Communicative Language Learning with PROMISE

Friedrich Kronenberg, Petra Ludewig
Institut für Semantische Informationsverarbeitung
Universität Osnabrück
D 49069 Osnabrück, Germany
Tel.: +49 541 969 2585
Fax: +49 541 969 2500
email: fritz@hal.cl-ki.Uni-Osnabrueck.DE
ludewig@dosuni1.rz.Uni-Osnabrueck.DE

Introduction

In a one-year project at the course of study "Computational Linguistics and Artificial Intelligence" students of the University of Osnabrueck, supported by scientists of the Institute for Semantic Information Processing and language teachers engaged in media-supported language teaching, developed a prototypical system called PROMISE which illustrates future perspectives of computer assisted language learning (CALL).
PROMISE is an acronym for "PROjekt Mediengestütztes Interaktives Sprachelnlernen Englisch" (i.e. media supported interactive language learning - English). This project aimed to demonstrate how to decrease the discrepancy between available language learning software and the actual state of the art within Natural Language Processing (NLP) and Knowledge Representation (KR) techniques on the other hand. By using these techniques the PROMISE-project realized (on the basis of the PROLOG development system G_LOG (Gust 1992)) exercise types which take into consideration the requirements of communicative language teaching which considers communicative competence as a main goal of modern language acquisition. All elements of the foreign language ought to be introduced in situational contexts. The learner should be aware that language is a means of communication and that the foreign language can be used to express his/her intentions. In PROMISE, this communicative approach is realized by putting the learner in an adventure-game-like setting where authentic and purposeful language use is possible. These simulations have stimulating effects on the communicative behaviour of students. Vocabulary and grammar will be learned more effectively when all language phenomena are embedded in thematic contexts, i.e. in situations. Within a situational frame (e.g. a road accident) the student is guided through a series of dialogue exercises with "free" learner input in which the student "talks" with a simulated dialogue partner in the foreign language and gets meaningful responses from this partner. Such a creative language behaviour cannot be dealt with standard technology but only via knowledge based systems with sophisticated language processing abilities. The components that achieve these abilities are described in this paper.

Knowledge based systems vs. conventional systems

To exaggerate, one might say that classical CAI (Computer Assisted Instruction) programmes have the wide scope of functions books can offer along with optimized cross references and fast feedback (cf. Puppe 1992). Their deficiencies originate primarily from the fact that they do not have at their disposal a model of the knowledge to be taught and, as a result, cannot adequately treat learner input which has not been explicitly anticipated before. In exercises that accept "free" learner input with regard to a given language fragment there is no acceptable way to anticipate all possible answers. Even for traditional exercises an almost complete anticipation of correct answers is not always easy and the anticipation of incorrect answers usually knows no bounds. For this reason multiple choice and similar exercises - in which most of the learner's answers can be predicted- enjoy great popularity from the developers' perspective (fur further details see: Gust et al. 1994). This state of affairs comes into serious conflict with the generativity of natural languages and the educational claim of linguistic creativity.
Having access to a model of the knowledge to be taught and of the student to be instructed as well as didactic principles knowledge based CALL systems are able to behave more flexible than traditional educational programmes.

Artificial Intelligence / Computational Linguistics and CALL

From the perspective of Artificial Intelligence (AI) and Computational Linguistics (CL) CALL can be regarded as an almost ideal application to demonstrate the potential benefit and the workability of CL and AI technology. In contrast to other language applications, e.g. computational translation, CALL itself postulates a deliberate restriction to special lexical and grammatical fragments and well-defined situations. While other language applications suffer from the incompleteness of grammar and lexicon this restriction can be viewed as an essential principle of language teaching and, thus, can be - at least to a large extent - justified by a claim inherent in the CALL application itself (cf. Kronenberg et al. 1994). For this reason the transition from experimental and prototypical systems to systems ready for use should be less difficult for CALL than for any other language application. But, to guarantee the acceptance of CALL systems a considerable cooperation between scientists on the one hand and language teachers on the other hand will be needed. Nevertheless almost all research topics occuring in CL and AI can be adressed in CALL and the problems arising with respect to the realization of CL&AI concepts may give valuable hints for future basic research.

System characteristics

In this section the main characteristics of the PROMISE system (cf. figure 1) are briefly described.

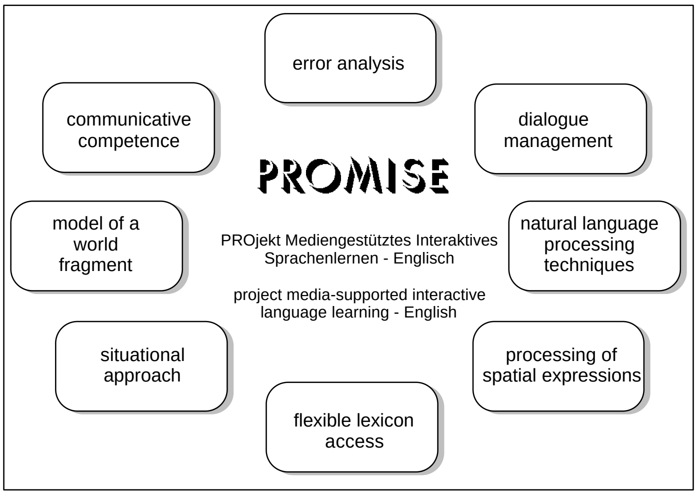

Fig.1: system characteristics

Situational approach / Model of a world fragment

The best way to promote authentic language behaviour is to present und use language in situational contexts. The learner should be motivated to use the foreign language as a means of communication within communicative situations. In PROMISE this approach is realized by embedding a series of dialogue exercises in a situational frame (e.g. road accident; cf. Fig. 2).

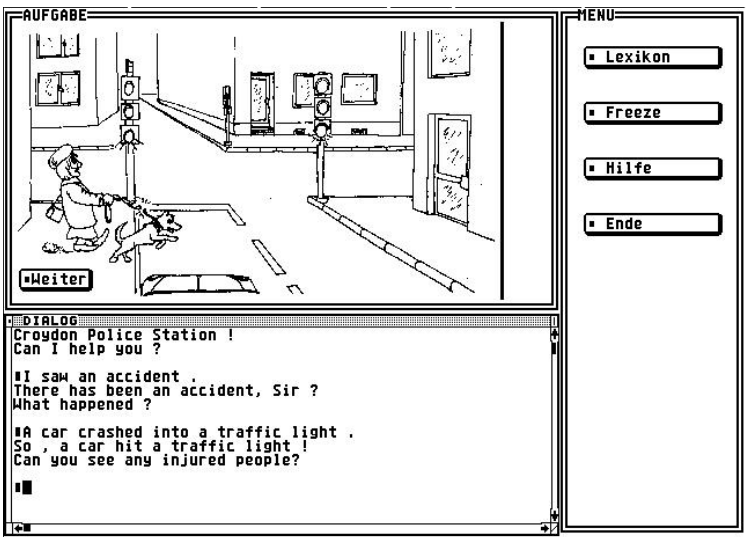

Figure 2: A sample dialogue of an emergency call

A dialogue of an emergency call is shown in figure 2. The student has to report to the police an accident he/she has just watched. Within this emergency call the learner is asked several questions about the accident. The dialogue partner - simulated by the system - reacts in a way corresponding to the semantics of the learner's utterance and its knowledge about the structure of emergency calls. A non-cooperative learner will soon discover that the “policeman” can “understand” what the learner is talking about and that the dialogue partner can “hang up the telephone” if necessary. Following this strategy, we clearly distinguish the tutor who meticulously corrects the learner and gives him valuable hints and the simulated dialogue partner who tries to be as cooperative as possible.

To enable a "dialogue partner" to react meaningful in situational contexts the system must have at its disposal grammatical, lexical and taxonomical knowledge as well as knowledge about dialogue structures and needs to "know" about the current situation, the objects, the learner's perspective in the situation. Situation specific and taxonomic knowledge is organized in sortal ontologies. Lexical elements and objects in graphically represented situations are linked to entities in an ontology via sortal pointers.

Communicative Competence

The PROMISE system comprises several linguistic knowledge bases (grammar and lexicon (cf. fig. 3), taxonomy and scripts descriping dialogue structures (cf. fig. 6)) and natural language processing components (lexicon access, morphology, parser, dialogue manager, sortal check) to achieve the ability to process natural language input syntactically and semantically. This is a necessary condition for having authentic and purposeful dialogues in exercises. On the basis of the dialogue management component the learner can enter natural language utterances in dialogue exercises within situative motivated contexts. The semantic representation of the learner's input produced by the parser is used by the dialogue management component to compare the information given by the learner with the information expected in the situational surrounding.
Computational Linguistics cannot offer a complete grammatical and lexical description for a language yet and in order to reduce processing time CALL applications - like all language applications - must operate on a strictly restriced lexical and grammatical fragment. As a result not every correct language input can be recognized as grammatical. But this time, it does not spoil the effect, since, anyway grammatical, lexical and situational restrictions are natural for language teaching.
In cases when the learner input was not parsed successfully or the predicate-argument-structure, produced by the parser, could not be sucessfully processed, the PROMISE system guarantees the continuation of the dialogue. In these cases the dialogue management (see below) uses keyword based techniques to find a “topic” that can be used as a basis for a reaction or asks for a re-formulation of the input sentence.

Natural Language Processing Techniques

We will now take a closer look on the parsing component named PACO11G (an acronym for: parser-compiler Lexical Functional Grammar (LFG); cf. Bresnan 1982) and the linguistic knowledge sources grammar and lexicon. The linguistic knowledge used by PACO11G (Sauer 1994; Sauer 1995) is encoded in the formal language of the LFG. From our point of view LFG formalisms are especially well suited for CALL systems since a number of primitives in LFG, e.g subject and object, are comparable with corresponding concepts in "classical" pedagogical grammars and, thus, permit an easier transfer from analysis via LFG grammar to error explanations in terms of a pedagogical grammar. An LFG-rule consists of a context free skeleteon plus annotations which describe feature terms and control their unification. In Fig. 3 we show a small fragment of our grammar and lexical entries[1] for the lexemes *be*, *is*, *know* and *on*. The grammar shows rules which describe how a matrix sentence (MS) and certain verbal phrases (VP) are to be analyzed. Since the phenomena which have to be processed can be restricted in CALL environments, we work with several sub-grammars which can be activated if the corresponding grammatical constructions (e.g. if-constructions) are likely to appear.

```
MS --▷ NP <↑ SUBJ>=↓ <↑ SUBJ CASE> = nom
       VP ↑=↓ <↑ MOOD> = fin
          (/ <↑ TENSE> = pres / <↑ TENSE> = past /) .

VP --▷ [  V ↑=↓ <↓ NEG>≈+ V <↑ XCOMP>=↓ <↓ MOOD>≈inf
       /  V ↑=↓ ~<↓ NEG>
       ]
       [  NP (/ <↑ NEG>≈+ <↑ XCOMP OBJ_DIR> = ↓ <↑XCOMP OBJ_DIR CASE> = acc
              / ~<↑ NEG>  <↑ OBJ_DIR> = ↓ <↑ OBJ_DIR CASE> = acc
             /)
         [P <↑<↓ PCASE>> = ↓ NP <↑ OBJ_IND >=↓  <↑ OBJ_IND CASE> = acc
       ] ]
       [ VPBAR (/  <↑ NEG>≈+ <↑ XCOMP XCOMP>=↓
               /  ~<↑ NEG>  <↑ XCOMP>=↓
               /)
       ]
       [ AP <↑ MODIFIER>=↓ ]
       [ PP (/ <↑ PCOMP>=↓ / ∃ ? <↓ PCASE> <↑ <↓ PCASE>>=↓ /)] .

VP --▷ VP PP ~<↓ PCASE> ↓ ε <↑ ADJUNCT> .

VPBAR --▷ V ↑=↓ <↑MOOD> ≈ part
            (/ <↑ TENSE> ≈ pres
               (/ <↑AKTIONSART> ≈ resultativ
                / <↑ AKTIONSART> ≈ dynamisch
                / <↑ AKTIONSART> ≈ durativ
               /)
            / <↑ TENSE> ≈ past
            /)
            [ NP <↑ OBJ_DIR> = ↓ <↑ OBJ_DIR CASE> = acc ]
            [ PP (/ <↑ PCOMP>=↓ / ∃ ? <↓ PCASE> <↑ <↓ PCASE>>=↓ /)] .

VPBAR --▷ [ TOCONJ <↓ STRUC> ≈ xcomp_to]
          V ↑=↓ <↑ MOOD> ≈ inf
            (/ <↑ AKTIONSART> ≈ punktuell
            / <↑ AKTIONSART> ≈ dynamisch
            / <↑ AKTIONSART> ≈ statisch
            /)
          [ NP <↑ OBJ_DIR> = ↓ <↑ OBJ_DIR CASE> = acc ]
          [ PP (/ <↑ PCOMP>=↓ / ∃ ? <↓ PCASE> <↑ <↓ PCASE>>=↓ /)] .

VPBAR --▷ CONJ <↓ CONN>≈ sc
          MS ↑=↓ .

PP --▷ P NP <↑ OBJ>=↓ <↑ OBJ CASE> = acc .
```

```
be :: V  <↓ S_FORM> = <BEFINDEN >
         <↑ MOOD> = fin
         <↑ AKTIONSART> = statisch .

is :: V
       (/ <↓ S_FORM> = <PRED0  ∃<↓ SUBJ> ∃<↓ XCOMP>: nil>
          <↓ SUBJ> = <↓ XCOMP SUBJ>
          <↓ XCOMP MOOD> ≈ part
          <↓ XCOMP ASPEKT> = prog
        / <↓ S_FORM> = <BEFINDEN ∃<↓ SUBJ>: agens ∃<↑ PCOMP>: position>
          <↑ PCOMP MODE > = local
        / <↓ S_FORM> = <PRED0  ∃<↓ SUBJ> ∃<↓ MODIFIER>: nil>
          <↓ SUBJ> = <↓ MODIFIER OBJ>
        / <↓ S_FORM> = <PRED0 ∃<↓ SUBJ> ∃<↓ OBJ_DIR>: nil>
        / <↓ S_FORM> = <PRED0 ∃<↓ SUBJ> ∃<↓ OBJ_DIR>: nil>
          <↓ SUBJ FORM > ≈ there
          <↓ OBJ_DIR NUM> ≈ sg
       /)
       <↓ SUBJ PERS> = III
       <↑ TENSE> = pres
       <↑ MOOD> = fin
       <↑ AKTIONSART> = statisch
       <↓↓ βGF>  = be .

know■ :: V
         <↓↓ βMORPH> = ["" ing s n]
         (/ ↓↓ = [&t WIWAVERM(KENNEN)]
          / ↓↓ = [&t INTR_PA(KENNEN)]  /) .
on :: P
        <↓ S_FORM> = <AUF ∃ <↓ OBJ>: ref_obj>
        <↓ OBJ S_FORM PRED> ≠ < STRASSE_R>
        (/ <↑ PCASE> = < ON >
         / ~ <↑ PCASE> /)
        <↑ MODE> = local .
```

Fig. 3: parts of LFG grammar and lexicon

---

[1]PACO11G supports the processing of unknown nouns especially proper nouns which, per se, cannot adequatly be treated via lexical enumeration.

The parser which operates on this linguistic knowledge sources is based on the Earley algorithm, which is a particular kind of active chart parsing algorithm. Two main characteristics of this approach are:

- a data-structure called chart is used to store all found constituents and all (partly) validated hypotheses about the input string;
- by using items in the chart the parser won't parse an already found constituent again.

Descriptions of active chart parsing and Early algorithm can be found in (Naumann 1994).
A chart parser for LFG like PACO11G assigns two levels of syntactic description to every input string that has been parsed: A Constituent Structure (c-structure) and a Functional Structure (f-structure). While the c-structure corresponds to a conventional phrase structure tree, the f-structure expresses grammatical information in a form which can serve as the input to semantic interpretation (cf. figure 4).

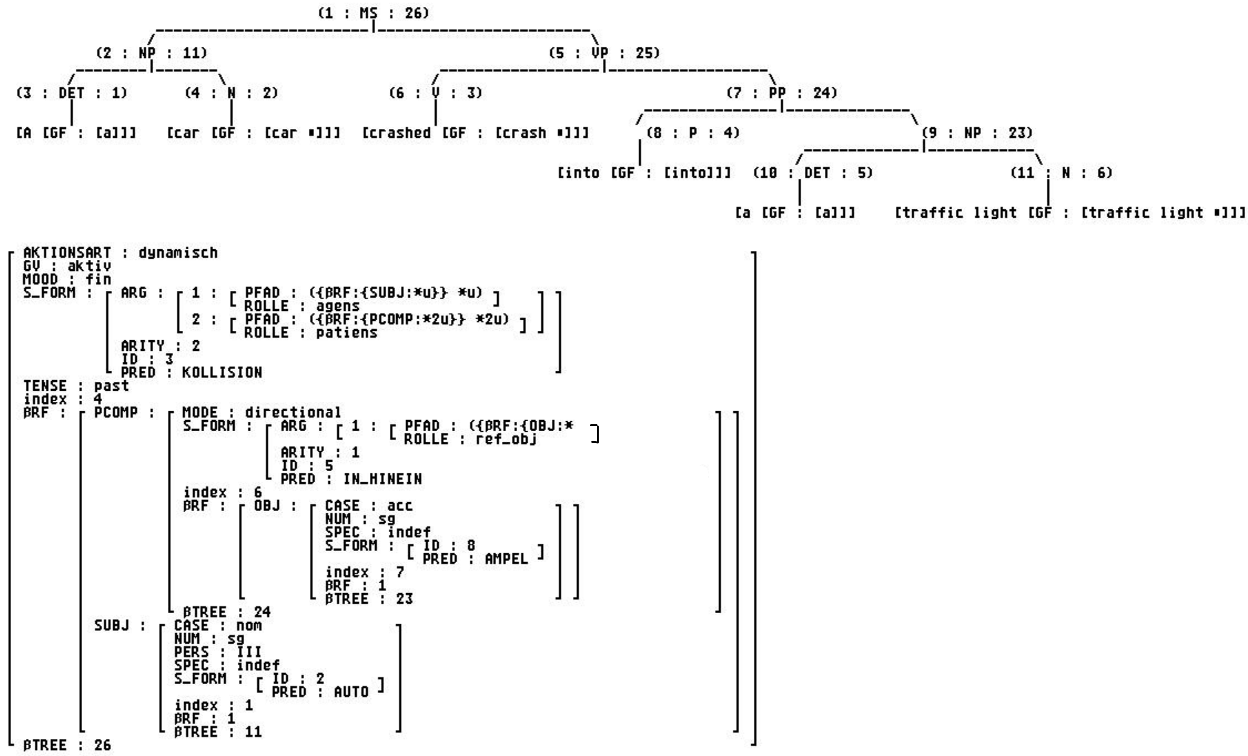

Fig. 4: c- and f-structure of the sentence "A car crashed into a traffic light"

Error analysis

Our main focus in syntactic error analysis[2] is on interlingual transfer errors (e.g. interference errors) which can be representated in terms of error rules in the grammar and in terms of lexical templates (abbreviation rules). Since the overall guidelines for English teaching in Germany give special attention to the avoidance of transfer errors (cf. Niedersächsisches Kultusministerium 1989) we decided to base our error analysis on special rules in lexicon and grammar describing phenomena which can be traced back to misconceptions caused by interlingual transfer from the learner's first language. The LFG-parser uses these error rules and templates in the same way as the descriptions for the fragment of the correct language. After a successful parse the system has to decide whether any analysis for the learner’s utterance was build without using error rules and templates or not. In the latter case an “error analysis”-control component builds up error explanations which are anticipated with the error rules. Schwind (1994) calls this approach sensitive parsing in contrast to robust parsing. This means that the parser tries to detect and to describe all possible errors (and their causes) while robust parsing aims to ignore errors.
Not yet implemented is our approach towards non-anticipated error diagnosis. In this approach errors won’t be detected on the basis of error rules but by using special parsing techniques. The basic idea behind our strategy (Sauer 1995) is to combine a bottom-up oriented and a top-down oriented chart parser which both operate on the same chart (a data structure that holds information about found constituents and about hypotheses which

[2]The organization of the lexical knowledge and the overall system architecture also support the detection of morphological errors based on erroneous suffixation.

still have to be (partially) validated). The top-down parser uses generalized chart-parsing techniques (like in robust parsing) which enable the construction of hypotheses even after the occurence of an error. By using both parsing strategies we might narrow down the possibly erronreous constituent. The ideas underlying this approach are described in (Mellish 1989).

The semantic error analysis operates on the basis of the functional descriptions built up by the parser. This description is transformed into a representation of the predicate-argument-structure. Each position in this structure in connected with sortal information which points to hierarchically organized entities in our world knowlege database. By using the sortal information the systems checks whether all positions in the predicate-argument-structure are compatible with our taxonomic knowledge. If incompatibilities are found the tutor component will report them whereas the reaction of the “dialogue partner” is not neccessarily affected by a sortal violation. The same statement is true for all syntactic errors: An error is always reported to the learner by the tutorial component, but within an exercise the dialogue will continue undisturbed if the functional describition of the learner’s input allows a meaningful reaction to the utterance, i.e. the “dialogue partner” has understood the learner despite of his/her imperfect language use. A dialogue should not be disturbed as long as the learner’s answer can be “understood” and a meaningful reaction is possible.

Flexible and content based lexicon access

In all exercises the learner can look up lexical information. In contrast to printed dictionaries electronic dictionaries offer more and new ways to access the vocabulary. To guarantee a flexible and content based access with graphical support a lexical query can consist of

- a single lexeme
- one or more feature/value pairs
- a mouse click on an object in a graphically represented situation

If more than tree matches are found the lexemes will be shown in a separate window. In this window the student can select the lexemes he/she wants to see in detail.

As an example for graphic and situation based access to lexemes we take a look at the English lexeme pedestrian crossing. A student, who does not know this word, can open a situation in which the corresponding object "pedestrain crossing" occurs. By clicking on this object the lexemes (here: *pedestrian crossing* and *zebra crossing*) are shown with all their lexical information.

The learner gets access to an appropriate situation by a) directly loading the situation from the main menu or b) by looking up a known word which can be found in the same situational surroundings as the word looked for. An example: A student who is looking for the lexeme *pedestrian crossing* and who knows that the word *traffic light* appears in the same situational context can look up the entry for *traffic light*. In this entry he/she finds a pointer to a traffic situation (in German: VERKEHR). By clicking on this pointer a graphic with a typical traffic situation appears (Figure 5). In this picture the denoted object (in this case the "traffic light") is marked by inversion. The student can now click on other depicted objects, and, thus, access lexical entries.

The depicted objects are releated to the lexicon via object types (sorts). Each lexical item which has as a semantic description the corresponding sort (e.g. the synonyms *pedestrian crossing* and *zebra crossing* both point to the sort ZEBRASTREIFEN) is found by following the sortal pointer.

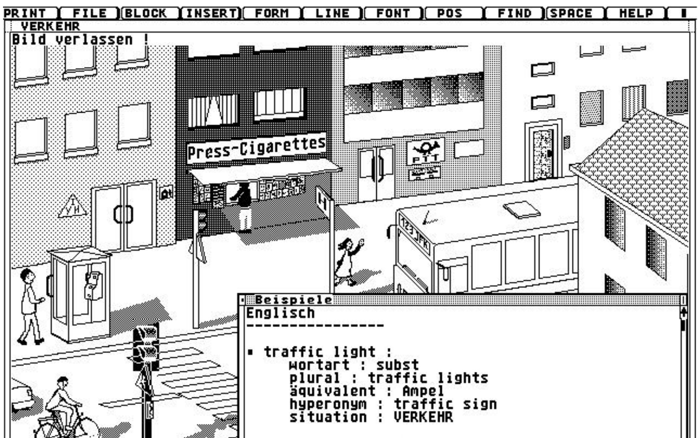

Fig. 5: Situation- and object based access to unknown words

The lexcion can be seen as one of the most important knowledge bases in a CALL system. It is accessed by the learner via lexical queries (lexical search pattern and object based access) and by the parser which tries to analyze the student's input. Both parties are requesting very different types of lexical information (regarding represenation and content). Considering these contraints we decided to construct two separate lexica each with its own specific characteristics: a technical or system lexicon (cf. figure 3) used by the parsing component and a pedagogical or learner lexicon (cf. figure 5). To get an efficient lexicon acquistion and to make sure that both lexica store consistent information we will use an approach to combine invariant lexical information from different sources developed at the Institut for Semantic Information Processing at the University of Osnabrueck (cf. Hötker et al 1991, Hötker et al. 1994). This approach of lexical structuring will also support the treatment of unknown words (Ludewig 1993).

Dialogue Management

The dialogue management of the PROMISE system plays an important part in the realization of interactive dialogue exercises with "free" learner input which are embedded in situational contexts. The dialogue manager guides the learner through a dialogue without the necessity of anticipating the learner's answers word by word, but rather on the basis of their semantic characterisations. On the basis of this component which relies on the output of the syntactic analyzis the realization of communicative exercise types - as demanded in the German overall guidelines for English language teaching (for further details see: Nealon 1992) - became possible. This component operates on a knowledge base ("dialogue script", cf. Fig. 6) which contains knowledge about structured dialogues (like emergency call, sales talk etc.). The knowledge base is organized like a transition graph that comprises all (predictable) information states that can be reached within the discourse. In Fig. 5 we show some sample entries for the information states named "meldung" (engl.: report), "injured" and "car". Following the name of the information state a feature term (e.g. *{PRED:KOLLISION agens:CAR patiens:AMPEL}*) describes the necessary "information content" which must be given by the learner in order to print the following NL-output or to activate a Prolog predicate (like *auseinander* in the last clause) and to go to a new information state.

When the student's input has been analyzed by the parser, the dialogue manager uses parts of the functional description to build up a predicate-argument-representation of the input sentence. With this kind of representation a semantic evaluation (sortal check) of the learner’s input is done on the basis of the situation specific ontologies (see section: Error Analysis). In a next step the dialogue manager looks for an edge in the dialogue transition graph which leads to an other information state. The semantic representation of the learner input is evaluated to see whether the conditions for changing to a new state are fulfilled, i.e.whether the learner has mentioned all important information which has been requested in the dialogue or not.

```
[»» (((meldung mehr *) *z) {PRED:KOLLISION agens:AUTO patiens:AMPEL}
                           [[So, a car hit a traffic light!][Can you see any injured people]] ((injured) *z)) ]

[»» (((injured) *z) {illokution:positiv}
                    [[All right ","] ["we'll" send you an ambulance] [Please "," tell me "," where you are .]] ((place) *z)) ]

[»» ((car *z) {agens:*lo position:*rel ref_obj:*ro}
              [(auseinander car car1 [*lo *rel *ro] *spruch *wei) *spruch] (*wei *z)) ]
```

Fig. 6: entries in a dialogue script

Processing of spatial expressions

The learner's input is not exclusivly analyzed syntactically and semantically but also on the basis of spatial knowledge in a given situation. The processing of spatial expression is used in PROMISE for training the use of spatial prepositions (like *on the left* of and *in front of)*. Within our accident scenario the learner should assist a policeman in completing a sketch that shows the traffic situation as it was just before the accident (cf. Fig. 7). As a result of the interpretation of a spatial expression an object in a sketch (depicting the current situation on the screen) can be localized (cf. Krüger forthcoming). Sketches are represented in the G_SBS-language (cf. Fig. 8) which is an extension of the Prolog environment G_LOG (Gust 1994). G_SBS permits the connection of graphics and formal descriptions of objects and events (e.g. in terms of Prolog clauses of a predicate *objekt* in Fig. 8). Thus, the system is able to communicate with the learner - in addition to dialogues - via sketches.

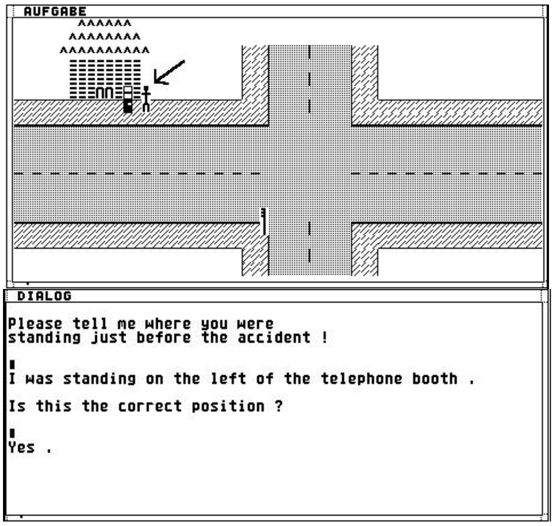

Fig. 7: Completing a sketch

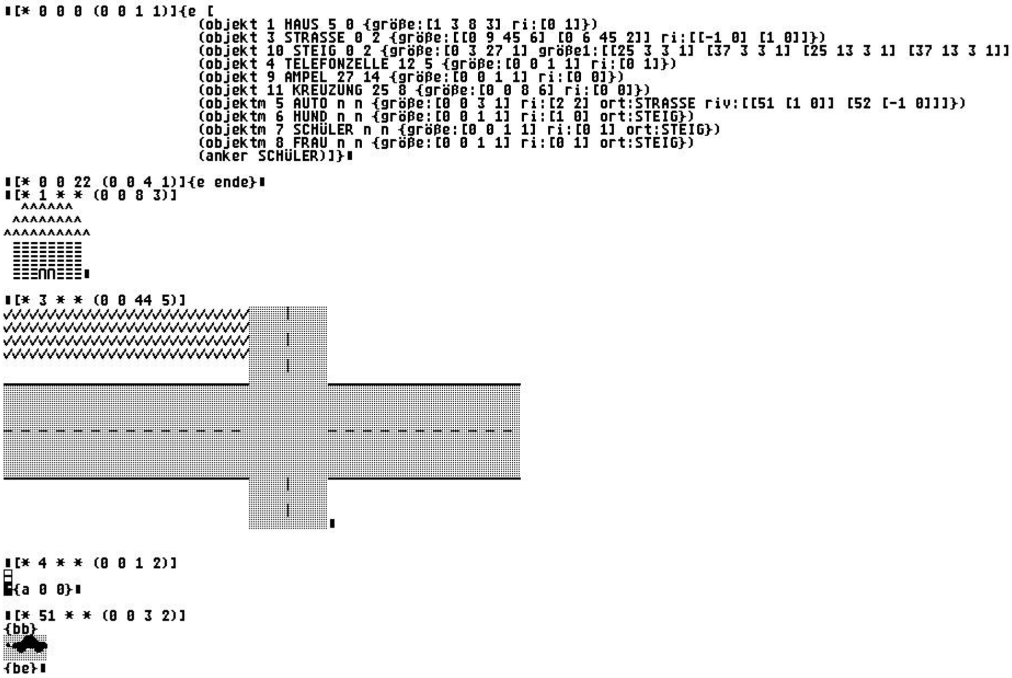

Fig. 8: (Parts of a) Representation of a situation in G_SBS

Outlook

Creative language behaviour cannot be dealt with standard technology. PROMISE shows that on the basis of Computational Linguistics and Artificial Intelligence techniques substantial knowledge based CALL systems are within reach in which not only classical exercise types gain flexibility but also new exercise types become possible.

To meet the requirements of communicative language teaching/learning future CALL systems must be capable to put the learner in authentic situations and to communicate with the learner in the foreign language. For this reason. multi media approaches and CL & AI technology have to be combined. The latest advances in speech recognition and the modularity of the PROMISE system give cause for hope that exercises will become possible in which the learner can authentically train his/her language use via spoken language.

But to be crowned with success the recent technological development demands new didactic concepts to be worked out. These concepts can only be achieved by interdisciplinary research.

References:

Bresnan, J.: The Mental Representation of Grammatical Relations. Cambridge, Massachusetts 1982.

Ellis, R.: Understanding second language acqusition. Oxford 1985.

Gust, H.: G_LOG Benutzerhandbuch. Software Dokumentation. University of Osnabrueck 1992.

Gust, H. / Ludewig, P / Krüger, A. Kronenberg, F.: CAVOL Projektbericht - Computer Assisted Vocabulary Learning. Bericht des Instituts für Semantische Informationsverarbeitung. University of Osnabrueck 1993.

Gust, H. / Ludewig, P. / Reuer, V. / Unsöld, R.F.: Evaluation auf dem Markt verfügbarer Software zum computergestützten Spracherwerb. Study on behalf of the IBM Germany Information Systems. University of Osnabrueck 1994.

Gust, H. / Krüger, A.: G_SBS Benutzeranleitung. Software Dokumentation. University of Osnabrueck. 1994.

Hötker, W / Ludewig, P. / Kanngießer, S.: DYLEX: Integration unterschiedlicher lexikalischer Ressourcen. Berlin 1991.

Hötker, W. / Ludewig, P.: Unterstützung der Wiederverwertung lexikalischer Ressourcen durch dynamische Wissensstrukturierung. In: KI 4/94.

Kronenberg, F./ Krüger, A. / Ludewig, P.: Contextual Vocabulary Learning with CAVOL. In: Appelo, L., de Jong, F.M.G. (eds.): Proceedings of the Twente Workshop on Language Technology 7 - Computer Assisted Language Learning. Enschede 1994, pp. 47-56.

Krüger, A: Interaktives Lernen räumlicher Präpositionen mit dem Sprachlehrsystem PROMISE. Magisterarbeit im Studiengang Computerlinguistik und Künstliche Intelligenz der Universität Osnabrück. Osnabrück, forthcoming.

Ludewig, P.: Inkrementelle wörterbuchbasierte Wortschatzerweiterungen in sprachverarbeitenden Systemen. Entwurf einer kontruktiven Lexikonkonzeption. DISKI 30. St. Augustin 1993.

Mellish, Ch.: Some chart based Techniques for Parsing Ill-Formed Input. 27th Annual Meeting of the ACL, Pittsburgh PA 1989, pp. 102-109.

Naumann, S. / Langer, H.: Parsing. Stuttgart 1994.

Nealon, H.: RECHERCHE. Version 1.4. Braunschweig 1992. (Software).

Niedersächsisches Kultusministerium (ed.): Rahmenrichtlinien für das Gymnasium Klassen 7-10, Englisch. Hannover 1989.

Puppe, F.: Intelligente Tutorsysteme. In: Informatik-Spektrum, Vol. 15 Issue 4. Berlin 1992.

Sauer, U.: Fehlersensitives Parsing in Sprachlehrsystemen mit PACO11G. PROMISE Abschlußbericht. University of Osnabrueck 1994.

Sauer, U.: CALL mit PACO11G. Modellierung und Implementierung sprachlichen Wissens für Sprachlehrsysteme. Expected to be published in: Proceedings der 19. Jahrestagung der Gesellschaft für Klassifikation: Lernen und Wissen - Strukrurelle Aspekte, Quantitaive Methoden und aktuelle Anwendungen. 1995.

Schwind, C.: Error analysis and explanation in knowledge based language tutoring. In: Appelo, L., de Jong, F.M.G. (eds.): Proceedings of the Twente Workshop on Language Technology 7 - Computer Assisted Language Learning. Enschede 1994, pp. 77-92.