



**Deutsches
Forschungszentrum
für Künstliche
Intelligenz GmbH**

**Research
Report**
RR-91-07

**A Head-Driven Approach to
Incremental and Parallel Generation
of Syntactic Structures**

Günter Neumann Wolfgang Finkler

February 1991

**Deutsches Forschungszentrum für Künstliche Intelligenz
GmbH**

Postfach 20 80
D-6750 Kaiserslautern, FRG
Tel.: (+49 631) 205-3211/13
Fax: (+49 631) 205-3210

Stuhlsatzenhausweg 3
D-6600 Saarbrücken 11, FRG
Tel.: (+49 681) 302-5252
Fax: (+49 681) 302-5341

Deutsches Forschungszentrum für Künstliche Intelligenz

The German Research Center for Artificial Intelligence (Deutsches Forschungszentrum für Künstliche Intelligenz, DFKI) with sites in Kaiserslautern und Saarbrücken is a non-profit organization which was founded in 1988 by the shareholder companies ADV/Orga, AEG, IBM, Insiders, Fraunhofer Gesellschaft, GMD, Krupp-Atlas, Mannesmann-Kienzle, Nixdorf, Philips and Siemens. Research projects conducted at the DFKI are funded by the German Ministry for Research and Technology, by the shareholder companies, or by other industrial contracts.

The DFKI conducts application-oriented basic research in the field of artificial intelligence and other related subfields of computer science. The overall goal is to construct *systems with technical knowledge and common sense* which - by using AI methods - implement a problem solution for a selected application area. Currently, there are the following research areas at the DFKI:

- Intelligent Engineering Systems
- Intelligent User Interfaces
- Intelligent Communication Networks
- Intelligent Cooperative Systems.

The DFKI strives at making its research results available to the scientific community. There exist many contacts to domestic and foreign research institutions, both in academy and industry. The DFKI hosts technology transfer workshops for shareholders and other interested groups in order to inform about the current state of research.

From its beginning, the DFKI has provided an attractive working environment for AI researchers from Germany and from all over the world. The goal is to have a staff of about 100 researchers at the end of the building-up phase.

Prof. Dr. Gerhard Barth
Director

A Head-Driven Approach to Incremental and Parallel Generation of Syntactic Structures

Günter Neumann, Wolfgang Finkler

DFKI-RR-91-07

A version of this paper has been published in the Proceedings of the 13th International Conference on Computational Linguistics (COLING 90), 1990.

© Deutsches Forschungszentrum für Künstliche Intelligenz 1991

This work may not be copied or reproduced in whole or in part for any commercial purpose. Permission to copy in whole or in part without payment of fee is granted for nonprofit educational and research purposes provided that all such whole or partial copies include the following: a notice that such copying is by permission of Deutsches Forschungszentrum für Künstliche Intelligenz, Kaiserslautern, Federal Republic of Germany; an acknowledgement of the authors and individual contributors to the work; all applicable portions of this copyright notice. Copying, reproducing, or republishing for any other purpose shall require a licence with payment of fee to Deutsches Forschungszentrum für Künstliche Intelligenz.

A Head-Driven Approach to Incremental and Parallel Generation of Syntactic Structures

Günter Neumann* and Wolfgang Finkler

Authors' Abstract

This paper describes the construction of syntactic structures within an incremental multi-level and parallel generation system. Incremental and parallel generation imposes special requirements upon syntactic description and processing. A head-driven grammar represented in a unification-based formalism is introduced which satisfies these demands. Furthermore the basic mechanisms for the parallel processing of syntactic segments are presented.

Table of Contents

1 Introduction.....	2
2 Requirements upon the Syntactic Level.....	2
3 Overview of POPEL-HOW	4
4 The Design of the Syntactic Level.....	5
4.1 Representation	5
4.2 Incremental and Parallel Processing.....	7
4.3 Inflection, Linearization and Reformulation	10
5 Conclusion	11
Acknowledgements	12
References	12

* Günter Neumann, Institute for Computational Linguistics, University of Saarbrücken, Im Stadtwald 15, 6600 Saarbrücken 11, FRG

1 INTRODUCTION

Incremental generation (i.e. immediate verbalization of the parts of a stepwise computed conceptual structure - often called "message") is an important and efficient property of human language use ([DeSmedt&Kempen87], [Levelt89]). There are particular situations of communication (e.g. simultaneous descriptions of ongoing events) where incremental generation is necessary in order to ensure that new information can be verbalized in due time. As [DeSmedt& Kempen87] mentioned, incremental generation can be viewed as a *parallel* process: While the linguistic module (or "how-to-say" component) of an incremental generator is processing partial conceptual structures which are previously computed from the conceptual module (or "what-to-say" component) the latter can run simultaneously and add more conceptual elements. In [Finkler&Neumann89] we show that it is possible to refine this parallelism: Every conceptual or linguistic segment can be viewed as an active unit which tries to verbalize itself as fast and as independently as possible. If the translation of a segment is not possible because of unspecified but required information, it is necessary to request missing information in order not to jeopardize the fast mapping. As a consequence, the linguistic module provides *feedback* for the selection of what to say next by the conceptual module (cf. [Hovy87], [Reithinger88]).

Incremental generation imposes special requirements upon syntactic description and processing (cf. [Kempen87]). Until now only a few approaches to syntax have been developed which consider explicitly the requirements of incremental construction of sentences, namely that of [Kempen87] and [DeSmedt& Kempen88]. But those do not provide for a bidirectional flow of control between conceptual and linguistic module.

In this paper we describe the principles of representation and processing of syntactic knowledge in the natural language generation system POPEL-HOW, which for the first time combines explicitly incremental multi-level generation with parallelism and feedback. The syntactic knowledge is declaratively represented by a unification-based head-driven grammar that is linguistically based on dependency grammar.

2 REQUIREMENTS UPON THE SYNTACTIC LEVEL

The following aspects concerning the representation of syntactic knowledge and the basic mechanism have to be regarded in an incremental multi-level and parallel model:

1. The grammar should be *lexically based*. The lexicon is assumed to be the "essential mediator" between conceptualization and grammatical encoding (cf. [Levelt89]). During incremental generation it is not plausible to assume that the syntactic structure is built up "from phrases to lexical elements" starting with a root node S.

2. The grammar should support *vertical* rather than horizontal orientation [Kempen87]. The rules should emphasize the growing of syntactic structures by individual branches (ideally, only by one branch, but see 4.1). One should avoid that, because of adding a new segment, unnecessary constraints are simultaneously added for sister segments.

3. Syntactic segments should be expanded in the following three ways [Kempen87]: *upward* (a new segment B becomes the root node of a previous segment A), *downward* (B becomes a daughter node of A) and *insertion* (B becomes the daughter node of A which already has a daughter and this daughter becomes the daughter of B)¹.

4. During incremental generation one should not assume that the chronological order in which syntactic segments are attached corresponds to the linear order of the resulting utterance. Hence one should separate knowledge concerning immediate dominance and linear precedence [Kempen87]. Especially during the parallel processing of languages with a relatively free word order (e.g., German), one should avoid building up unnecessary syntactic paraphrases resulting from ordering variations.

5. When the whole syntactic structure of an utterance is built up in a parallel fashion, it should be possible to decide for every partial structure whether it is locally complete. In a head-driven grammar this is possible if segments are based on head elements. Thus only the information that is necessary for the inflection and linearization of the head is considered (see 4.2).

6. During spontaneous speech it may happen that already existing structures need to be modified because of new information that can only be considered by replacing old one (which means "reformulation"). Because of the dynamic behaviour of an incremental multi-level and parallel system, syntactic structures should not only cooperate but also compete during generation (see 4.3).

The basic demands 2., 3., 4. are put forward by Kempen's framework for syntactic tree formation [Kempen87]. They serve as a theoretical basis for the development of the syntactic formalism proposed in this paper. The other aspects are explicitly mentioned because they are important for incremental multi-level and parallel generation. Before the formalism and the basic operations are described in more detail we briefly introduce POPEL-HOW, the generator in which the formalism is embedded.

¹In our formalism we need only downward and upward expansion because of the nature of the syntactic structures.

3 OVERVIEW OF POPEL-HOW

POPEL-HOW [Finkler&Neumann89] is the how-to-say component of the natural language generation system POPEL [Reithinger88]. The main features of POPEL-HOW are:

Incremental generation: POPEL-WHAT (the what-to-say component) immediately passes *segments* of the conceptual structure as input to POPEL-HOW when they are considered relevant for the contents to be produced. POPEL-HOW tries to build up the corresponding semantic and syntactic segments immediately, too, in order to utter the input segments as fast as possible, i.e. POPEL-HOW generates structures at *each* level in an incremental way.

Feedback: It is possible that new conceptual segments cannot be uttered directly because they lack necessary linguistic information. In this case POPEL-HOW is able to interact with POPEL-WHAT to demand the missing information. The bidirectional flow of control has two main advantages: Firstly, the determination of the contents can be done on the basis of conceptual considerations only. Therefore, POPEL-HOW is *flexible* enough to handle underspecified input². Secondly, POPEL-WHAT has to regard feedback from POPEL-HOW during the computation of the further selection process. This means, an incremental system like POPEL can model the influence of linguistical restrictions on the process that determines what to say next (cf. [Hovy87], [Reithinger88]).

Reformulations: The addition of new conceptual segments could result in constructing semantic or syntactic segments that stand in competition with previously computed segments. POPEL-HOW is able to handle such reformulations.

Unified parallel processing: Every segment at each level is conceived as an active and independent process. This view is the foundation of the parallel and incremental generation with feedback at each level of POPEL-HOW. The processing approach is uniform since every process (either conceptual, semantic or syntactic) runs the same basic program and processes its segment in the same way.

²[Ward88] shows that most generation systems lack this ability so that their inputs have been tailored to determine a good sentence.

4 THE DESIGN OF THE SYNTACTIC LEVEL

The syntactic level in POPEL-HOW is divided into two sublevels: the dependency-based level (DBS-level) and the level of inflected and linearized structures (ILS-level). At the DBS-level the syntactic structure is constructed in an incremental and parallel way. At the ILS-level there exists one process which performs inflection and linearization for every incoming segment of the DBS-level.

4.1 REPRESENTATION

The central knowledge source of both sublevels is POPELGRAM, a unification-based head driven grammar. The grammar is declaratively expressed in a modified version of the PATR formalism (cf. [Shieber85]). PATR is modified in the sense that the representation and derivation devices are separated. To use it for generation, the parser (i.e. the derivation device) is replaced by operations suitable for incremental and parallel generation³.

POPELGRAM is divided into immediate dominance (ID) and linear precedence (LP) structures⁴. In order to allow for vertical orientation, the ID-structures are furthermore divided into those that describe basic syntactic segments and others that describe the relationship between complex structures. Basic segments are phrases that consist of a (non-empty) set of constituents. At least one constituent must have a lexical category. The central point of view for the description of basic segments is that the *dependency relation* is defined for the constituents of a phrase: One constituent that must have a lexical category is denoted as the head of a phrase. All other constituents are denoted as complements that are immediately dependent on the head. The dependency relation of the constituents of a phrase is expressed as a feature description named STRUCT which belongs to the phrasal category. E. g., fig. 1 shows the feature set of the basic segment for constructing a sentence phrase where the head governs two complements⁵:

³This is in contrast to [Shieber et al.89] where the same operations are used for both analysis and generation.

⁴In this paper we only consider the ID-structures in more detail.

⁵The integer attributes correspond to the order of the elements in the corresponding ID-rule $S \Rightarrow V, NP1, NP2$. Note: The order of constituents is not assumed to be the order of the corresponding surface string.

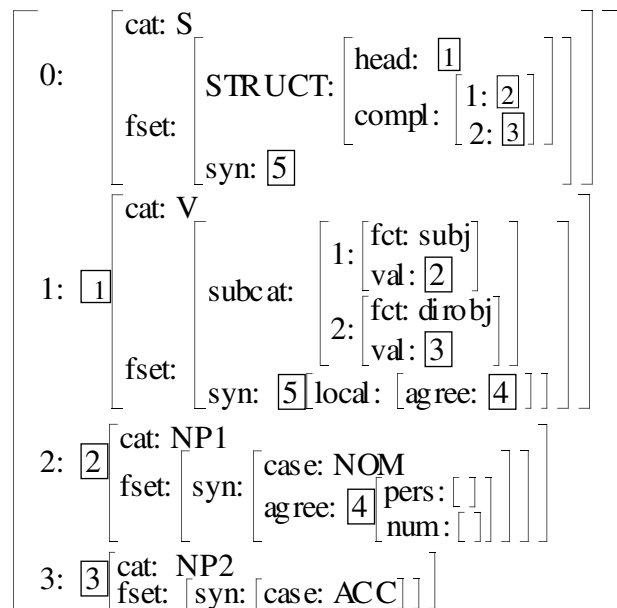


Fig. 1

The head element is the central element of the phrase and determines the characteristic properties of the whole phrase that is defined as the projection of its head⁶. The particular quality of ID-structures of POPELGRAM makes it possible to interpret such structures as theoretically based on dependency grammar (cf. [Kunze75], [Hellwig86]). Basic segments can also be defined as abstract descriptions of certain classes of lexical elements which have the same category and valence (or *subcategorization*) restrictions. The abstract description of lexical information (in the sense of dependency grammar) is the foundation of our lexically based grammar. We assume that this view supports incremental and parallel generation.

Obviously, basic segments build up syntactic constraints for their complements. Although this seems to emphasize horizontal orientation, these constraints are not redundant and therefore do not violate the view of vertical orientation. A basic segment must access the information that is necessary to build up a minimal syntactically well-formed partial utterance. If the dependencies between a head element and its modifiers are strong as in the case of complements, then this information has to be formulated in an incremental grammar, too⁷.

⁶If complements of basic segments are defined as phrases, then the head elements of the complements are immediately dependent on the head element of the basic segment, because of the projection principle. Hence, complex syntactic structures are compositionally viewed as hierarchical head and modifier structures.

⁷This point of view is in contrast to [Kempen87]. In his framework, all syntactic segments are defined as node-arc-node structures because segments have to branch only by one element at a time.

Adjuncts are not necessary for building minimal well-formed structures. Therefore they are not part of basic segments. The combination of basic segments and adjuncts is expressed by means of complex segments. Those ID-structures are applied only when the corresponding basic segment already exists.

4.2 INCREMENTAL AND PARALLEL PROCESSING

ID-structures are processed at the DBS-level. At this level basic segments are considered as independent and active units. In order to realize this every basic segment is associated with its own active process (or *object*) whereby the structure of an object reflects the underlying dependency relation of the segment. For the feature set of fig. 1 this can graphically be represented as follows:

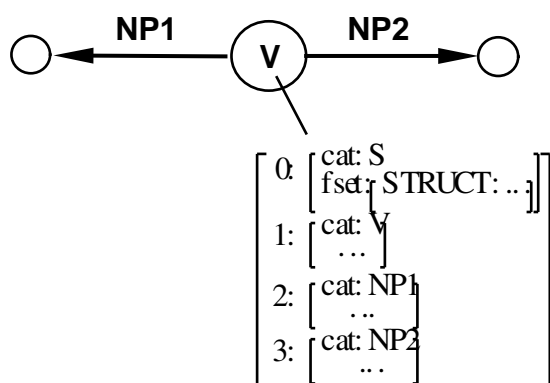


Fig. 2

\textcircled{V} is denoted as the *body* and the directed labeled slots $\textcircled{\leftarrow} \xrightarrow{\text{NP1}}$ and $\xrightarrow{\text{NP2}} \textcircled{\rightarrow}$ as the *context* of the process. The names of the body and the slot correspond to the feature sets of the associated basic segment. Every labeled slot serves as a communication link to a process which represents the basic segment of the corresponding slot's complement. The topology of the network of active processes represents the corresponding dependency-based tree structure.

Objects at the DBS-level are created during the evaluation of local transition rules. These rules describe the relation between semantic and syntactic knowledge in a distributed way [Finkler&Neumann89]. They are local because every rule describes the mapping of semantic segments (represented as concepts of a network-like formalism) to syntactic

segments of the ID-part of POPELGRAM. In principle this local mapping is case-frame based: A semantic head (e.g., a predicate) and its deep cases (e.g., agent or benefactive) are related to a corresponding syntactic head (e.g. a verb) and its syntactic frame (e.g., subject or direct object constituents). During the mapping lexical material which is determined through the choice of words⁸ directs and restricts the choice of possible syntactic structures.

In order to construct syntactic structures in an incremental and parallel way, every DBS-object has to solve the following two central subtasks:

- a. building up connections to other objects
- b. mapping itself as fast as possible into the next level

In order to solve task a., every new created DBS-object tries to integrate itself into the topology of the already existing objects. Thereby the following cases have to be distinguished:

1. The new object is taken up into the context of an object. Syntactically, this means that the new object represents an immediately dependent constituent. The existing structure is expanded downward (see fig. 3).

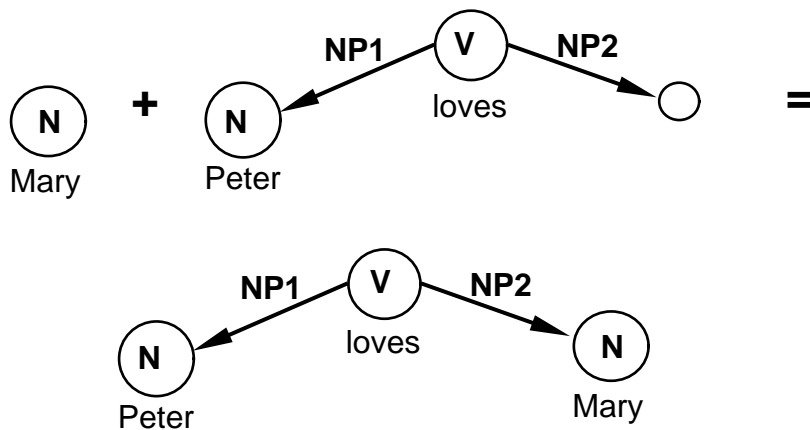


Fig. 3:
The new active object is the left one.

⁸The choice of words is performed in two steps in POPEL-HOW. While conceptual segments are translated into semantic segments, the *content words* - e.g. verbs and nouns - are selected. The selection of *function words* - e.g. prepositions which depend on the meaning of the verb - is performed during the mapping between the semantic level and the DBS-level.

2. The new object takes up (or binds) an already existing object. The existing structure is expanded upward (see fig. 4).

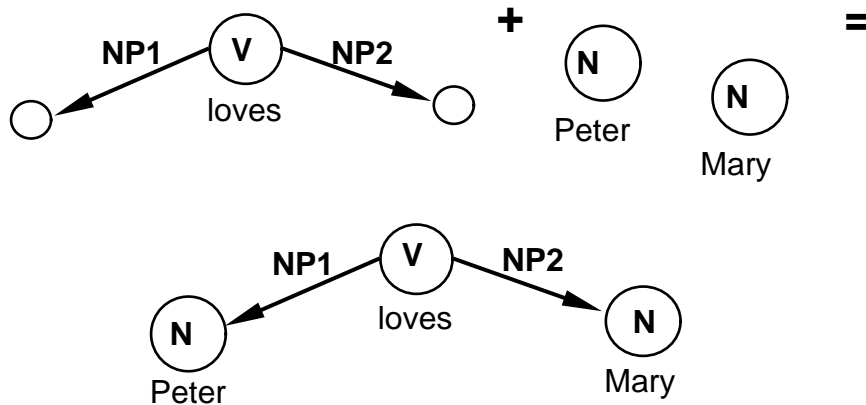


Fig. 4:
In this example the new verbal object binds two existing nouns.

3. The new object describes a reformulation of the basic segment of an already existing object. This object must be replaced with the new one (see fig. 5).

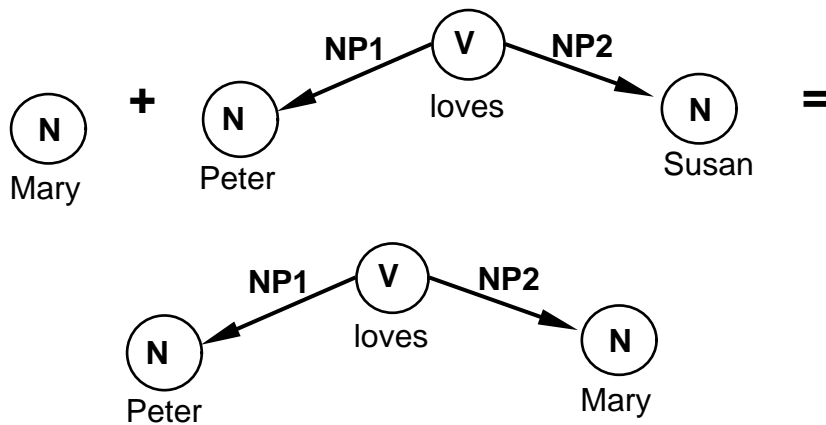


Fig. 5:
This is actually uttered as:
"Peter loves Susan ... uh ... Mary."

While new communication links are built up, participating objects exchange their syntactic features⁹. An adjunct is added to an object by augmenting the object's context with a new labeled slot. When the slot is being filled with an object, its associated feature set and the basic segment are combined to yield a complex segment that represents the new segment of the augmented object.

Every DBS-object tries to map itself into the next level as fast as possible (subtask b.) in order to facilitate spontaneous speech. The feature set of the head of the associated segment is checked to see if it contains sufficient information for inflecting and linearizing it. For every segment this information has to be specified under the head's feature *local*. Actual values come either from the lexicon, e.g., gender for nouns, from dependent elements by means of structure sharing, e.g., person and number for a verb from the subject constituent or from a governing constituent, e.g., case for dependent nouns of a verb.

When all subfeatures of the *local* feature have values, i.e. a value other than the empty feature set, then the segment is said to be *locally complete*. The reason for the segment of a DBS-object not to be locally complete is that DBS-objects that share values with the *local* feature set are either not yet created or not yet locally complete themselves (e.g., a verb is locally incomplete when a phrase that has to fill the subject role is not yet created). In the first case, the missing objects are requested (using the object's context) from that object of the semantic level above that has created the requesting object. In the second case, DBS-objects that are already connected to the underspecified object are invited to determine the missing values.

4.3 INFLECTION, LINEARIZATION AND REFORMULATION

Segments from the DBS-level that are locally complete are inflected and linearized at the ILS-level in POPEL-HOW. The inflection is performed by means of the package MORPHIX [Finkler&Neumann88], which handles most of the inflection phenomena of German. The order of segments in an utterance is syntactically constrained by the LP-part of POPELGRAM. It is assumed that the order of activation of conceptual segments (which is

⁹ This is performed with the help of a search algorithm using the ID-structures of POPELGRAM. The ID-structures implicitly represent the search space. To constraint the space, heuristic information in form of partial feature descriptions is used. This search operation serves as the basis for other operations, too, e.g. for unifying lexical elements with basic segments.

determined using pragmatical knowledge (cf. [Reithinger88])) should be maintained if it is syntactically wellformed; otherwise the segments are reordered by means of relevant LP-structures. Therefore, the order of the input segments affects the variations of the linear order in the resulting utterance, which makes POPEL-HOW more flexible (e.g. by deciding whether to realize an active or passive form). But the order of the input segments can affect lexical choice, too. For example, if the modifier denoting the vehicle (e.g., "airplane") for a conceptual segment of type "commute-by-public-transportation" is known in due time the more restricted verb "to fly" is chosen instead of the more general verb "to go". Otherwise, a prepositional phrase realizing the modifier must be verbalized explicitly (e.g., "to go by airplane").

It is possible that because of the incremental composition of large structures across several levels new conceptual segments lead to the *reformulation* of structures at successive levels (cf. [DeSmedt&Kempen88]) as indicated in the following sample utterances:

"Mary is going ... uh ... Mary and Peter are going to school."

"Eric goes ... uh ... flies to the USA."

In POPEL-HOW the reformulation of a segment is represented by an object that is in competition to the corresponding already existing object (e.g., the first utterance where the new conceptual segment "Peter" leads to a syntactic segment "Mary and Peter" which is in competition with the noun phrase "Mary"). In order to integrate such new objects the connections to an object which is to be replaced must be broken up and reconnected to the new object (see also fig. 5). Of course, the replacement must be performed for the relevant associated segments, too. For syntactic segments we have developed an operation which allows the replacement of parts of a given feature set. In principle this is realized by resetting the corresponding partial feature set (i.e., relevant features get the empty value) and subsequently unifying it with the new information.

5 CONCLUSION

This paper has demonstrated the representation and processing of syntactic knowledge within the generation system POPEL-HOW which for the first time combines explicitly incremental sentence production with parallelism and feedback. Such a generation model imposes special requirements on syntactic representation and processing. A unification-based

head-driven grammar, linguistically based on dependency grammar, is introduced that satisfies these demands. Furthermore the basic mechanisms for incremental and parallel generation have been presented.

The whole generation system is implemented in Commonlisp on a Symbolics 3640 lisp-machine by simulated parallelism using the process-facilities and the scheduler. It also runs in Kyoto Commonlisp and CLOS (using a selfwritten scheduler). The system is fully integrated in the natural language access system XTRA (cf. [Allgayer et al.89]).

ACKNOWLEDGEMENTS

Major parts of the work presented in this paper were developed during the master's thesis of the authors. This work was supported by the German Science Foundation (DFG) in its Special Collaborative Program on AI and Knowledge-Based Systems (SFB 314), project XTRA.

Thanks to Gregor Erbach, Norbert Reithinger and Harald Trost for their helpful comments on earlier versions of the paper.

REFERENCES

[Allgayer et al.89] J. **Allgayer**, R. **Jansen-Winkeln**, C. **Reddig** and N. **Reithinger**. *Bidirectional use of knowledge in the multi-modal NL access system XTRA*: In: Proceedings of the 11th IJCAI, pp. 1492 - 1497, Detroit, Michigan USA, 1989.

[DeSmedt&Kempen87] K. **DeSmedt** and G. **Kempen**. *Incremental Sentence Production, Self-Correction and Coordination*. In: G. Kempen (ed.), *Natural Language Generation: New Results in Artificial Intelligence, Psychology and Linguistics*, pp. 365-376, Dordrecht: Martinus Nijhoff, 1987.

[DeSmedt&Kempen88] K. **DeSmedt** and G. **Kempen**. *The Representation of Grammatical Knowledge in a Model for Incremental Sentence Generation*. Paper presented at the 4th IWG, Santa Catalina Island, 7 1988.

[Finkler&Neumann88] W. **Finkler** and G. **Neumann**. *MORPHIX: A Fast Realization of a Classification-Based Approach to Morphology*. In: Proceedings of the WWWS, Springer, Berlin, 1988.

[Finkler&Neumann89] W. **Finkler** and G. **Neumann**. *POPEL-HOW: A Distributed Parallel Model for Incremental Natural Language Production with Feedback*. In: Proceedings of the 11th IJCAI, pp. 1518 - 1523, Detroit, Michigan USA, 1989.

[Hellwig86] P. **Hellwig**. *Dependency Unification Grammar*. In: Proceedings of the 11th COLING, Bonn, FRG, 1986.

[Hovy87] E. **Hovy**. *Generating Natural Language Under Pragmatic Constraints*. Ph.D. thesis, Yale University, 1987.

[Kempen87] G. **Kempen**. *A Framework for Incremental Syntactic Tree Formation*. In: Proceedings of the 10th IJCAI, Mailand, Italy, 1987.

[Kunze75] J. **Kunze**. *Abhängigkeits-grammatik*. Akademie-Verlag, Berlin, 1975.

[Levelt89] W.J.M. **Levelt**. *Speaking: From Intention to Articulation*. Massachusetts Institute of Technology: The MIT Press, 1989.

[Reithinger88] N. **Reithinger**. *POPEL: A Parallel and Incremental Natural Language Generation System*. Paper presented at the 4th IWG, Santa Catalina Island, 7 1988.

[Shieber85] S.M. **Shieber**. *An Introduction to Unification-Based Approaches to Grammar*. Volume 4 of CSLI Lecture Notes, CLSI, Stanford, California, 1985.

[Shieber et al.89] S.M. **Shieber**, G. van **Noord**, R.M. **Moore** and F.C.P. **Pereira**. *A Semantic Head-Driven Generation Algorithm for Unification-Based Formalisms*. In: Proceedings of the 27th ACL, Vancouver, British Columbia, Canada, 1989.

[Ward88] N. **Ward**. *Issues in Word Choice*. In: Proceedings of the 12th COLING, Budapest, Hungary, 1988.



**Deutsches
Forschungszentrum
für Künstliche
Intelligenz GmbH**

DFKI
-Bibliothek-
PF 2080
67608 Kaiserslautern
FRG

DFKI Publikationen

Die folgenden DFKI Veröffentlichungen sowie die aktuelle Liste von allen bisher erschienenen Publikationen können von der oben angegebenen Adresse oder per anonymem ftp von ftp.dfki.uni-kl.de (131.246.241.100) unter pub/Publications bezogen werden.

Die Berichte werden, wenn nicht anders gekennzeichnet, kostenlos abgegeben.

DFKI Research Reports

RR-92-43

Christoph Klauck, Jakob Mauss: A Heuristic driven Parser for Attributed Node Labeled Graph Grammars and its Application to Feature Recognition in CIM
17 pages

RR-92-44

Thomas Rist, Elisabeth André: Incorporating Graphics Design and Realization into the Multimodal Presentation System WIP
15 pages

RR-92-45

Elisabeth André, Thomas Rist: The Design of Illustrated Documents as a Planning Task
21 pages

RR-92-46

Elisabeth André, Wolfgang Finkler, Winfried Graf, Thomas Rist, Anne Schauder, Wolfgang Wahlster: WIP: The Automatic Synthesis of Multimodal Presentations
19 pages

RR-92-47

Frank Bomarius: A Multi-Agent Approach towards Modeling Urban Traffic Scenarios
24 pages

RR-92-48

Bernhard Nebel, Jana Koehler: Plan Modifications versus Plan Generation: A Complexity-Theoretic Perspective
15 pages

RR-92-49

Christoph Klauck, Ralf Legleitner, Ansgar Bernardi: Heuristic Classification for Automated CAPP
15 pages

DFKI Publications

The following DFKI publications or the list of all published papers so far are obtainable from the above address or per anonymous ftp from ftp.dfki.uni-kl.de (131.246.241.100) under pub/Publications.

The reports are distributed free of charge except if otherwise indicated.

RR-92-50

Stephan Busemann:
Generierung natürlicher Sprache
61 Seiten

RR-92-51

Hans-Jürgen Bürckert, Werner Nutt:
On Abduction and Answer Generation through Constrained Resolution
20 pages

RR-92-52

Mathias Bauer, Susanne Biundo, Dietmar Dengler, Jana Koehler, Gabriele Paul: PHI - A Logic-Based Tool for Intelligent Help Systems
14 pages

RR-92-53

Werner Stephan, Susanne Biundo:
A New Logical Framework for Deductive Planning
15 pages

RR-92-54

Harold Boley: A Direkt Semantic Characterization of RELFUN
30 pages

RR-92-55

John Nerbonne, Joachim Laubsch, Abdel Kader Diagne, Stephan Oepen: Natural Language Semantics and Compiler Technology
17 pages

RR-92-56

Armin Laux: Integrating a Modal Logic of Knowledge into Terminological Logics
34 pages

RR-92-58

Franz Baader, Bernhard Hollunder:
How to Prefer More Specific Defaults in Terminological Default Logic
31 pages

RR-92-59

Karl Schlechta and David Makinson: On Principles and Problems of Defeasible Inheritance
13 pages

RR-92-60

Karl Schlechta: Defaults, Preorder Semantics and Circumscription
19 pages

RR-93-02

Wolfgang Wahlster, Elisabeth André, Wolfgang Finkler, Hans-Jürgen Profitlich, Thomas Rist: Plan-based Integration of Natural Language and Graphics Generation
50 pages

RR-93-03

Franz Baader, Berhard Hollunder, Bernhard Nebel, Hans-Jürgen Profitlich, Enrico Franconi: An Empirical Analysis of Optimization Techniques for Terminological Representation Systems
28 pages

RR-93-04

Christoph Klauck, Johannes Schwagereit: GGD: Graph Grammar Developer for features in CAD/CAM
13 pages

RR-93-05

Franz Baader, Klaus Schulz: Combination Techniques and Decision Problems for Disunification
29 pages

RR-93-06

Hans-Jürgen Bürckert, Bernhard Hollunder, Armin Laux: On Skolemization in Constrained Logics
40 pages

RR-93-07

Hans-Jürgen Bürckert, Bernhard Hollunder, Armin Laux: Concept Logics with Function Symbols
36 pages

RR-93-08

Harold Boley, Philipp Hanschke, Knut Hinkelmann, Manfred Meyer: COLAB: A Hybrid Knowledge Representation and Compilation Laboratory
64 pages

RR-93-09

Philipp Hanschke, Jörg Würtz: Satisfiability of the Smallest Binary Program
8 Seiten

RR-93-10

Martin Buchheit, Francesco M. Donini, Andrea Schaerf: Decidable Reasoning in Terminological Knowledge Representation Systems
35 pages

RR-93-11

Bernhard Nebel, Hans-Juergen Buerckert: Reasoning about Temporal Relations: A Maximal Tractable Subclass of Allen's Interval Algebra
28 pages

RR-93-12

Pierre Sablayrolles: A Two-Level Semantics for French Expressions of Motion
51 pages

RR-93-13

Franz Baader, Karl Schlechta: A Semantics for Open Normal Defaults via a Modified Preferential Approach
25 pages

RR-93-14

Joachim Niehren, Andreas Podelski, Ralf Treinen: Equational and Membership Constraints for Infinite Trees
33 pages

RR-93-15

Frank Berger, Thomas Fehrlé, Kristof Klöckner, Volker Schölles, Markus A. Thies, Wolfgang Wahlster: PLUS - Plan-based User Support Final Project Report
33 pages

RR-93-16

Gert Smolka, Martin Henz, Jörg Würtz: Object-Oriented Concurrent Constraint Programming in Oz
17 pages

RR-93-17

Rolf Backofen: Regular Path Expressions in Feature Logic
37 pages

RR-93-18

Klaus Schild: Terminological Cycles and the Propositional μ -Calculus
32 pages

RR-93-20

Franz Baader, Bernhard Hollunder: Embedding Defaults into Terminological Knowledge Representation Formalisms
34 pages

RR-93-22

Manfred Meyer, Jörg Müller: Weak Looking-Ahead and its Application in Computer-Aided Process Planning
17 pages

RR-93-23

Andreas Dengel, Ottmar Lutz: Comparative Study of Connectionist Simulators
20 pages

RR-93-24

Rainer Hoch, Andreas Dengel:
Document Highlighting —
Message Classification in Printed Business Letters
17 pages

RR-93-25

Klaus Fischer, Norbert Kuhn: A DAI Approach to
Modeling the Transportation Domain
93 pages

RR-93-26

Jörg P. Müller, Markus Pischel: The Agent
Architecture InteRRaP: Concept and Application
99 pages

RR-93-27

Hans-Ulrich Krieger:
Derivation Without Lexical Rules
33 pages

RR-93-28

*Hans-Ulrich Krieger, John Nerbonne,
Hannes Pirker:* Feature-Based Allomorphy
8 pages

RR-93-29

Armin Laux: Representing Belief in Multi-Agent
Worlds via Terminological Logics
35 pages

RR-93-33

Bernhard Nebel, Jana Koehler:
Plan Reuse versus Plan Generation: A Theoretical
and Empirical Analysis
33 pages

RR-93-34

Wolfgang Wahlster:
Verbmobil Translation of Face-To-Face Dialogs
10 pages

RR-93-35

Harold Boley, François Bry, Ulrich Geske (Eds.):
Neuere Entwicklungen der deklarativen KI-
Programmierung — *Proceedings*
150 Seiten

Note: This document is available only for a
nominal charge of 25 DM (or 15 US-\$).

RR-93-36

*Michael M. Richter, Bernd Bachmann, Ansgar
Bernardi, Christoph Klauck, Ralf Legleitner,
Gabriele Schmidt:* Von IDA bis IMCOD:
Expertensysteme im CIM-Umfeld
13 Seiten

RR-93-38

Stephan Baumann: Document Recognition of
Printed Scores and Transformation into MIDI
24 pages

DFKI Technical Memos**TM-91-14**

Rainer Bleisinger, Rainer Hoch, Andreas Dengel:
ODA-based modeling for document analysis
14 pages

TM-91-15

Stefan Busemann: Prototypical Concept Formation
An Alternative Approach to Knowledge Representation
28 pages

TM-92-01

Lijuan Zhang: Entwurf und Implementierung eines
Compilers zur Transformation von
Werkstückrepräsentationen
34 Seiten

TM-92-02

Achim Schupeta: Organizing Communication and
Introspection in a Multi-Agent Blocksworld
32 pages

TM-92-03

Mona Singh:
A Cognitive Analysis of Event Structure
21 pages

TM-92-04

*Jürgen Müller, Jörg Müller, Markus Pischel,
Ralf Scheidhauer:*
On the Representation of Temporal Knowledge
61 pages

TM-92-05

Franz Schmalhofer, Christoph Globig, Jörg Thoben:
The refitting of plans by a human expert
10 pages

TM-92-06

Otto Kühn, Franz Schmalhofer: Hierarchical
skeletal plan refinement: Task- and inference
structures
14 pages

TM-92-08

Anne Kilger: Realization of Tree Adjoining
Grammars with Unification
27 pages

TM-93-01

Otto Kühn, Andreas Birk: Reconstructive
Integrated Explanation of Lathe Production Plans
20 pages

TM-93-02

Pierre Sablayrolles, Achim Schupeta:
Conflict Resolving Negotiation for COoperative
Schedule Management
21 pages

TM-93-03

Harold Boley, Ulrich Buhrmann, Christof Kremer:
Konzeption einer deklarativen Wissensbasis über
recyclingrelevante Materialien
11 pages

DFKI Documents**D-92-21**

Anne Schauder: Incremental Syntactic Generation of Natural Language with Tree Adjoining Grammars
57 pages

D-92-22

Werner Stein: Indexing Principles for Relational Languages Applied to PROLOG Code Generation
80 pages

D-92-23

Michael Herfert: Parsen und Generieren der Prolog-artigen Syntax von RELFUN
51 Seiten

D-92-24

Jürgen Müller, Donald Steiner (Hrsg.): Kooperierende Agenten
78 Seiten

D-92-25

Martin Buchheit: Klassische Kommunikations- und Koordinationsmodelle
31 Seiten

D-92-26

Enno Tolzmann: Realisierung eines Werkzeugauswahlmoduls mit Hilfe des Constraint-Systems CONTAX
28 Seiten

D-92-27

Martin Harm, Knut Hinkelmann, Thomas Labisch: Integrating Top-down and Bottom-up Reasoning in COLAB
40 pages

D-92-28

Klaus-Peter Gores, Rainer Bleisinger: Ein Modell zur Repräsentation von Nachrichtentypen
56 Seiten

D-93-01

Philipp Hanschke, Thom Frühwirth: Terminological Reasoning with Constraint Handling Rules
12 pages

D-93-02

Gabriele Schmidt, Frank Peters, Gernod Laufkötter: User Manual of COKAM+
23 pages

D-93-03

Stephan Busemann, Karin Harbusch(Eds.): DFKI Workshop on Natural Language Systems: Reusability and Modularity - Proceedings
74 pages

D-93-04

DFKI Wissenschaftlich-Technischer Jahresbericht 1992
194 Seiten

D-93-05

Elisabeth André, Winfried Graf, Jochen Heinsohn, Bernhard Nebel, Hans-Jürgen Profitlich, Thomas Rist, Wolfgang Wahlster: PPP: Personalized Plan-Based Presenter
70 pages

D-93-06

Jürgen Müller (Hrsg.): Beiträge zum Gründungsworkshop der Fachgruppe Verteilte Künstliche Intelligenz Saarbrücken 29.-30. April 1993
235 Seiten
Note: This document is available only for a nominal charge of 25 DM (or 15 US-\$).

D-93-07

Klaus-Peter Gores, Rainer Bleisinger: Ein erwartungsgesteuerter Koordinator zur partiellen Textanalyse
53 Seiten

D-93-08

Thomas Kieninger, Rainer Hoch: Ein Generator mit Anfragesystem für strukturierte Wörterbücher zur Unterstützung von Texterkennung und Textanalyse
125 Seiten

D-93-09

Hans-Ulrich Krieger, Ulrich Schäfer: TDL ExtraLight User's Guide
35 pages

D-93-10

Elizabeth Hinkelman, Markus Vonerden, Christoph Jung: Natural Language Software Registry (Second Edition)
174 pages

D-93-11

Knut Hinkelmann, Armin Laux (Eds.): DFKI Workshop on Knowledge Representation Techniques — Proceedings
88 pages

D-93-12

Harold Boley, Klaus Elsbernd, Michael Herfert, Michael Sintek, Werner Stein: RELFUN Guide: Programming with Relations and Functions Made Easy
86 pages

D-93-14

Manfred Meyer (Ed.): Constraint Processing – Proceedings of the International Workshop at CSAM'93, July 20-21, 1993
264 pages
Note: This document is available only for a nominal charge of 25 DM (or 15 US-\$).