NON-EXPERT ACCESS TO UNIFICATION BASED SPEECH UNDERSTANDING

Tom Brøndsted

Center for PersonKommunikation
Fredrik Bajers Vej 7A-5
Institute for Electronic Systems,
Aalborg University
DK-9220, Aalborg Ø, Denmark
tb@cpk.auc.dk

ABSTRACT

This paper describes the concepts behind a sub grammar design tool being developed within the EU-funded language-engineering project REWARD. The tool is a sub-component of a general platform for designing spoken language systems and addresses dialogue designers who are non-experts in natural language processing and speech technology. Yet, the tool interfaces to a powerful and “professional” unification grammar formalism that is interpreted by a corresponding natural language parser and in a derived finite state approximation form is used for constraining speech recognition. The tool performs some basic intersection and unification operations on feature sets to generate template-like rules complying with the unification grammar formalism.

1. INTRODUCTION

The EU-funded language engineering project REWARD (“Real World Applications of Robust Dialogue” LE1-2632) aims at developing a generic platform for rapid implementation of telephone services using spoken language dialogue technology [3]. A special characteristic of the project is the fact that it brings together platform developers with a number of user organisations so that the users directly influence the design of the platform [8], [9]. The users are non-expert in natural language processing and speech technology but are as experts in their own fields interested in designing and maintaining dialogue applications within special domains. Their domains are telemarketing, marketing research, travel agency, and hardware maintenance. Especially the users dealing with marketing research have prospects of developing many dialogues so that a main focus in the project is on prototyping, resource management etc.

For an overall description of the REWARD platform, refer to [3]. The present paper concentrates on the concept behind the sub grammar design tool, which allows non-experts users to design sub grammars in a unification-based formalism.

2. THE USERS

So far, the users have been designing dialogues that are extremely system directed. Consequently, they have had modest demands on sophisticated natural language processing (NLP) technology. Most recognition and understanding tasks are solved with plain keyword spotting, and "sub grammars" simply define the set of words that can be expected at the various system prompts in the dialogues. Hence, the parallel development of a sub grammar design tool based on a powerful and "professional" unification grammar formalism may seem to be inconsistent with the profile of the users.

The idea to choose a unification grammar as underlying (and partly "hidden") formalism in REWARD originally arose from a vague suggestion that users should be able to type in some word and phrase examples whereupon an intelligent sub grammar design tool automatically would do "the rest of the job". This suggestion was preferred to a solution allowing the users to draw grammar networks in a graphical environment (cf. [3]). However, such a tool obviously presupposes a kind of compound feature based knowledge base (or lexicon) together with software controlling basic operations performed on compound features. The operations in question are most notably *intersection* and *unification*. For instance, in order to conclude that a user means "any British city" when typing in the examples "London, Leeds", one needs to know that both London and Leeds are British cities (the intersection of the feature sets characterising the two words). Further, one has to know that the intersection feature set "British city" also matches (unifies with) "Manchester", "Liverpool" etc.

Three further considerations consolidating unification grammar as the appropriate choice should be mentioned:

- Firstly, there is a significant trend within spoken dialogue technology away from system-directed and towards more mixed initiative based dialogues. Rigid “system prompts” may guide untrained users through dialogues, however user trials reported in e.g. [10] show that trained users knowing of the systems functionality attempt to disregard prompts and proceed directly with their requests. Mixed-initiative systems allowing users to say “everything” at any time obviously presuppose more sophisticated NLP-technology than system-directed ones.

- Secondly, there is hardly any application with a simple 1:1 relation between keyword and domain specific “meaning”. The word “London” as e.g. airport-indication is ambiguous (Gatwick? Heathrow? ...), whereas, for instance, “Copenhagen” and “Kastrup” are synonyms. It is obvious to describe such relations in a declarative, feature-based formalism presupposing exhaustive parsing (returning all found meanings).

- Finally, unification grammars have (in many variants) been state of the art within computational linguistics for almost two decades.

3. UNIFICATION APPROACH

The linguistic components of the REWARD platform consist of

- A suite of general unification based natural language processing modules (implemented in C++).
- The window based sub grammar design tool (implemented in Java) providing non-experts access to *a part of* the functionality of the general modules.

3.1. General Modules

The general modules consist of a number of separate programs the core of which are a natural language parser and a grammar converter deriving language models (finite state grammar networks) to be used by continuous speech recognisers from the grammar formats used for parsing. Together, the parser and converter constitute two core modules of a unification-based speech understanding system, cf. figure 1.

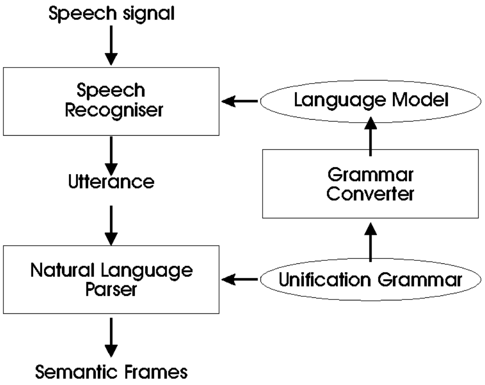

Figure 1. Unification based speech understanding.

Formally, semantic frames returned by the parser are represented as lisp-like predicate-argument structures that can be nested, e.g. *date(day(30),month(11),year(1998))* or even simpler *recognised_word(Sidney).*

The general modules, which are partly rooted in The Danish Spoken Language Dialogue Project [1], have been designed to enable integration into various platform architectures with varying requirements for NLP. For instance, the modules are also integrated into the "Intellimedia Workbench" described in [4] and [5]. The parser and converter are described in greater detail in [5], [6], and [7].

A number of parsing and recognition grammar formats are supported. Currently, the strongest parsing grammar format is an Augmented Phrase Structure Grammar (APSG) formalism based on a subset of the EUROTRA User Language developed originally for Machine Translation, however sufficient general to meet other NLP demands as well [2]. The recognition grammar applied in REWARD is a format supported by the recogniser being developed by the project partner Vocalis Ltd.

3.2. Grammar Formalism.

The sub grammar design tool assists non-expert users in implementing sub grammars using *a subset of* the APSG formalism supported by the general modules.

The core of the APSG formalism is the feature set, that is a set of features enclosed in '{' '}' each of which consists of an attribute assigned to a value with the '=' operator. A domain dependent lexical rule for e.g. the city "Manchester" may look like:

{lex=Manchester, cat=city, country=UK, airport=MAN}.

Such a rule can easily be derived from an attributed database available in a user organisation selling flight tickets. The possibility of using variables (often called "empty features") pointing to shared structures allows values to be percolated from child to parent nodes. For instance, if the lexical rule for "Manchester" is unified with the right-hand-side of the structure building rule (where variables are indicated by labels preceded by '$'):

{cat=DUMMY, destination_airport=$D}
  [{cat=city, airport=$D}
  ].

the result is the new structure:

{cat=DUMMY, destination_airport=MAN}
  [{lex=Manchester, cat=city, country=UK,
  airport=MAN}
  ].

The full APSG formalism includes four types of rules:

(i) Axioms (denoting the possible start features of the grammar).

(ii) Structure building rules

(iii) Lexical rules, and

(iv) Semantic mapping rules.

The rules (i)-(iii) are used to build syntactic structures (so-called parse trees) analysing the recognised utterances. The semantic mapping rules (iv) do not have any direct counterpart in the EUROTRA User Language (which, as mentioned above, is the pattern of the APSG format). However, there is some resemblance with the so-called "filter rules" (f-rules) of this framework. A mapping rule consists of (a) a condition describing a syntactic sub structure to look for and (b) an action describing the semantic frames to create if the condition is met.

In the APSG format, mapping rules are defined as facultative entries. If a sub grammar does not include mapping rules, semantic frames are derived from the (percolated) values found in the top of the parse tree. This simpler method of semantic extraction is applied in sub grammars implemented with the tool described in the next section. Presupposing a sub grammar labelled "reservation" consisting of the two rules mentioned above and an axiom *{cat=DUMMY}*, parsing of the utterance “Manchester” results in the semantic frame structure:

reservation(destination_airport(MAN))

The uppermost predicate is the name of the sub grammar. At points in a dialogue where two or more sub grammars are active simultaneously, the sub grammar name may be used for branching the dialogue flow. The arguments of the sub grammar name are the features found in the top node of the parse tree with the exception of the redundant axiom (for a detailed description of the supported APSG formalism, refer to [5] p. 70 ff.).

3.3. Sub Grammar Design Tool.

The main aim of the sub grammar design tool is to enable non-expert users implementing sub grammars using a subset of the unification grammar formalism described above and to derive language models from these sub grammars to be used for speech recognition. The tool and its working environment is shown in figure 2:

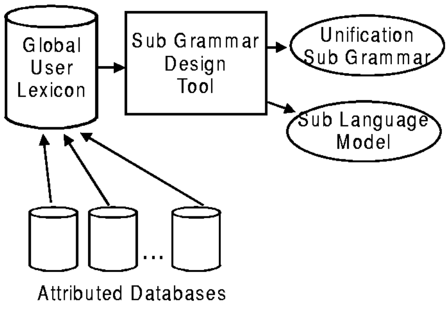

Figure 2. Sub Grammar Design Tool

The tool allows the users to implement such sub grammars within a minimum of time. Two databases, a Common Grammar Library and a Global User Lexicon, both of which are established and maintained by the user, manage reusable resources:

- The *Common Grammar Library* simply covers up the trivial fact that the user can maintain a library of common sub grammars (parsing "dates", "hours", "yes-no answers" etc.) to be reused in other dialogues.
- The *Global User Lexicon* has a more important role as it provides the tool with information necessary for generalising list of words selected by the user and for generating expressions and semantic values for each word of the generalised list. This implies that a user can design a sub grammar e.g. describing routes from domestic airports in the UK to domestic or foreign airports literally with a few "mouse clicks" and a minimum of typing. .

The sub grammar design tool performs some very basic operations on feature sets to generalise from word examples typed in or selected from the global lexicon by the user. The operations are: Intersection, unification, and disjoining. To illustrate the generalising capability of the tool, we will elaborate the reservation sub grammar described above (sec. 3.2) and start from a global user lexicon which may have been imported from an attributed database available in a user organisation selling flight tickets:

{lex=Manchester, cat=city, country=UK, airport=MAN}.
{lex=Leeds, cat=city, country=UK, airport=LBA}.
{lex=London, cat=city, country=UK, airport=LHR}.
{lex=London, cat=city, country=UK, airport=LGW}.
{lex=London, cat=city, country=UK, airport=STN}.
{lex=Heathrow, cat=city, country=UK, airport=LHR}.
{lex=Copenhagen, cat=city, country=DK, airport=CPH}.
{lex=Kastrup, cat=city, country=DK, airport=CPH}.
{lex=Aalborg, cat=city, country=DK, airport=AAL}.
....

To create a feature set unifying with e.g. British cities, the user simply has to select two or more appropriate lexical items from the global lexicon. Selecting “Manchester” and “Leeds”, leads to the expression *{cat=city, country=UK}* which is the ***intersection*** of the two lexical rules. As the intersection also ***unifies*** with “London” (all three rules), “Heathrow” etc., the selected list can be generalised to include all British Cities. ***Disjoining*** the selected feature sets, leads to the features *{lex=$A, airport=$B}* which are candidates for being percolated to the semantic frames. For instance, when parsing “Manchester”, we may be interested in the *lex* value *Manchester* or the *airport* value *MAN* or in both. Leaving everything in the default state suggested by the tool, the initial two “mouse clicks” by the user will lead to a sub grammar consisting of a number of lexical rules describing British cities, a “hidden” axiom *{cat=DUMMY}*, and the following template-like structure building rule:

```
{cat=DUMMY, lex=$A, airport=$B }
    [{cat=city, country=UK, lex=$A, airport=$B }
    ].
```

where also the head feature *{cat=DUMMY}* is “hidden” for the user.

Parsing 1. “Manchester”, 2. “London”, 3. “Heathrow”, and 4. “Aalborg” respectively, results in the following frames where 2 is ambiguous, 2 and 3 are partly synonyms, and 4 cannot be parsed:

1. reservation(lex(Manchester),airport(MAN))

2. *reservation(lex(London),airport(LHR))*
   reservation(lex(London),airport(LGW))
   reservation(lex(London),airport(STN))
3. *reservation(lex(Heathrow),airport(LHR))*
4. *NULL*

Removing *airport=$B* from the structure building rule will disambiguate "London", however does not solve the actual problem in the dialogue, namely determining which London airport (Heathrow, Gatwick, Stansted ...) the speaker has in mind. This problem, of course, is not linguistic but must be dealt with in the dialogue description.

Designing sub grammars for phrases largely follows the pattern described above. The tool supports the direct entry of function words such as "from", "to" without any reference to predefined lexical items. On the APSG level, such entries correspond to "ad hoc feature sets", written <from>, <to> and internally expanded *{lex=from, cat=from}*, *{lex=to, cat=to}*. Basically, ad hoc feature sets express that the grammar designer refers to "this word" and takes no interest in the "correct" linguistic properties of the word.

A sub grammar e.g. describing routes from domestic airports in the UK to any other airport may look

*{cat=DUMMY, **source**=$A, **destination**=$B }*
*[<**from**>, {cat=city, country=UK, airport=$A },*
*<**to**>, {cat=city, airport=$B}*
].

where everything has been generated with mouse selecting or mouse deselecting except for the four bold entries ***source, destination, from, to*** which must be typed.

4. CONCLUSION

The REWARD project has followed a double-threaded working-plan where the user organisations have implemented dialogues using existing dialogue creation tools developed by the project partner Vocalis Ltd. and in parallel have assisted in the specification and design phase of a new suite of tools. This working-scheme, which is due to a revision of the project in the spring of 1997 (cf. [3]), has till now left no time for an actual user evaluation of the new environment, including the sub grammar design tool presented in this paper. However, some changes of the tool as a consequence of user responses should be reported.

A version of the sub grammar design tool supporting more of the APSG formalism than the subset indicated in section 3.3 was reported by the users to be too complex. Consequently, the tool is now being redesigned and simplified to correspond to the template-like functionality described in this paper. Further, the principal decision was taken that an access to the full functionality of the linguistic components is only possible when coding grammars directly in the native APSG format with a text editor.

The users also reported that they have access to attributed databases, which can be used for establishing the Global User Lexicon. Consequently, we think that this lexicon can turn out to be an extremely useful resource managing facility of the platform.

We believe that the unification approach to grammar design by non-experts described in this paper is a better solution than allowing users "drawing" label based grammar networks in a graphical environment. The expressive power of unification grammar is superior to that of label based transition networks and allows declarative solutions to semantic representations.

5. REFERENCES

[1] A. Baekgaard, O. Bernsen, T. Brøndsted, P. Dalsgaard, H. Dybkjær, L. Dybkjær, J. Kristensen, L.B. Larsen, B. Lindberg, B. Maegaard, B. Music, L. Offersgaard, C. Povlsen: "The Danish spoken language dialogue project - a general overview". *ESCA WS on Spoken Dialogue Systems: Theory and applications, Vigsø, Denmark 1995, pp. 89-92.*

[2] A. Bech: "Description of the Eurotra Framework". *C. Copeland, J. Durand, S. Krauwer, B. Maegaard (eds). The Eurotra Formal Specifications. Studies in Machine Translation and Natural Language Processing, Volume 2, 1991.*

[3] T. Brøndsted, B. Bai, J.Ø. Olsen: "The REWARD Service Creation Environment. An Overview". *These Proceedings.*

[4] T. Brøndsted, L.B., M. Manthey, P. Mc Kevitt, T. Moeslund, K.G. Olesen: "The Intellimedia WorkBench - a Generic Environment for Multimodal Systems". *These Proceedings.*

[5] T. Brøndsted, P. Dalsgaard, L.B. Larsen, M. Manthey, P. Mc Kevitt, T. Moeslund, K.G. Olesen: "A Platform for developing Intelligent MultiMedia Applications". *Technical Report R-98-1004. Institute of Electronic Systems, Aalborg University 1998.*

[6] T. Brøndsted: "The Linguistic Components of the REWARD Dialogue Creation Environment and Run Time System". *IEEE, IVTTA Turin 1998. In press.*

[7] T. Brøndsted: "The Natural Language Parsing Modules in REWARD and IntelliMedia 2000+". *S. Kirchmeier-Andersen, H.E. Thomsen (eds.): Proceedings from the Danish Society for Computational Linguistics (DALF), Copenhagen Business School, Dep. of Computational Linguistics, 1998. In press.*

[8] K. Failenschmid: "Spoken Dialogue System Design – The Influence or the Organisational Context on the Design Process". *IEEE, IVTTA Turin 1998. In press.*

[9] K. Failenschmid, S. Thornton: "End-user driven Dialogue System Design". *These Proceedings.*

[10] L.B. Larsen: "Investigating a Mixed-Initiative Dailogue Management Strategy". *IEEE, ASRU Santa Barbara 1997, pp. 65-71.*