         (a NP with
          (Features (Numerus= Plural))
          (Head ["Ziel", "Zweck"]) ; purpose
          (Adj "mehrere"))         ; multiple
        (IndObjs
         ((a PP with
           (Prep ["durch", "mit", "mittels"])
            (PObj
              (a NP with
               (Features (Numerus Singular))
               (Head "Aeusserung") ;utterance
               (Adj "einzeln") ; single )))))
```

Such an instantiated realization schema (IRS) will be the input of the generation front end that takes care of a syntactically and morphologically correct German surface structure (see Fig. 2).

EXPERIMENTS WITH OTHER GENERATION MODULES

We recently studied three generation modules (running in Lisp on our SYMBOLICS 3600) with the objective to find out, whether they could serve as a generation front end for SEMSYN: SUTRA (Busemann, 1983), the German version of IPG (Kempen & Hoenkamp, 1982), and MUMBLE (McDonald, 1983).

Our IRS is a functional grammar description. The input of SUTRA, the "preterminal structure", already makes assumptions about word order within the noun group. To use SUTRA, additional transformation rules would have to be written.

IPG's input is a conceptual structure. Parts of it are fully realized before others are considered. The motivation for IPG's incremental control structure is psychological. In contrast, the derivation of our IRS and its subsequent rendering is not committed to such a control structure. Nevertheless, the procedural grammar of IPG could be used to produce surface strings from IKBS by providing it with additional syntactic features (which are contained in IRS).

Both MUMBLE and IPG are conceptually oriented and incremental. MUMBLE's input is on the level of our IKBS. MUMBLE produces functional descriptions of sentences "on the fly". These descriptions are contained in a constituent structure tree, which is traversed to produce surface text. Our approach is to make the functional description explicit.

ACKNOWLEDGEMENTS

We have to thank many colleagues in the generation field that helped SEMSYN with their experience. We are especially thankful to Dave McDonald (Amherst), and Eduard Hoenkamp (Nijmegen) whose support - personally and through their software - is still going on. We also thank the members of the ATLAS/II research group (Fujitsu Laboratories) for their support.

REFERENCES

Uchida,H. & Sugiyama: A machine translation system from Japanese into English based on conceptual structure, Proc. of COLING-80, Tokyo, 1980, pp.455-462

Winograd, T.: Language as a cognitive process, Addison-Wesley, 1983

McDonald, D.D.: Natural language generation as a computational problem: An Introduction; in: Brady & Berwick (Eds.) Computational model of discourse, MIT-Press, 1983, pp.209-265

Kempen, G. & Hoenkamp,E.: Incremental sentence generation: Implication for the structure of a syntactic processor; in Proc. COLING-82, Prague, 1982, pp.151-156

Busemann,B.: Oberflaechentransformationen bei der Generierung geschriebener deutscher Sprache; in: Neumann, B. (Ed.) GWAI-83, Springer, 1983, pp.90-99