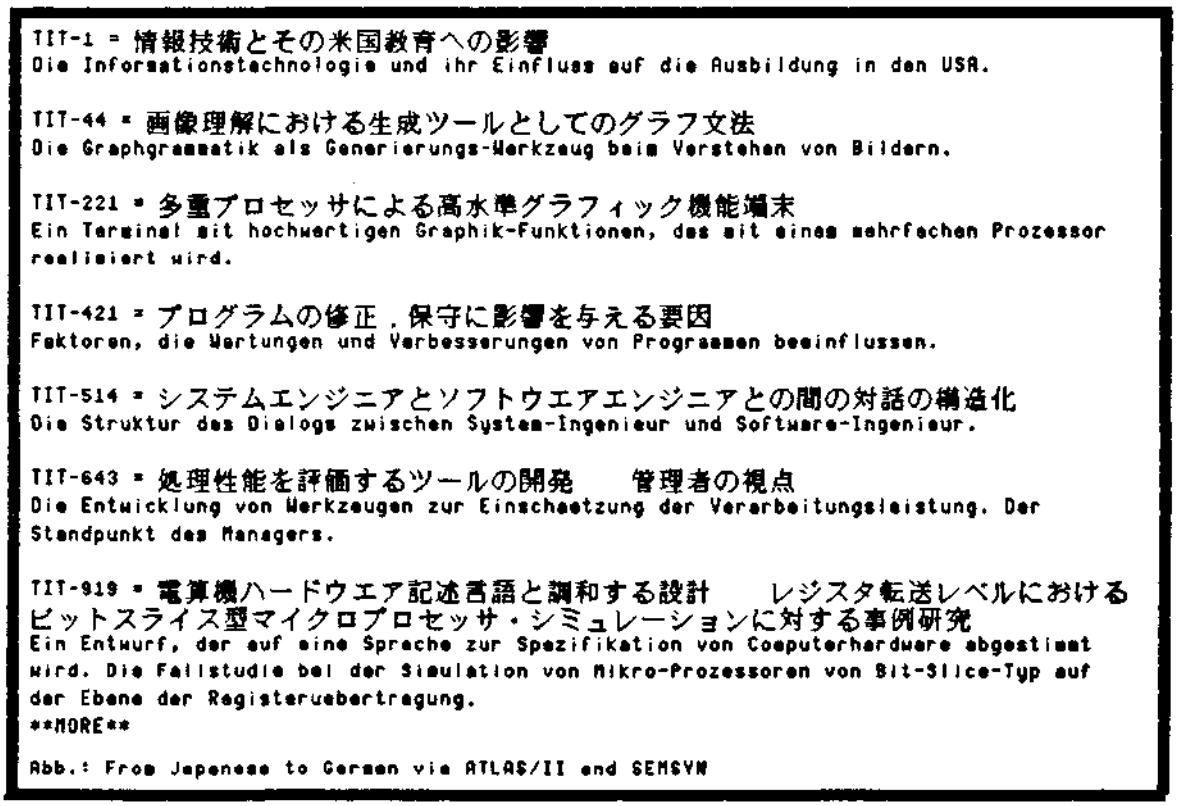
Generating Natural Language from Semantic Representations

- An AI Approach to a Japanese/German
Machine Translation Project -

M.Emele, W.Kehl, D.Rösner
Project SEMSYN, Institut f. Informatik
Univ. Stuttgart, Herdweg 51
D-7000 Stuttgart 1
West Germany

Background

An american study - published in Nature, 308 (1984) - evaluated cir. 9000 Japanese scientific papers. 75 percent of them are published exclusively in Japanese, only a 5th of Japanese papers are currently evaluated from Western refereeing and information services. The main conclusion of the study was, that the general opinion all important Japanese stuff would be published in English is not true, at least for the applied sciences. From this background and from the Japanese success in a lot of fields of modern technologies stems a wider interest in having access to Japanese material and in having help to overcome the language barrier.



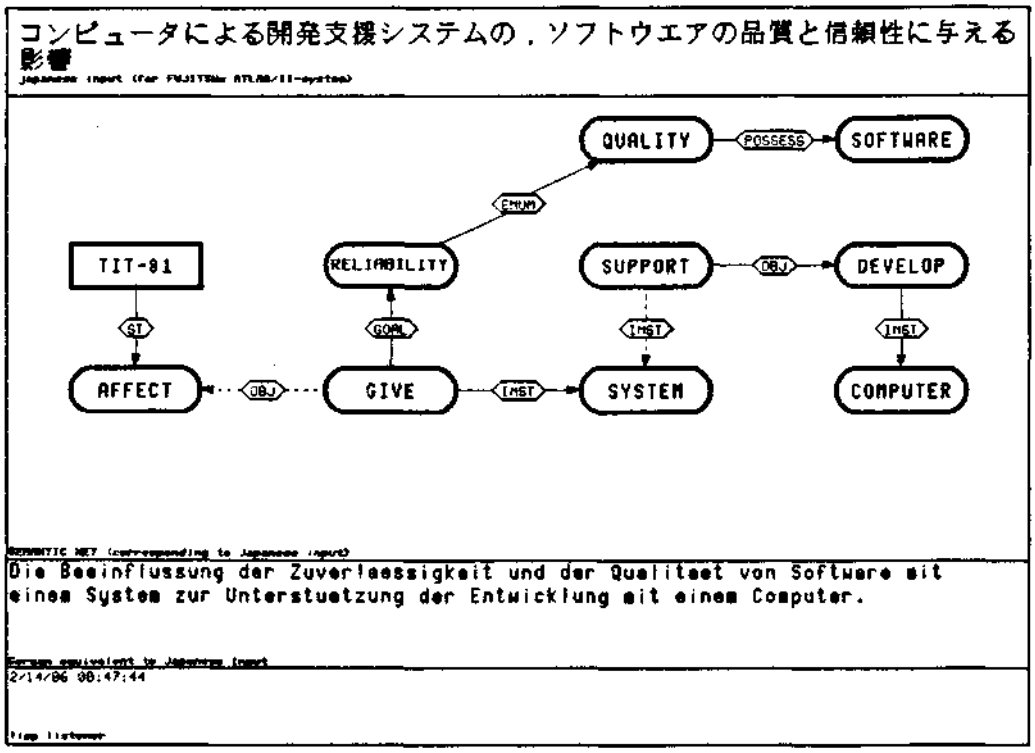
In this paper we will report on our experience from a 2 1/2 year project that designed and implemented a prototypical Japanese to German translation system for titles of Japanese papers.

1. SEMSYN - a Japanese/German translation system

The project SEMSYN - SEMSYN is an acronym for SEMantic SYNthesis - produced a machine translation system that is unique in some sense. This uniqueness does not only originate from the language pair Japanese to German that is dealt with, but as well from the approach that was taken in the project.

If one has a close look, SEMSYN is only a subsystem. Only in conjunction with the ATLAS/II-System of the Japanese cooperation partner FUJITSU we get a complete Japanese to German translation. Interface of the subsystems is a semantic representation that should reflect the content of the Japanese input.

The analysis of the Japanese input - currently at most titles of scientific papers from the field of information technology - and its transformation into the semantic representation is the task of ATLAS/II. SEMSYN's part is to produce a correct and understandable German text for these semantic representations.



The ATLAS/II system was designed for later use in multilingual translation (Uchida & Sukiyama 1980). Besides the component for analysing Japanese it has a component for generating English (that is currently commercially available in Japan) and some experimental components for the generation of other languages.

2. The overall design of the SEMSYN-System

In our project we started by analysing a first sample of semantic nets delivered from FUJITSU. During these discussions an overall concept of the generation model was already developed (Laubsch et al., 1984). In the current implementation this design was refined and further elaborated.

SEMSYN's generation from FUJITSU'S nets to German surface structures is done in three main steps. In each of these steps, different data structures and different knowledge bases come into play.

The first step is to transform the semantic net delivered by FUJITSU into an expression of our own representation language the so called IKBS-descriptions. IKBS stands for Instantiated Knowledge Base Schemata. This transformation does not only lead to a more structured representation, it helps as well to keep the generation module somewhat independent from the special form of the FUJITSU interface.

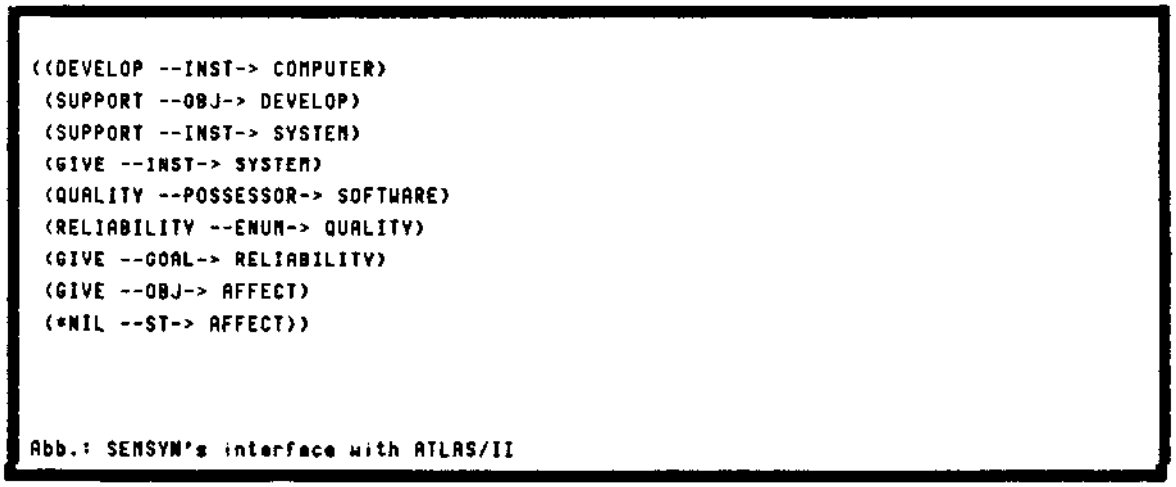


Abb.: SEMSYN's interface with ATLAS/II

The second - and probably most important - step is to decide in which way the content of the semantic representation should be uttered as German text. The output of this step is a functional description of the intended utterance in grammatical terms (IRS = Instantiated Realization Schema). The IRS description already contains base forms of German words and their semantic features.

The third step - the generator-front-end - takes the IRS description and produces a corresponding syntactically and morphologically correct German surface structure.

3. Generation from frame descriptions

The main part of the generation starts from the frame description derived from FUJITSU's semantic nets. Since such frame-like structures are used in a variety of systems for knowledge representation the generator of the SEMSYN project is as well applicable for use in help systems, explanation components and other natural language interfaces.

```
(THE :OBJECT
     FROM
     (A GIVE
        WITH
        (:GOAL =
               (AN ENUMERATION
                   WITH
                   (:ARGL = (A RELIABILITY) (A QUALITY))
                   (:POSSESSOR = (A SOFTWARE))))
        (:INSTRUMENT =
                     (THE :INSTRUMENT
                          FROM
                          (A SUPPORT
                             WITH
                             (:OBJECT =
                                      (A DEVELOP WITH (:INSTRUMENT = (A COMPUTER))))
                             (:INSTRUMENT = (A SYSTEM)))))
        (:OBJECT = (AN AFFECT))))
```

Abb.: Frame-Description for TIT-81

3.1 The frame description language

The formal description of SEMSYN's frame representation is as follows:

<IKBS-DESCR> : : = = (A <FRAME-NAME>)
(A <FRAME-NAME> WITH . <SLOT-FILLER-DESCRS>)
(THE <SLOT-NAME> FROM <IKBS-DESCR>)

<SLOT-FILLER-DESCRS> : : = = (<SLOT-NAME> = <IKBS-DESCR>)
...)

Conceptually we distinguish the following three main classes of frames:

1. Case schemata for verb concepts or actions (among these are all those frames that have case roles as slots).
2. Concept schemata for noun concepts or "picture producers".
3. Relation schemata - ENUMERATION, PURPOSE-, SCOPE-Relation etc.

Within this scope the repertoire of the semantic representation includes:

- "classical" case roles a la Fillmore (agent, object, method, instrument, goal ....)
- roles for the further specification of actions (manner, place, time ....)
- roles for the further specification of concepts (name, concern, specialize ....)
- ways to quantify and attribute concepts
- modality (e.g. not, possible ...).
- other semantic relations (SCOPE-relation, ISA-relation, PURPOSE-relation with roles :MEANS and :PURPOSE, PROPERTY-relation with roles :PROPERTY and :OWNER, ...)
- conjunctive and disjunctive ENUMERATION.

3.2 Knowledge bases during generation

SEMSYN's main generation phase may be viewed as communication between two knowledge bases: General knowledge about principal possibilities for realizing the semantic structures - the so called realization schemata - and specific knowledge mainly about diverse possibilities for lexicalization of semantic symbols.

The general knowledge about realization schemata is combined with the classes of the semantic representation. They decide - mainly by taking into account their actual roles or via globally specified stylistic preferences - which structures might be generated and how the fillers of the diverse roles take part in this generation.

3.3 Object-oriented implementation

The knowledge about realization schemata was implemented using the FLAVOR system of the LISP machine (Weinreb, Moon 1981). The classes of the realization schemata correspond to flavor classes. Realization schemata and the knowledge about the realization of roles are defined as flavor methods. This object-oriented architecture has shown to be very flexible. It supported experimenting with the system and its step-by-step improvement.

3.4 Realization schemata

Frame descriptions as used in SEMSYN are recursive structures. Therefore it is not astonishing that the control structure in generation is mainly of recursive type. In other words: the same decisions have to be redone on each level of embedding. In embedded frames of course some decisions are already restricted by the context.

What will be the syntactic form of the text generated for such a frame? At least for case schemata we have as first alternative the choice between the realization types :CLAUSE and :NG (noun group). For semantic structures from titles we used as default to generate a noun group (a toplevel case schema was lexicalised as noun). Only in a few cases we had titles that had to be generated as questions like "What is a model of ...?".

If the general syntactic form has been decided upon, there are more choices: a clause for example could be realized as an active or a passive clause. Within a noun group the attribute could be realized as a relative clause or in the form of a prepositional group.

These decisions are done with respect to several factors. One is the type of the actually filled roles. If a case schema for example has an :OBJECT, but no :AGENT, we prefer the passive construction in a clause realization. On the other hand stylistic preferences could be another factor. In the above case a preference could be to avoid passive, so we would take the realization schema "ACTIVE with an anonymous agent of 'man'".

In titles these preferences come from global switches. In real text they could come from the context.

```
  :NG as Title-Default:

Die Beeinflussung der Zuverlaessigkeit und der Qualitaet von Software mit einem
System zur Unterstuetzung der Entwicklung mit einem Computer.

  :NG with relative clause:

Die Beeinflussung der Zuverlaessigkeit und der Qualitaet von Software mit einem
System, mit dem die Entwicklung mit einem Computer unterstuetzt wird.

  :CLAUSE in passive voice:

Die Zuverlaessigkeit und die Qualitaet von Software wird mit einem System zur
Unterstuetzung der Entwicklung mit einem Computer beeinflusst.

  :CLAUSE with anonymous Agent:

Man beeinflusst die Zuverlaessigkeit und die Qualitaet von Software mit einem System
zur Unterstuetzung der Entwicklung mit einem Computer.


Abb.: Different Realisations for TIT-81
```

3.5 Role realizations

For frames without roles - the so called terminal structures the realization is more or less the lexicalisation of the semantic symbol. After this, process control and the produced IRS structure is given back to the surrounding frame or the toplevel.

If there are roles, there is some more work to be done. Some fillers of roles are realized as distinct structures of their own (mostly noun groups). They could be uttered for themselves.

Other roles only lead to changes in the IRS structure of their frame:
- decision about semantic features: fillers of a number role may e.g. lead to the pluralization of the noun group of the modified frame.
- creation of noun compounds as head of the actual nominal group: the filler of a :NAME role may become a prefix ("das SEMSYN-Projekt"). This holds as well for the terminal filler of a :SPECIALIZE role (variant: realization as an adjective). A negative :MODALITY could - in a noun group realization - lead to the prefix "Nicht-".

For those frames that have roles with realizations of their own this procedure recursively repeats for the frame descriptions of the fillers of those slots.

```
;;;*****************************************************************************
;;; ELEMENTS
;;;*****************************************************************************

(DEFMETHOD (concept-schema :realize-ELEMENTS-ROLE)
           (ROLE-FILLER)
  (SEND ROLE-FILLER :TRY-TO-REALIZE :NG))

(DEFMETHOD (concept-schema :HOW-TO-ADD-ELEMENTS-TO-IRS)
           (role-real irs &aux (pure-real (copytree role-real)))
  ;;;*****************************************************************
  ;;; z.B. NETZWERK AUS COMPUTERN
  (irs-join-or-overwrite :features '(:NUM PLURAL) pure-real)
  (irs-join-if-not-present :features '(:DET ZERO) pure-real)
  ;;;*****************************************************************
  (multiple-value-bind (prep case)
      (ge :elements-prep pure-real)
    (when case (irs-join-or-overwrite :features `(:kasus ,case) pure-real))
    (cond ((not-in-irs :possattr irs)
           (irs-join-as :POSSATTR `(:pg (:prep ,prep)(:pobj ,pure-real)) irs))
          (t (irs-add-to :QUALIFIERS `(:pg (:prep ,prep)(:pobj ,pure-real)) irs)))))

(defmethod (concept-schema :ELEMENTS-prep)
           (role-real)
  (first&rest (randomly-choose
                '((aus . dat)
                  (bestehend/ aus . dat)))))
```

Abb.: FLAVOR-Methods for the :ELEMENTS-Role

The knowledge about the realization of role fillers is combined with the slot names. For each decision we have two methods in the FLAVOR implementation: one decides if (and if 'yes', how) the filler of the role should be realized as separate structure or if

the meaning could be expressed in another way (cf. above). If a role filler will be explicitly realized it has to be decided how his IRS-structure shall be integrated in the overall structure (mostly as prepositional group) and which syntactic features could additionally be inferred.

3.6 The semantic to German lexicon SLEX

The specific knowledge about a semantic symbol (mainly about his lexicalisation) is stored within the semantic to German dictionary SLEX. Depending on which syntactic form should be realized and in which role a semantic symbol appears, lexicalisation may be desired as :NOUN, :VERB, :ADVERB or :ADJECTIVE . For these lexical categories there may be entries in SLEX.

Within our corpus of cir. 2000 titles the entries in these diverse categories were unique for most of the semantic symbols, lexicalisation within a category was not context dependent. For context dependent lexicalisation we have the additional possibility to have a so called 'lexical choice function' (LCF) combined with the semantic symbol.

Entries within the categories :NOUN or :VERB in SLEX are not necessarily single words. A semantic symbol may as well have a noun group or verb group as lexicalisation. Additionally SLEX entries may decide about the preposition that the chosen entry governs for the integration of role realizations.

```
DIES IST DER STAND VON: 10/11/85 12:22:40

#<EQ-HASH-TABLE 75525104> is a hash-table with 2582 entries ...

 *ENGLAND → ((:NOUN (ENGLAND (:DET ZERO) (:NUM SG)))
             (:SCHEMA-TYPE (GEOGRAPHIC-ENTITY)))
 ...
 ABSORB → ((:NOUN (ABSORBTION)) (:VERB (ABSORBIER)))
 ABSTRACT → ((:NOUN (ABSTRAKTION)) (:ADJ (ABSTRAKT)))
 ...
 ACCESS → ((:NOUN (ZUGANG (:OBJ-PREP (ZU . DAT)))
                  (ZUGRIFF (:OBJ-PREP (AUF . AKK))))
           (:VERB (ZUGREIF (:OBJ-PREP (AUF . AKK)))
             ((:VG (:VERB HAB)
                (:DIROBJ (:NG (:HEAD ZUGANG) (:FEATURES (:DET ZERO)))))
              (:OBJ-PREP (ZU . DAT))))
           (:SCHEMA-TYPE (OBJECT-AS-PREP-OBJ)))
 ...
 SCHEDULE → ((:NOUN (SCHEDULING (:DET DEF))) (:LCF <cf. LCFs>))
 ...
 SEMANTIC → ((:ADJ (SEMANTISCH)) (:NOUN (SEMANTIK)))
 SEMANTIC-NETWORK → ((:NOUN ((:NG (:HEAD NETZ) (:CLASSIFIER SEMANTISCH)))))
 ...
 SOLVE → ((:NOUN (LOESUNG)) (:VERB (LOES)
                            ((:VG (:VERB LOES) (:DIROBJ (:NG (:HEAD PROBLEM)))))))
           (:LCF <cf. LCFs> ))


ABB.: Excerpts from the semantic/german lexicon SLEX
```

```
for SCHEDULE:

(lambda (lex-type)
    (let ((specialize-slot (assq :specialize (me :role-descriptions))))
      (if (and (eq lex-type :noun)
               specialize-slot
               (equal (filler-descr specialize-slot) '(a SCHOOL)))
          (progn (me :copy-to-lexicalized :specialize)
                 (values 'STUNDENPLAN nil)))))

for SOLVE:

(lambda (lex-type)
    (let ((obj-slot (me :filler-descr :object)))
      (when (instance-descr-p obj-slot)
            (cond ((and (eq lex-type :noun)
                        (eq (semantic-symbol-from obj-slot) 'ROOT))
                   (values 'BERECHNUNG '((:det def) (:num sg))))
                  ((and (eq lex-type :verb)
                        (eq (semantic-symbol-from obj-slot) 'ROOT))
                   (values 'BERECHN nil))))))
```

Abb.: Exeaples of lexical-choice-functions (LCFs)

4. The morpho-syntactic generator-front-end

The generator-front-end SUTRA-S of the SEMSYN project may be used as black box: for a given IRS structure as input it produces the corresponding text in morphologically and syntactically correct German.

```
(:NG
 (:HEAD BEEINFLUSSUNG)
 (:FEATURES (:DET DEF) (:NUM SG))
 (:POSSATTR (:NG (:HEAD (:NG-CONJUNCT (:NGS (:NG (:HEAD ZUVERLAESSIGKEIT)
                                                 (:FEATURES (:NUM SG) (:DET DEF)))
                                            "und"
                                            (:NG (:HEAD QUALITAET)
                                                 (:FEATURES (:NUM SG) (:DET DEF))))))
                 (:FEATURES (:DET DEF) (:NUM PL))
                 (:POSSATTR (:PG (:PREP VON)
                                 (:POBJ (:NG (:HEAD SOFTWARE)
                                             (:FEATURES (:NUM SG) (:DET ZERO))))))))
 (:QUALIFIERS
  (:PG
   (:PREP MIT)
   (:POBJ
    (:NG
     (:HEAD SYSTEM)
     (:FEATURES (:DET INDEF) (:KASUS DAT))
     (:POSSATTR
      (:PG
       (:PREP ZU)
       (:POBJ
        (:NG
         (:HEAD UNTERSTUETZUNG)
         (:FEATURES (:NUM SG) (:DET DEF))
         (:POSSATTR
          (:NG (:HEAD ENTWICKLUNG)
               (:FEATURES (:DET DEF) (:NUM SG))
               (:QUALIFIERS (:PG (:PREP MIT)
                                 (:POBJ (:NG (:HEAD COMPUTER)
                                             (:FEATURES (:DET INDEF)
                                                        (:KASUS DAT))))))))))))))))
```

Abb.: IRS-Description for TIT-#1

SUTRA-S is an extension of the program SUTRA that has been developed by Busemann in the HAM-ANS project (Busemann, 1982). This module was rewritten in Zeta-LISP and extended for the purposes of SEMSYN (Emele & Momma, 1985):

- the repertoire of German surface structures was extended:
  - dynamic formation of noun compounds
  - handling of proper names, ordinal and cardinal numbers
  - coordinated noun groups
  - the position of constituents within a sentence may be chosen (e.g. used for focusing)
  - reflexive verbs, modal verbs, infinitive constructs
  - sentences with connectives etc.
- reorganisation of the German root form dictionary (currently cir. 3500 entries)
- menu based interface for lexicon maintenance and extension.

SUTRA-S performs all actions that are necessary to produce the German surface structure corresponding to a given IRS-structure. This includes all morphological tasks - formation of the correct forms of conjugation, declination and comparison - as well as syntactic tests (e.g. agreement of subject and verb) and decisions about the position of clause constituents (if not already decided by focus marking).

For all these tasks SUTRA-S has access to the following knowledge sources:
- German root form dictionary
- knowledge about the declination classes of German nouns
- knowledge about the conjugation classes of German verbs
- knowledge about the rules for building noun compounds
- knowledge about the relative position of sentence constituents.

5. Inferring of missing information

SEMSYN's generation module starts from a semantic representation that was designed to be language independent. For the primitives used - especially for the semantic relations expressed by the arcs in the semantic net - this may be true.

On the other hand the data delivered to us by FUJITSU are not really universal representations. The fact that the semantic nets are derived from Japanese is recognizable if one looks at the information that is not explicitly represented. This is true for all those semantic features that usually are not explicitly expressed in Japanese texts.

In Japanese number or definiteness of nouns or time of verbs normally is not expressed - correspondingly our data do not have semantic correlates for these features (except in the rare case when they have been expressed in the Japanese original). The Japanese reader infers the missing information from the context. In titles there is no such context available. For correct and acceptable German on the other hand we need determiners and our nouns need a number. Therefore we had to develop heuristics to reconstruct this information.

Some examples of such heuristics:

- a nominalized case frame has to be realized with definite article in singular ("Die Generierung natuerlicher Sprache").

- the :OBJECT role of a nominalized case frame should be realized indefinite and plural ("Die Generierung von Titeln"), except in cases with an exception information in SLEX ("Die Wartung von Software").

- concepts that have a :NAME role will be realized definite and singular ("Die Fourier-Transformation").

If no heuristic is applicable and if no SLEX information is found we use as title defaults 'indefinite' and 'singular' ("Ein Verfahren").

6. Concluding remarks

In contrast to the traditional transfer based approach to machine translation SEMSYN's approach with a semantic representation as "interlingua" proved to be advantageous:

- due to (far reaching) independence from the source language it was possible to let distinct and geographically separated groups develop the modules for analysis and synthesis

- the approach is inherently multilingual, texts in diverse languages might be generated from the semantic representation

- translation gets based upon the content of the original text, syntactic patterns of the source language will not constrain translation results.

Our current concern is to broaden the applicability of SEMSYN's generator for German: On the one hand we are experimenting to generate full texts, on the other hand we plan to allow for other semantic representations as well.

Acknowledgement

Thanks to my colleagues K. Hanakata, J. Laubsch, A. Lesniewski and Sh. Yokoyama, and to our students M. Emele, W. Kehl and S. Momma.

References

Busemann, S. (1982) "Probleme der automatischen Generierung deutscher Sprache" HAM-ANS Memo 8, Universitaet Hamburg

Emele, M. & St.Momma (1985) "SUTRA-S - Erweiterungen eines Generator-Front-Ends fuer das SEMSYN-Projekt", Studienarbeit, Inst. f. Informatik, Univ. Stuttgart

Laubsch. J., Roesner, D., Hanakata, K., Lesniewski, A. (1984) "Language Generation from Conceptual Structure: Synthesis of German in a Japanese/German MT Project"
in: COLING-84, Proceedings, Stanford

Uchida, H. & K. Sugiyama (1980) "A machine translation system from Japanese into English based on conceptual structure "
in: COLING-80, Proceedings, Tokyo, S. 455-462

Weinreb, D. & D. Moon "LISP machine manual", MIT, 1981