

# FSS-WASTL

## Interactive Knowledge Acquisition for a Semantic Lexicon

Roman M. Jansen-Winkel      Alassane Ndiaye      Norbert Reithinger \*

Universität des Saarlandes  
D-6600 Saarbrücken 11  
Federal Republic of Germany  
{roman, ndiaye, bert}@cs.uni-sb.de

### Abstract

The following paper describes FSS-WASTL, a system for the acquisition of semantic knowledge within XTRA – a natural language access system to expert systems. Starting from user-supplied example utterances, hypotheses for lexical entries are induced, which then are refined top down through interactive classification. The refinement process of FSS-WASTL is a variant of the well-known KL-ONE classification procedure and is an extension of T. Finin's interactive classifier. The integration of XTRA's natural language generator, which is a highlight of FSS-WASTL, allows for the paraphrasing of the user's input for clarification purposes. FSS-WASTL is compared with other research systems. It is argued that this approach can be transferred to applications beyond the natural language domain.

## 1 Introduction

A key issue during the development and the application of knowledge-based systems is maintaining and extending their knowledge bases. Hence for adaptable systems, such as transportable natural language systems or expert system shells, there arises the problem of applying these systems to new application areas. Providing special knowledge acquisition tools has proved useful.

We present a tool for KL-ONE-like knowledge representation languages named *interactive classifier*. In the natural language system XTRA it is applied to acquire new semantic knowledge. This module called FSS-WASTL infers new lexical entries from exemplary user input. The interactive classifier provides a comfortable interface with the ability to explain its various reasoning steps. However, the highlight of the interface is the integration of XTRA's natural language generator, which allows for the paraphrasing of the user's input for clarification purposes.

After a short survey of the XTRA system, the interactive classification algorithm will be presented. Then the application of the interactive classifier is shown, and the paraphraser will be explained in particular. The last section refers to similar approaches and gives development perspectives beyond the state of the art.

---

\*This research has been supported by the German Science Foundation in its Special Collaborative Research Programme on Artificial Intelligence and Knowledge Based Systems (SFB 314).  
To appear in "Ardizzone E., S. Gaglio and F. Sorbello, eds: Trends in Artificial Intelligence, Proceedings of the second AI\*IA 1991, Lecture Notes on Artificial Intelligence 529, Springer Verlag."

## 2 An Overview of the XTRA System

The XTRA access system to expert systems [Allgayer et al., 89a; Allgayer et al., 89b] is aimed at rendering the interaction with expert systems easier for inexperienced users. XTRA communicates with the user in a natural language (German), extracts data relevant to the expert system from her/his natural language input, answers user queries as to terminology, and provides user-accommodated natural language verbalizations of results and explanations provided by the expert system. Special emphasis is placed upon the separation of linguistic and world knowledge, the bidirectional use of knowledge sources, and the combination of pointing gestures and written natural language.

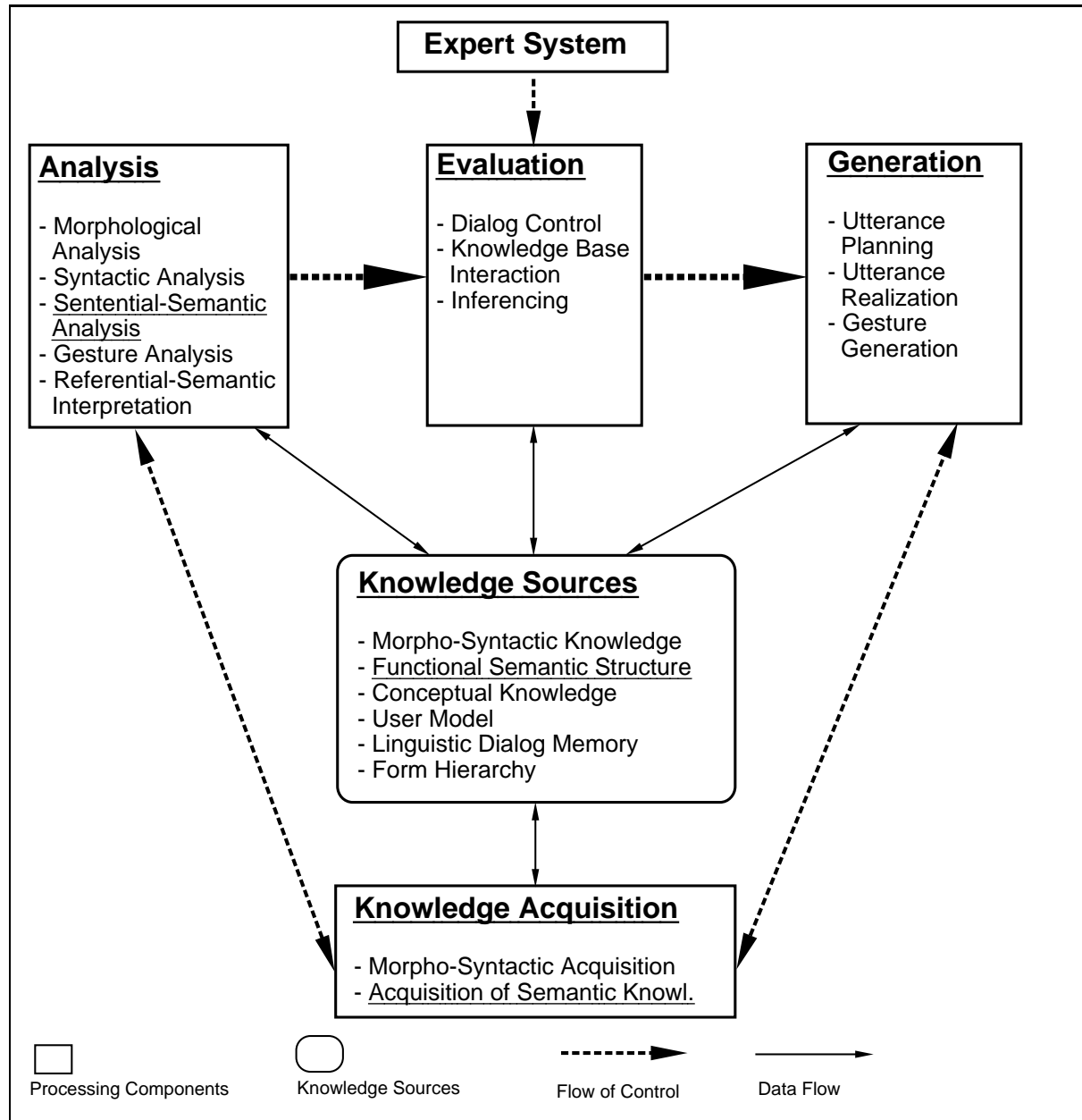


Figure 1: XTRA – system architecture

Processing in the XTRA system can be roughly separated into three phases: analysis, processing, and generation (see figure 1). A user's natural language input is first subjected to a morphological, syntactic and sentential-semantic analysis carried out by the system's parsing components and then transformed into an internal, domain-dependent representation – the conceptual knowledge base (CKB). Based upon this representation, interaction with the respective expert system takes place, the result of which also is represented in the CKB. These results provide the basis for the subsequent generation of natural language output and pointing gestures.

To adapt the XTRA system to different expert systems and to extend existing dictionaries and knowledge bases, research on knowledge acquisition has been carried out. One research goal was to develop an acquisition tool – WASTL [Jansen-Winkel, 88] –, which learns the various aspects (namely morphological, semantic and contextual aspects) of new words from natural language examples. WASTL makes use of subproblem-specific knowledge acquisition modules as well as of selected modules of the natural language system. The component for the acquisition of semantic lexicon entries, represented within Functional Semantic Structure (FSS), is FSS-WASTL. This component makes use of the sentential semantic Analysis (SSA) module, the knowledge representation language SB-ONE, as well as its tools.

### 3 Interactive Classification

Our interactive classification algorithm was developed for the knowledge representation language SB-ONE [Kobsa, 91] – a KL-ONE lookalike [Brachman & Schmolze, 85]. In SB-ONE a distinction is made between the definitional knowledge, represented as concepts and roles in the so-called terminological box (TBox), and the description of a specific state of the world, expressed by assertions which use the terminology of the TBox in the so-called assertional box (ABox).

The terminological knowledge in SB-ONE is modeled through *general concepts* and *general roles*. In a logical interpretation<sup>1</sup> they correspond to one- and two-place predicates, respectively. The concepts of the TBox are ordered in a subsumption hierarchy (taxonomy), where the ISA-link between two concepts is interpreted as the subset-relation between the corresponding predicate extensions. General concepts are characterized by attribute descriptions. Each attribute description consists of:

- a role predicate, which denotes a two-place predicate between the domain concept and a value restriction concept
- a number restriction, expressing how many extensions of a role are needed to describe a concept in the minimal, maximal and default case,
- a modality, which indicates whether the attribute is part of the definition of every instance of the concept, i.e. whether the role is necessary or optional.

SB-ONE distinguishes between primitive and defined concepts. In the first case the concept is not fully defined by its attribute descriptions. Primitiveness means that the concept is described by necessary but not sufficient conditions. Defined concepts are fully specified by necessary and sufficient conditions.

Various tools for SB-ONE have been developed, one of which is the matcher [Aue et al., 89]. It can be used to retrieve elements and substructures of SB-ONE knowledge bases. The matching algorithm maps pattern structures – ABox structures enhanced with so-called variable concepts and roles – onto SB-ONE individualizations.

Due to SB-ONE's well defined semantics, there exists a classification algorithm which can be used to verify the subsumption hierarchy and to insert new knowledge – represented as TBox-expressions – into the correct place within the taxonomy. The input of the classification algorithm is a complete concept definition with all its attribute descriptions (roles) given. This makes classification in KL-ONE-like languages a useful, but non-interactive tool. Especially for knowledge acquisition, a more interactive process would be appreciated in which a concept description can be developed incrementally, guided by the reasoning of classification.

Finin and Silverman [Finin & Silverman, 84; Finin, 86] were the first who developed such a procedure, which they called *interactive classifier*. Their classifier works for a representation language similar to

---

<sup>1</sup>First order predicate calculus is assumed

SB-ONE, but more restricted<sup>2</sup>. The algorithm which we developed is an extension of interactive classification to full-scale SB-ONE allowing for subsumption graphs (instead of trees), roles with cardinality and modality attributes, and several kinds of relations between roles (inheritance, restriction, and differentiation). Following Finin we subdivide the interactive classification into three phases: acquisition of the initial concept description, establishing the most specific subsumers (which become the parents in the subsumption graph), and establishing the most general subsumees (which become the children in the subsumption graph).

Within these phases the second is the most important for our approach, because here an interactive top-down refinement of the user's initial concept description takes place. The refinement process is based upon the *structural differences* between concepts which guide the traversal of the graph.

**Definition Structural Differences:** Let both  $C_1$  and  $C_2$  be concepts, with  $C_2$  being a direct subconcept of  $C_1$ .  $C_1$  and  $C_2$  *differ structurally* if at least one of the following conditions occurs:

- $C_2$  has all roles of  $C_1$  and at least one additional role.
- At least one role of  $C_1$  is restricted in either its value, cardinality or modality at  $C_2$ .
- At least one role of  $C_1$  is differentiated at  $C_2$ .
- $C_2$  has at least one primitive superconcept, which does not subsume  $C_1$ .
- $C_2$  is a primitive concept.

The *structural difference* is the collection of all parts of  $C_2$ 's definition differing structurally from  $C_1$ .

## The Classification Algorithm

The algorithm starts with a given initial partial concept description  $C$  linked to an adequate superconcept  $S$ . It is the task of the user to supply these starting conditions, which depends on the application. The algorithm iterates over all subconcepts  $C_i$  of  $S$ , computes the structural differences between  $S$  and  $C_i$ , and queries the user as to which of these differences are part of  $C$ 's definition. Each selected structural difference is added to the description of  $C$ , and the interactive step is recursively performed for each of the corresponding  $C_i$  as new superconcepts of  $C$ . This recursive refinement of  $C$ 's definition ends, when no more structural difference is added. After that, the user can add new or restrict existing roles to complete  $C$ 's concept definition. Finally the non-interactive SB-ONE classifier is applied onto the now completed description of concept  $C$  in order to (a) verify all established most specific subsumers, (b) detect hidden superconcepts, and (c) find all most general subsumees of  $C$ .

## 4 Acquisition of Semantic Knowledge

We apply the interactive classification as part of the acquisition of new entries for the semantic lexicon of XTRA. The semantic knowledge in XTRA is represented in terms of a generalization of Fillmore's case frames, implemented as a terminological SB-ONE knowledge base, called the Functional Semantic Structure (FSS). The FSS is subdivided in two parts: the generic, domain-independent FSS, and the application-dependent lexical part, which is the object of the knowledge acquisition. The functional semantic structure of the user's utterances are constructed through the sentential-semantic analysis within the corresponding ABox.

The module SEMANTIX performs this sentential-semantic analysis. Based on transformation rules and the semantic lexicon, it translates the syntactic derivation's DAGs<sup>3</sup> into individualizations of the FSS. SEMANTIX interprets successively the constituents of each DAG by looking up the corresponding word stem information from the semantic lexicon and individualizing them into a coherent FSS structure as

---

<sup>2</sup>In [Finin, 86, page 1415] T. Finin describes the underlying representation language as follows: "The KB is constrained to be a tree structure, so each node has at most one parent. Nodes have single-valued attributes which represent components or characteristics that apply to the [...] concept described. Values of attributes can be numbers, intervals, symbols, or sets of symbols. [...] Each node inherits all attributes of its parent node, but its values can be restrictions of the parent attribute's values."

<sup>3</sup>Directed Acyclic Graphs as known from the PATR formalism.

the semantic representation of the example sentence. For our purposes, SEMANTIX was extended to SEMANTIX+ in order to work on only partly interpretable sentences, using variable concepts and roles as placeholders for unknown words.

The acquisition component uses example sentences containing an unknown word as its primary source of information. In a first phase, SEMANTIX is applied to analyse an example sentence semantically. The resulting ABox is matched against the TBox, using the above mentioned TBox-matcher, to dereferentiate the variable parts and find all consistent assignments. The purpose of this phase is to make a good guess for variable-free but possibly partial concept description, which becomes the initial description for the subsequent refinement process. This second phase of the knowledge acquisition is the interactive classification, starting with the most general candidate from the set of assignments, to which our classification algorithm is applied until the concept description is complete and correctly classified. To improve the performance of this classification, we encoded heuristics about how knowledge is represented within the FSS<sup>4</sup>. The resulting description becomes the new lexicon entry. To verify the result and to continue processing, the semantic analysis is restarted. Figure 2 shows the system architecture.

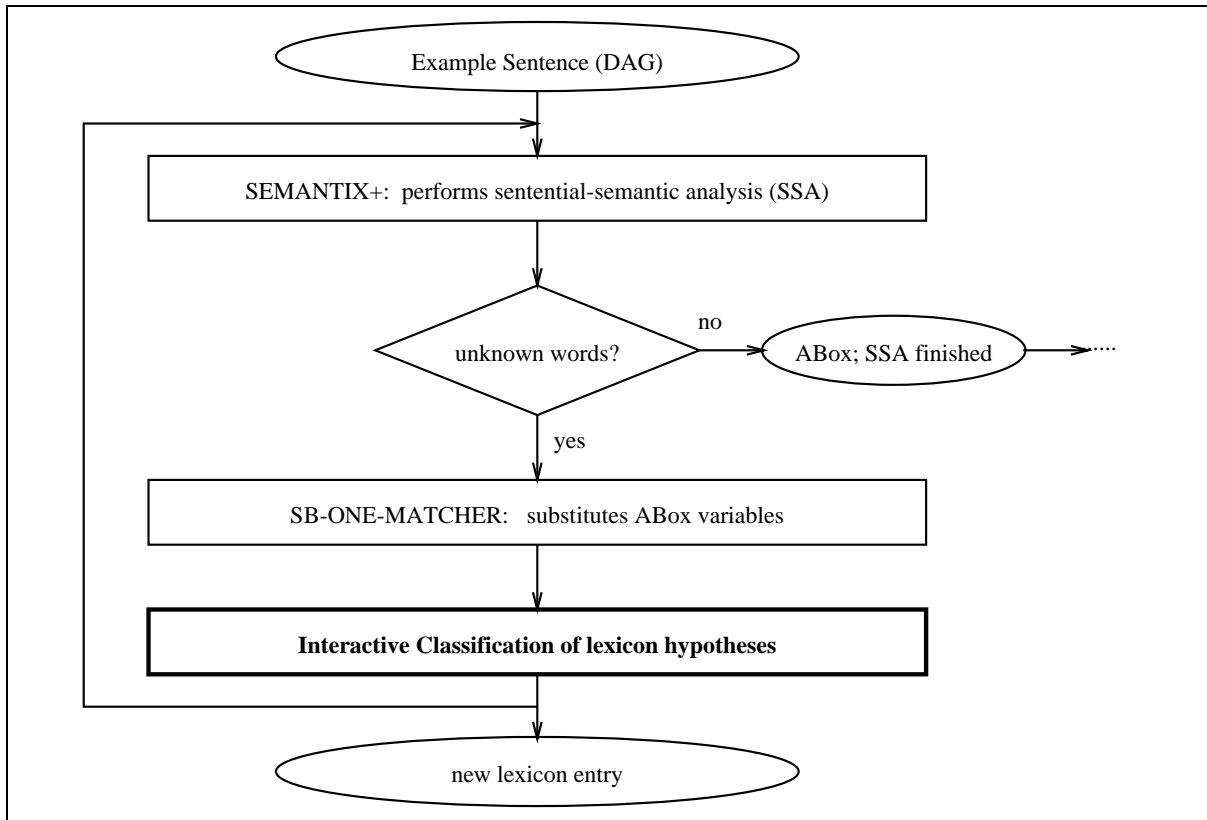


Figure 2: FSS-WASTL – System architecture

## An Example

To illustrate our idea of the application of the interactive classifier during the example-driven knowledge acquisition, consider the following user's input<sup>5</sup>:

<sup>4</sup>These heuristics encode very general conventions of the FSS's design (e.g., new roles can only be introduced for generic concepts and not for lexicon entries) as well as very specific information (e.g., concepts, which are subconcepts of both PREDICATE and THING, represent verb nominalizations and cannot be meaningful hypotheses for unknown verbs).

<sup>5</sup>The following examples will be given in German with the English translation in parenthesis.

Der Mann radelt vom Tagungshotel zum Strand.  
 (The man cycles from the conference hotel to the beach.)

We assume that the system recognizes all words except for `radelt` (*cycles*). Moreover, we assume, that the sentence successfully passes the morphological analysis<sup>6</sup>. The unification based parser analyzes the user's input sentence syntactically and constructs as a result a derivation graph (DAG). This DAG is the input to the sentential-semantic analysis, during which an error, due to the incomplete lexicon, will arise. Following the acquisition algorithm described above, FSS-WASTL builds an ABox enhanced by variable concepts and roles. The TBox-matcher will be used in order to find TBox concepts consistent with the unknown word. These are MOTION and all its subconcepts, which are established due to the semantic roles agent, source and destination (see figure 3). The final step is the interactive classification, starting the incremental refinement with the most general hypothesis (MOTION). As described above, the interactive classification is guided by the structural differences. Thus, given the TBox in figure 3, the user is asked the following question:

“Wird die optionale Rolle `instrument` auf die notwendige Rolle `means` restringiert?”  
 (“Will the optional role `instrument` be restricted to the necessary role `means`?”)<sup>7</sup>

If the user responds “yes”, MOTION-BY-MEANS will become the actual most specific subsumer and the process continues recursively.

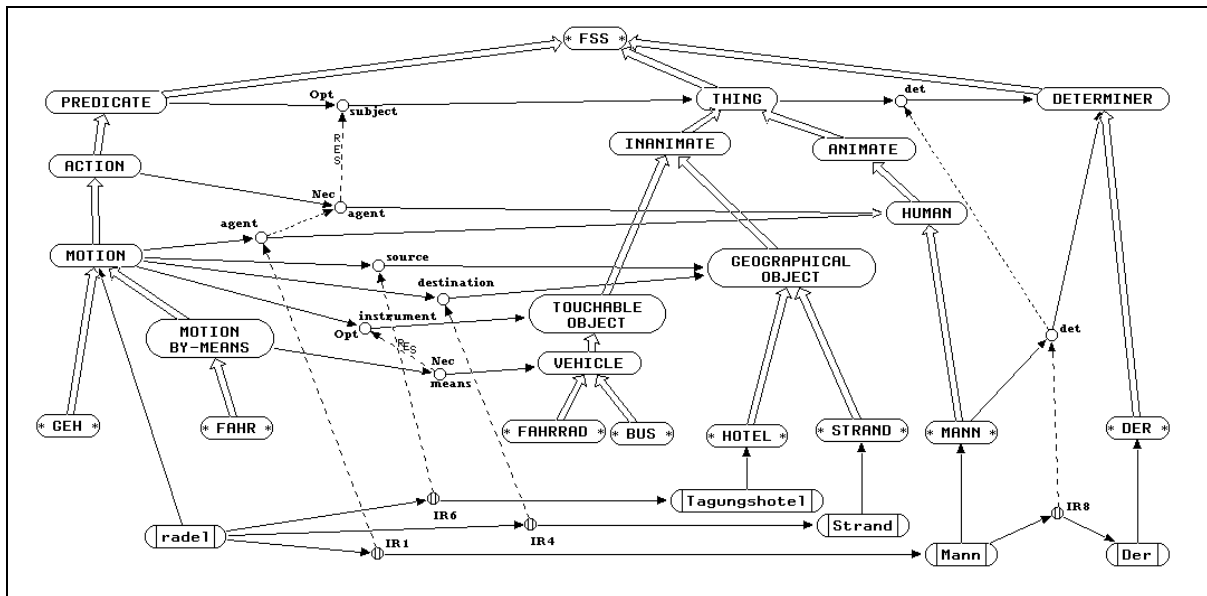


Figure 3: An extract of the Functional Semantic Structure. The upper part of the TBox contains the generic FSS. Lexicon entries, which are marked with asterisks, can be found in the middle of the picture. The concepts and roles in the lower part form the ABox with variable structures (e.g. `rade1`).

The results which can be achieved through the application of FSS-WASTL depend on the example sentences. Bad example sentences are those with very limited information about unknown words, e.g. very short sentences: *Ich radle.* (*I cycle.*). In this case the only possible hypothesis for `radeln` depends on its syntactic category and makes it a subconcept of PREDICATE. On the other hand, the

<sup>6</sup>If an error occurs during this phase, the morphologic acquisition component WIMOX [Klein, 90] is activated.

<sup>7</sup>This is an application independent question, which asks for elements of the knowledge base. Another approach is to use the generator in order to verbalize a question like “Is a vehicle used in order to `radeln`?”. This kind of questions, however, can only be generated with additional knowledge about how the respective knowledge base is structured and how to verbalize this additional knowledge.

choice of appropriate examples prunes the search space strongly, as e.g. in *Ich fahre mit dem Bus nach Wemmetweiler. (I go by bus to Wemmetweiler.)*, where *Wemmetweiler* is the unknown word and the most general hypothesis, which the TBox-matcher finds, is `GEOGRAPHICAL OBJECT`. This initial description leads within one or two steps to a correct lexicon entry.

## 5 Building Paraphrases

The user interface to the interactive classifier offers some explanation facilities. It supports *Why*- and *How*-questions. The *Why*-option informs about the goal considered at the moment, while the *How*-explanation works like a tracer and lists the steps already performed.

Referring to the example above, we remark that the user has to know the structure of the FSS, viz. the meaning and the interpretation of the concepts and roles within the FSS, in order to sensibly answer the questions of the interactive classifier. Although the interactive classifier has a menu-based interface and an explanation component, we decided to generate example sentences as alternative explanations. The generated sentences, which we call *paraphrases*, have to clarify the structural differences between two concepts. The paraphraser is useful for two purposes: verbalizing the content of the knowledge base, and verifying the consistency and plausibility of new concept definitions<sup>8</sup>. The paraphraser is interfaced to POPEL, the generation component of XTRA.

### The Generation Component POPEL

The natural language generation system POPEL<sup>9</sup> [Reithinger, 90] has been designed as a generator which pays special attention to discourse phenomena and integration within an overall system.

POPEL's main features are the interaction between the "what-to-say" and the "how-to-say" components, and the incremental selection and realization of the utterance. The "what-to-say" component consists of a selector which determines and activates the conceptual content, and a context handler which handles the choice of descriptions. The decisions of this component are heavily based on contextual knowledge. The "how-to-say" component POPEL-HOW is especially designed for the requirements of XTRA and uses the knowledge sources of the whole system that were designed to be used bidirectionally by both generation and analysis. Using intermediate linguistic-based descriptions, namely the FSS and syntactic structures, it translates the content of the non-linguistically oriented conceptual knowledge representation into natural language. POPEL is realized on a (simulated) parallel processor allowing for incremental generation.

### The FSS-WASTL Paraphraser

The generation of paraphrases begins from the FSS processing level within POPEL-HOW. In order to select a possible structural description for the new concept, a specific subsumer is determined. The paraphrase's FSS-structure is then constructed according to the FSS structure of the subsumer, the structural differences already computed, and the syntactic structure of the input sentence. Although processing in POPEL is based on the SB-ONE knowledge representation language, the generator is not tightly coupled to it; especially word choice is performed independently. This facilitates its use during paraphrase generation: it is possible to separate the words that are to be uttered from the FSS structure. The FSS-WASTL paraphraser verbalizes the content of the knowledge base around the subsumer. In case the user accepts the paraphrase, the new concept is inserted as subconcept of the current subsumer. Let us consider again the example above. We assume that the value restriction of the role *means* from `FAHR (RIDE)` should be `FAHRRAD (BICYCLE)`. For the transition from `MOTION-BY-MEANS` to `FAHR`, the interactive classifier asks the user by means of a menu:

"Wird der Wertebereich der Rolle 'means' von `VEHICLE` auf `FAHRRAD` restringiert?"  
("Will the range of the role *means* be restricted from `VEHICLE` to `FAHRRAD`?")

---

<sup>8</sup>A more detailed discussion can be found in [Weischedel et al., 87] or [Ndiaye, 90].

<sup>9</sup>POPEL is the acronym of "Production Of {Perhaps, Possibly, P...} Eloquent Language".

On demand, the FSS-WASTL paraphraser explains the effects of this question by verbalizing the resulting intermediate description. For our example, the input of the generator consists of the TBox-expression corresponding to the individualized part of Figure 3, where the general concept of `radel` is restricted from `MOTION-BY-MEANS` to `FAHR`, and where `radel` has the additional role `means` with the value `Fahrrad`. The paraphraser generates the following sentence:

“Der Mann fährt mit dem Fahrrad vom Tagungshotel zum Strand.”  
(*The man rides a bicycle from the conference hotel to the beach.*)

## 6 Conclusion

The problem FSS-WASTL deals with – the acquisition of new semantic knowledge – arises frequently in natural language systems. Thus several solutions to this problem have already been proposed. IRACQ, developed as part of BBN’s IRUS system [Weischedel et al., 87], uses a similar approach. It starts from a natural language example, establishes hypotheses for an unknown word by partly interpreting the example sentence, and then tries an interactive top-down refinement of these hypotheses. Unlike FSS-WASTL, the modules for IRACQ were especially developed for this application and did not evolve from the extension of existing analysis modules (SEMANTIX) and the accommodation of general tools (TBox-matcher and Interactive Classifier). DANTE, developed at the IBM Scientific Center of Rome [Antonacci et al., 89], solves its task in a multiple example sentences, bottom up approach. For DANTE, an induction algorithm was developed which like FSS-WASTL and IRACQ partly analyzes its inputs sentences and then forms lexicon hypotheses which cover all given examples.

The interactive classifier has its origin in the idea of Finin and Silverman in making the major reasoning procedure for KL-ONE-like languages – the classification – an interactive task. Unlike conventional net browsers, the interactive classifier traverses the taxonomy, relying on the defined structure of the concepts and not on their names or the user-ascribed position in the net. This meets the central idea of KL-ONE based knowledge representation languages, thus eliminating a major drawback of other frame-based languages. This approach is useful as a general-purpose, stand-alone tool as well as embedded in a specialized knowledge acquisition module, as was done with FSS-WASTL.

Using structural differences facilitates the natural language feedback during the development of knowledge bases. We demonstrated this within a natural language system, where such a feedback can be integrated easily. Even more important, however, is the ability to interface this algorithm to a natural language generator in order to verbalize example sentences based on the explanations about the internal representation of any new concept. This makes the approach transferable to other applications with arbitrary SB-ONE knowledge bases.

The problem to extending the use of exemplary natural language input to arbitrary SB-ONE knowledge bases is even more serious. In our understanding of natural language processing, an arbitrary SB-ONE knowledge base can only be integrated as the conceptual knowledge of the natural language system. This requires at least one further acquisition step: In addition to the acquisition of lexicon information as described above, which is a prerequisite for further interpretation of an example sentence, the acquisition step for the conceptual information itself is needed. Furthermore, an extension of interpretable natural language input to e.g. definitional sentences as used in dictionaries and as NANOKLAUS [Haas & Hendrix, 84] tried to interpret (e.g. “To cycle means to ride a bicycle”) would increase the power of our method. Both problems are topics of our current research.

All systems mentioned are implemented in Common Lisp and work on various hardware platforms. They are integrated into the XTRA system and help to extend the semantic lexicon of XTRA’s current application.

## Acknowledgments

We would like to thank Jürgen Allgayer as well as John Nerbonne for their proof-reading of this paper and valuable discussions on the classification algorithm.

## References

- [Allgayer et al., 89a] J. **Allgayer**, K. **Harbusch**, A. **Kobsa**, C. **Reddig**, N. **Reithinger**, and D. **Schmauks**. *XTRA: A natural-language access system to expert systems*. International Journal of Man-Machine Studies., 31(2):161–195, August 1989.
- [Allgayer et al., 89b] J. **Allgayer**, R. **Jansen-Winkel**, C. **Reddig**, and N. **Reithinger**. *Bidirectional use of knowledge in the multi-modal NL access system XTRA*. In: 11<sup>th</sup> IJCAI, pp. 1492–1497, 1989.
- [Antonacci et al., 89] F. **Antonacci**, M. **Russo**, M. T. **Pazienza**, and P. **Velardi**. *A system for text analysis and lexical knowledge acquisition*. Data & Knowledge Engineering, 4(1):1–20, Juli 1989.
- [Aue et al., 89] D. **Aue**, S. **Heib**, and A. **Ndiaye**. *SB-ONE Matcher: Systembeschreibung und Benutzeranleitung*. Sfb 314, memo 32, Fachbereich Informatik, Universität des Saarlandes, Saarbrücken, 1989.
- [Brachman & Schmolze, 85] R. J. **Brachman** and J. G. **Schmolze**. *An Overview of the KL-ONE Knowledge Representation System*. Cognitive Science, 9(2):171–216, 1985.
- [Finin & Silverman, 84] T. **Finin** and D. **Silverman**. *Interactive Classification: A Technique for the Acquisition and Maintenance of Knowledge Bases*. Report ms-cis-84-17, University of Pennsylvania, Computer and Information Science, 1984.
- [Finin, 86] T. **Finin**. *Interactive Classification: A Technique for Acquiring and Maintaining Knowledge Bases*. In: IEEE, volume 74, pp. 1414–1421, October 1986.
- [Haas & Hendrix, 84] N. **Haas** and G. **Hendrix**. *Learning by being told*. In: R. S. Michalski, J. G. Carbonell, and T. M. Mitchell (eds.), Machine Learning: An Artificial Intelligence Approach, pp. 405–427. Springer-Verlag, 1984.
- [Jansen-Winkel, 88] R. M. **Jansen-Winkel**. *WASTL: An Approach to Knowledge Acquisition in the NL Domain*. In: EKAW, pp. 22.1–22.15, Bonn, W. Germany, June 1988.
- [Klein, 90] H. **Klein**. *WIMOX – ein Wissenserwerbssystem für ein morphologisches Lexikon*. Master thesis, Fachbereich Informatik, Universität des Saarlandes, Saarbrücken, 1990.
- [Kobsa, 91] A. **Kobsa**. *Utilizing Knowledge: The Components of the SB-ONE Knowledge Representation Workbench*. In: J. Sowa (ed.), Principles of Semantic Networks: Explorations in the Representation of Knowledge. San Mateo, CA: Morgan Kaufmann, 1991.
- [Ndiaye, 90] A. **Ndiaye**. *FSS-WASTL – Ein interaktives Wissensakquisitionssystem zur Erweiterung des semantischen Lexikons in XTRA*. Master thesis, Fachbereich Informatik, Universität des Saarlandes, Saarbrücken, 1990.
- [Reithinger, 90] N. **Reithinger**. *POPEL – A parallel and incremental Natural Language Generation System*. In: C. Paris, W. Swartout, and W. Mann (eds.), Natural Language Generation in Artificial Intelligence and Computational Linguistics. Kluwer Academic Publishers, 1990.
- [Weischedel et al., 87] R. **Weischedel**, D. **Ayuso**, A. **Haas**, E. **Hinrichs**, R. **Scha**, V. **Shaked**, and D. **Stal-lard**. *Research and Development in Natural Language Understanding as Part of the Strategic Computing Program*. Annual technical report 6522, BBN Laboratories Incorporated, Cambridge, MA, 1987.