Robust Multimodal Discourse Processing

Norbert Pfleger and **Ralf Engel** and **Jan Alexandersson** *
DFKI GmbH
D-66123 Saarbrücken, Germany
{pfleger,rengel,janal}@dfki.de

Abstract

Providing a generic and robust foundation for the correct processing of short utterances is vital for the success of a multimodal dialogue system. We argue that our approach based on a three–tiered discourse structure in combination with partitions provide a good basis for meeting the requirements in such a system. We present a detailed description of the underlying representation together with some show cases.

1 Introduction

Continuous recognition of speech and gesture puts extra requirements on the part of a dialogue system concerned with semantic processing. The successful processing of tasks, like selecting the correct or best analysis out of competing ones belong to the basic processing abilities, but also resolving ambiguities are of great importance.

Utilizing some kind of discourse context is vital for disambiguation, especially for vague, reduced, or partial expressions, e. g., elliptical or referential expressions. A typical example is depicted in figure 1, where the correct interpretation of U2 and U3 relies on an elaborated discourse model.

In this paper we argue that our approach to discourse modelling which has borrowed its main inspiration from (Luperfoy, 1992; Salmon-Alt, 2000) and (Wahlster, 2000) provides a generic, simple and yet powerful basis for multimodal discourse modelling and processing. We construe an unified representation of user and system contributions for mono- as well as multimodal systems supporting the resolution of elliptical and cross-modal referential expressions in a simple and generic manner. Our work described here (see also (Pfleger, 2002)) contributes to the DFKI core dialogue backbone for multimodal dialogue systems.

U1: *I'd like to see a film tonight.*
S1: [Displays a list of films] *Here* [↗] *you see a list of the films running in Heidelberg.*
U2: *Hmm, none of these films seems to be interesting... Please show me the TV program.*
S2: [Displays a list of broadcasts] *Here [↗] you see a list of broadcasts on TV tonight.*
U3: *Then tape the first one for me!*

Figure 1: Dialogue excerpt 1

2 The Dialogue Backbone

Our long term effort is to develop a reusable dialogue backbone. Though the present status has been heavily coloured by the SMARTKOM project (www.smartkom.org), parts of the backbone are and have been used in several mono- as well as multimodal dialogue systems including monomodal and multimodal as well as typed and spoken input/output (including gesture and facial expressions), e. g., Miamm (www.miamm.org),

*The research presented here is funded by the German Ministry of Research and Technology under grant 01 IL 905. The responsibility for the content is with the authors. We would like to thank Stephan Lesch and Massimo Romanelli for their help with implementation and evaluation.

Comic (`www.hcrc.ed.ac.uk/comic`) and NaRaTo, an industrial project aiming at a typed NL interface for the ARIS tool-set (see `www.ids-scheer.com`).

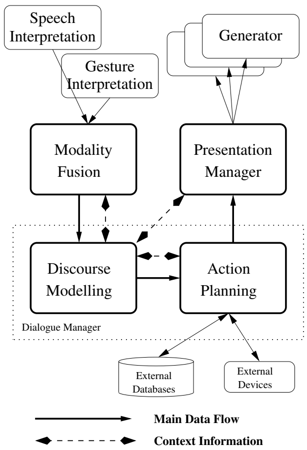

Figure 2: Architecture of the backbone

We use a pipe-line approach enhanced with several request-response interfaces (forward and backward) between different components. (Löckelt et al., 2002) contains a quite detailed description of our architecture. In our current system, some additional interfaces were introduced, e. g., the output from the system is completely processed by the discourse modeller (henceforth DIM) supporting for instance the processing of cross-modal referring expressions (see also (Pfleger et al., 2003)).

The main communication representation within the backbone is the so-called *intention lattice* containing instances of our domain model, syntactic information and scoring information etc. Our domain model (Gurevych et al., 2003) is a hand crafted ontology encoded in OIL-RDFS (Fensel et al., 2001). Basis for its development are the ideas of (Russel and Norvig, 1995; Baker et al., 1998). The current version comprises more than 700 concepts and about 200 relations. We apply *closed world reasoning* - everything that can be communicated is encoded explicitly in our ontology.

Instances of the top-level types in the ontology are called *application objects*. These are complete descriptions of actions, such as accessing a database or zooming a map. Parts of application objects are called *subobjects* which are more often atomic objects, e. g., channels and cities, although they might be structured for example as time expressions and seat collections (which in our ontology have cardinality and a set of seats). Some subobjects are meaningful for the action planner (henceforth AP), these subobjects are called *slots*. A slot is a pair consisting of a name and a path. Slots are unique, so given a slot name we can uniquely find its corresponding path (and vice versa). The effect of the presence of a slot in an user intention is described in (Löckelt et al., 2002).

Central to this paper is the notion of *path*. Some paths are defined by the *action plans* which, in a sense, connect to the ontology by pointing into it. Action plans define *states* and *slots*. Some states are called *goals* and in each plan there is at least one goal. For each goal there is a path called *goal path*. The goal path is usually not a slot. A slot is a pair consisting of a symbol and a path. Although the plans are mainly used by AP they are additionally utilized by several components of the system.

We give a very short recapitulation of the processing within the backbone (see also (Löckelt et al., 2002)): user actions - speech and gesture - are analyzed and interpreted and finally brought together in MF. DIM enriches the hypotheses with contextual information and - based on the different scores - finally selects the most probable hypothesis and sends it to AP. Depending on input and the dialogue state, AP may choose to access some external devices before the modality fission is requested to generate and present the system reaction.

The analysis components of our backbone score each hypothesis in different ways; a task we call *validation*. Scoring is based on the knowledge in the respective modules and on different views of the user intention. Whereas, e. g., the language analysis computes a score based on how *linguistic* a certain path in the word lattice is, DIM computes its score based on how well the hypothesis fits the discourse context (see (Pfleger et al., 2003)).

2.1 Language Understanding

The task of the language understanding component is to analyze the different hypotheses of the speech recognizer and to assign to them a semantic meaning in terms of the domain model. We use a template based semantic parser (henceforth SPIN (Engel, 2002)). The basic idea of the used approach is to apply so-called *templates* to a working memory (WM) in a depth-first search fashion. Initially, the WM is filled with the recognized words. In a first phase, templates capable of transforming the initial words to simple objects (typically subobjects) are applied. Then, these objects are combined to more complex objects (typically application objects or nested subobjects). In case some objects (or words) contribute nothing to the final interpretation, a lower score is assigned to that particular interpretation.

Referring expressions are internally represented as subobjects together with a feature called `reference`. It is filled with information about the characteristics of referring expression, e. g., definiteness and/or position in a list.

Referring expressions containing no type information of the referenced object, like *this one*, are initially given the most common type - PHYSICALOBJECT. However, it might, during template application, be refined due to the intrasentential context. For instance, if a template responsible for creating an application object of type INFORMATIONSEARCH expects a domain object of type BROADCAST and the WM contains a domain object of type PHYSICALOBJECT, then the type of the object in the WM is (destructively) refined to the type BROADCAST.

2.2 Modality Fusion

The task of the modality fusion component is to combine and integrate multiple hypotheses produced by the analyzers for the different modalities. Pointing gestures are integrated into a speech recognition hypothesis containing deictic expressions by replacing the referring expression with the object associated with the gesture. A different strategy is applied if a gesture is recognized accompanying a spoken utterance without a deictic expression. In that case, the ontology is utilized in order to find possible insertion places.

In case a spoken utterance contains referring expressions that cannot be resolved with gestures, MF requests DIM and replaces the referring expression with the possible discourse objects.

Following (Nigay and Coutaz, 1993), we currently process *synergistic* input, i. e., a combination of coherent information from gesture and speech which can be mapped onto a single domain object and *exclusive* input, i. e., either the input is speech- or gesture-only.

3 Discourse Modelling

Context Representation: Our approach to discourse modelling is based on a generalization of (Luperfoy, 1992) together with some ideas from (Salmon-Alt, 2000) and (Wahlster, 2000). Following the ideas of (Luperfoy, 1992), we use a three-tiered context representation where we have extended her linguistic layer to a *modality* layer (see Figure 3). Additionally, we have adopted some ideas from (Salmon-Alt, 2000) by incorporating directly perceived objects and compositional information of collections. Basic discourse operations used in (Wahlster, 2000) has been further developed (see (Alexandersson and Becker, 2003)). For more details please see (Pfleger, 2002; Pfleger et al., 2003)

The advantage of our approach to discourse representation lies in the *unified* representation of discourse objects introduced by the different modalities. As we show below, this not only supports the resolution of elliptical expressions but allows for, i. e., cross-modal reference resolution. The context representation of the discourse modeller consists of three levels:

- **Modality Layer:** The objects at the modality layer (MOs) encapsulate information about the concrete realization of referential objects. We employ three types of objects: (i) *Linguistic Objects* (LOs) providing information about linguistic features, e. g., number and gender, (ii) *Visual Objects* (VOs) providing information about the position on the screen, and (iii) *Gesture Objects* (GOs) providing no realization information but which are used to group objects together. Each modality object is linked to a corresponding discourse object and shares its information about its concrete realization with the discourse object (see figure 4). Important for this paper is that a MO provides information about its original position within the event structure - its path in the corresponding application object it is embedded in.

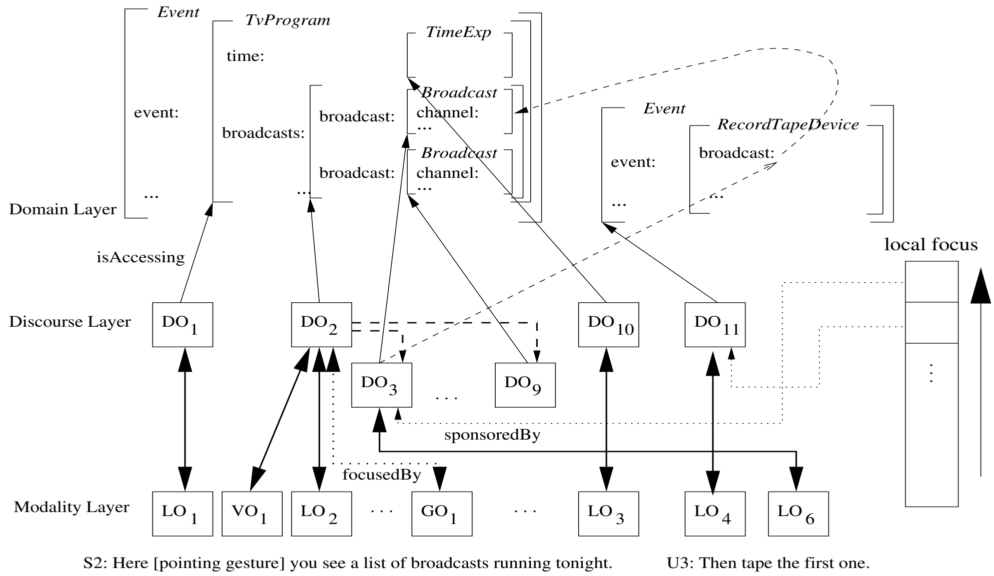

Figure 3: The Multimodal Context Representation. The dashed arrow(s) indicates that the value of the broadcast in the (new) structure to the right is shared with that of the old one (to the left).

• **Discourse Object Layer:** This layer contains discourse objects (DOs) which serve as referents for referring expressions. A DO is created every time a concept is newly introduced into the discourse by speech and for directly perceived concepts, e. g., graphical presentations (Salmon-Alt, 2000).

Two classes of information are used by a DO (i) modality specific information, and (ii) domain information. For each concept introduced during discourse there exists only one DO independent of how many MOs mention this concept. Each DO is hence unique.

The compositional information of DOs representing collections of objects is provided by *partitions* (Salmon-Alt, 2000).

Partitions represent collections of objects and are based either on perceptive information, e. g., the list of broadcasts visible on the screen, or discourse information stemming from grouping discourse objects. The elements of a partition are distinguishable from one another by at least one *differentiation criterion*. Yet, one element alone of a partition may be in focus, according to gestural or linguistic salience. Figure 4 depicts a sample configuration of a discourse object (`DO2`) with a partition.

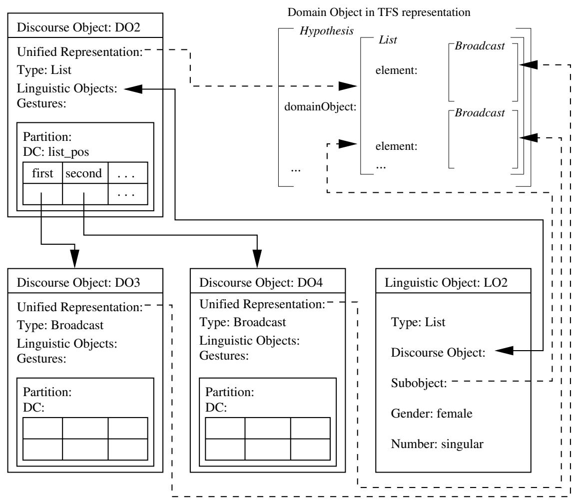

Figure 4: Discourse Objects

• **Domain Object Layer:** The domain object layer encapsulates the instances of the domain model and provides access to the semantic information of objects, processes, and actions. Initially, the semantic information of a DO is defined by a subobject (possibly embedded in an application object) representing the object, process, or action it corresponds to. This information is only accessed by the DO. However, the semantic information of a DO might be extended as soon as the object is accessed again. Consider for example DO42 initially representing a movie with the title “The Matrix” (created by the user request *“When will the movie The Matrix be shown on television?”*). The system will respond with presenting a list of broadcasts of the movie “The Matrix” (accompanied with information specific to the different broadcasts, like time, database key, channel etc). Now, if the user selects one of the movies - *“tape this [↗] one”* - the initial information of DO42 will be extended with this additional information.

3.1 Modelling Attentional State

We differentiate between two focus structures restricting access to objects stored in the discourse model: (i) a *global* focus structure and (ii) a *local* focus structure. The former represents the topical structure of discourse and resembles a list of focused items - *focus spaces* - ordered by salience. In SMARTKOM, the global focus is imposed by the action planner providing a flat structure of discourse in terms of discourse topics. A focus space covers all turns belonging to the same topic and enables access to a corresponding local focus structure (see also (Carter, 2000)).

A local focus structure provides and restricts access to all discourse objects that are antecedent candidates for later reference. Also on this level, the content of the structure, i. e., discourse objects, are ordered by salience. For each user or system turn, the local focus structure for this topic is extended with all presented concepts (see also figure 3).

3.2 Initiative–Response Units

To further restrict and provide access to referents we use simplified, flat *initiative–response units* or IR-units (Ahrenberg et al., 1991), mirroring who is having the initiative. This information has effects on the interpretation of partials (Löckelt et al., 2002). Our flat treatment is robust and despite its simpleness capable of processing one-level subdialogues, i. e., cases where, if the user has the initiative, the system imposes a sub-dialogue by stealing the initiative by requesting additional required information.

4 Discourse Processing

We now turn to the task of processing, e. g., referring expressions. In this section we describe how the structures presented in the last section are utilized.

4.1 Context Dependent Interpretation

Our main operations for the manipulation on instances of our domain model is unification and a default unification operation we call OVERLAY (Alexandersson and Becker, 2003). The starting point for the development of the latter operation was twofold: First we view our domain model as typed feature structures and employ closed world reasoning on their instances. Second, we saw that adding information to the discourse state can be done using unification as long as the new information is consistent with the context. However, as the user changes her mind and specifies competing information, unification will fail. Instead, we saw the need for a non-monotonic operation capable of overwriting parts of the old structure with the new information *and* at the same time keep the information still consistent with the new information. The solution is default unification, e. g., (Carpenter, 1993; Grover et al., 1994) which has proven to be a powerful and elegant tool for doing exactly this: Overwriting old, contextual information - *background* - with new information - *covering* - thereby keeping as much consistent information as possible.

U4: *What is on TV tonight?*
S4: [Displays a list of broadcasts] *Here* [↗] *you see a list of the broadcasts running tonight.*
U5: *What is running on CBS?*
S5: [Displays a list of broadcasts for CBS tonight] *Here* [↗] *you see a list of the broadcasts running tonight on CBS.*
U6: *and CNN?*
S6: ...

Figure 5: Dialogue excerpt 2

We distinguish between full or partial utterances. Example of the former is a complete description of an user action, e. g., U4 in Figure 5, whereas partial utterances often but not necessarily are elliptical responses to a system request. We handle these cases differently as described below.

4.2 Full Utterances

For, e. g., task-oriented dialogues, there are many situations where information can and should be inherited from the discourse history as shown in the dialog excerpt in figure 5. Due to spatial restrictions on the screen it may be impossible for the system to display every broadcast for all channels, e. g., in S4. The system therefore chooses some broadcasts of some channels. Clearly, the intention in U5 is to ask for the program on CBS *tonight* thus requiring the system to inherit the time expression from U4. Default unification provides an elegant mechanism for inheriting information from the background for these cases. Full utterances are processed by traversing the global focus structure and pick the focused application object in each focus space (if any). In figure 5, default unifying U5 (covering) with U4 (background) results in *what is running on TV on channel CBS tonight*.

4.3 Partial Utterances

For the interpretation of partial utterances (henceforth partials) we gave a detailed description in (Löckelt et al., 2002). The general idea is to convert the partial to an application object - referred to as bridging - and then use this application object as covering and the focused application object as background. There is, however one more challenge we have to face: resolving referring expressions. Given resolved referring expressions *and* the correct bridging we can use the basic processing technique as described above in section 4.2. Next, we concentrate of the latter whereas the former task is described in section 4.5.

4.4 Resolving Referring Expressions

There have been many proposals in the literature for finding antecedents for referring expressions, e. g., (Grosz et al., 1995). These approaches typically advocate a search over lists containing the potential antecedents where information like number, gender agreement etc. are utilized to narrow down the possible candidates. Our approach to reference resolution is a bit different. Additionally, in a multimodal scenario, the modality fusion first has to check for accompanying pointing gestures before accessing the discourse memory. In case of a missing gesture, DIM receives a request from modality fusion containing a subobject as specific as possible inferred by the intra-sentential context. This goes together, if possible, with the linguistic features and partition information. DIM searches the local focus structure and returns the first object that complies with the linguistic constraints in the respective LO and unifies with the object of the corresponding DO.

We resolve three different referring expressions:

- **Total Referring Expressions:** A *Total Referring Expression* is the condition where a referring expression co-refers with the object denoting its referent. Those referring expressions are resolved through the currently active local focus. The first discourse object that satisfies the type restriction and that was mentioned by a linguistic object with the same linguistic features is taken to be the intended referent. In this case, there is a linguistic sponsorship relation established between the referring expression and its referent.

If no linguistic sponsorship relation can be established, DIM tries to establish a discourse sponsorship relation. This condition is characterized by a mismatch between the linguistic features but the objects themselves are compatible (unifiable).

However, if both conditions are not fulfilled, the focused discourse objects (but only the ones most focused on) of the other global focus spaces are tested thereby searching for a discourse object that allows for a linguistic sponsorship relation.

- **Partial Referring Expressions:** In the case of a partial referring expression, the focus structures are searched for a discourse object that shows compositionality and satisfies (i) the differentiation criterion specified in the `partition` feature of the request, and (ii) has a discourse object - DO_i - in its value feature of the partition that satisfies the value feature of the partition (see figure 4). The first such DO_i is returned.
- **Discourse Deictic Expressions:** If the type of the referring expression cannot be identified, the focused discourse object of the currently active local focus is tested as to whether it shares the linguistic features with the request. If it does, that discourse object is taken to be the intended referent, otherwise the referring expression is inter-

$$\begin{bmatrix} \text{CinemaReservation} & \\ f1 & \begin{bmatrix} \text{T1} & \\ f2\ldots & \begin{bmatrix} \text{T2} & \\ seats & \begin{bmatrix} \text{Seats} & \\ Cardinality & 2 \\ seat & \begin{bmatrix}\text{Seat}\\ \ldots\end{bmatrix} \\ seat & \begin{bmatrix}\text{Seat}\\ \ldots\end{bmatrix}\end{bmatrix}\end{bmatrix}\end{bmatrix}\end{bmatrix}$$

Figure 6: An application object representing the reservation of two seats.

preted as being a discourse deictic one in which case the discourse object representing the last system turn is returned.

4.5 Partial Utterances Revisited

We return to the interpretation of partials again. After being processed by MF, an intention is now guaranteed to be either an application object or, as we will focus on now, a subobject representing a partial. Processing partials consists of two steps (see also (Löckelt et al., 2002)):

1: Find an anchor for the partial. If present, the anchor is searched for and possibly found under (i) the list of expected slots, (ii) in the local focus stack, or, finally, (iii) in the list of possible slots.

2: Compute the bridge by using the *path* in either expected or possible slots, or in the MOs in the local focus stack.

There are, however, some exception to this general scheme of which we provide two examples:

• **User provides too much information:** If the system has the initiative thus requesting for information, the user might provide more, still compatible information than asked for. A good example is the case where, during seat reservation for a performance, the system asks the user to specify where she would like to sit. The expected slot is in this case pointing to a seat. In our domain model the seat part of a set construction contains not just one seat but, e. g., a set of seats and cardinality. If the user contribution specifies something like “*Two seats here [↖]*[1]” then the user contribution will not fit the expectation.

[1]. . . where [↖] stands for an encircling gesture.

A representation of the reservation is schematically depicted in figure 6. The system asks for a specification of a seat, i. e., an information at the end of the path *f1:f2:. . . :seats:seat* which is a partial of type SEAT. The answer *contains* the expectation which, however, is embedded in an object consistent with the expectation. The correct processing of such an answer is to walk along the expectation until the answer is found. If this happens before the end of the expected path, the rest of the path (*seat*) has to be part of the subobject.

• **Manipulations of Sets:** For the correct processing of the example above, we had to extend OVERLAY with operations on sets, like union. Using almost the same example, we have the case where the user is not satisfied with the reservation and replies to the system request “*Is the reservation OK?*” with a modification containing manipulations of, in this case, a set of seats by uttering “*two additional seats here [↖]*”. The processing consists of two steps: (i) SPIN marks the seats in the intention hypothesis with a set modification flag. (ii) MF requests the focused set of seats from DIM and computes the allowed set of, in this case additional seats. (iii) The intention is now containing two seats (pointed at with the set modification flag) which are then processed by DIM in the same way as described above: The path from the root of the focused application object corresponding to the set of seats in the discourse memory. These are found in the local focus stack. After computing the covering, OVERLAY is performed where the sets of seats in the background and covering are unified with union.

5 Evaluation

The evaluation of DIM is part of a bigger adventure where we are seeking the answer to the following questions:

Ellipses and Anaphora: including cross-modal anaphora. In how many cases do we find the right antecedent for spoken referential expressions, e. g., “Tape it!”, “Tape this one”, “Tape the first”. In addition to speech recognition, the performance of MF and SPIN plays a central role.

Enrichment: Using default unification as basic operation for discourse has the drawback that sometimes too much - still consistent - contextual information is inherited. Consequently we are currently rather bothering about what *not* to inherit than what to inherit.

Score: Did the scoring from all components contributed to the selection of the correct hypothesis?

The evaluation is at the moment of writing not completed, but we hope for some indication of the system performance at the workshop.

6 Conclusions

We presented a generic robust discourse module which has been developed for and used in several mono- as well as multi-modal dialogue systems. Our "largest" system is a multimodal system for which about 50 different functionalities have been implemented, e. g., (Reithinger et al., 2003). During the development, we have tested the system on far more than 300 test dialogs. Our next, obligatory step is evaluation. Another topic for future development is more support for modality fission.

References

Lars Ahrenberg, Arne Jönsson, and Nils Dahlbäck. 1991. Discourse Representation and Discourse Management for a Natural Language Dialogue System. Research Report LiTH–IDA–R–91–21, Institutionen för Datavetenskap, Universitetet och Tekniska Högskolan Linköping, August.

Jan Alexandersson and Tilman Becker. 2003. The Formal Foundations Underlying Overlay. In *Proceedings of the Fifth International Workshop on Computational Semantics (IWCS-5)*, Tilburg, The Netherlands, February.

Collin F. Baker, Charles J. Fillmore, and John Lowe. 1998. The berkeley framenet project. In *Proceedings of COLING-ACL*, Montreal, Canada.

Bob Carpenter. 1993. Skeptical and credulous default unification with application to templates and inheritance. In A. Copestake E. J. Briscoe and V. de Paiva, editors, *Inheritance, Defaults and the Lexicon*, pages 13–37. Cambridge University Press, Cambridge, England.

David Carter. 2000. Discourse focus tracking. In Harry Bunt and William Black, editors, *Abduction, Belief and Context in Dialogue*, volume 1 of *Studies in Computational Pragmatics*, pages 241–289. John Benjamins, Amsterdam.

Ralf Engel. 2002. SPIN: Language understanding for spoken dialogue systems using a production system approach. In *Proceedings of 7th International Conference on Spoken Language Processing (ICSLP-2002)*, pages 2717–2720, Denver, Colorado, USA.

D. Fensel, F. van Harmelen, I. Horrocks, D. McGuinness, and P. F. Patel-Schneider. 2001. OIL: An ontology infrastructure for the semantic web. *IEEE Intelligent Systems*, 16(2):38–45.

B.J. Grosz, A. K. Joshi, and S. Weinstein. 1995. Centering: A Framework for Modelling the Local Coherence of Discourse. Technical Report IRCS Report 95-01, The Institute For Research In Cognitive Science, Pennsylvania.

Claire Grover, Chris Brew, Suresh Manandhar, and Marc Moens. 1994. Priority union and generalization in discourse grammars. In *32nd. Annual Meeting of the Association for Computational Linguistics*, pages 17–24, Las Cruces, NM. Association for Computational Linguistics.

Iryna Gurevych, Robert Porzel, Hans-Peter Zorn, and Rainer Malaka. 2003. Semantic coherence scoring using an ontology. In *Proceedings of the Human Language Technology Conference - HLT-NAACL 2003*, Edmonton, CA, May, 27–June, 1.

Markus Löckelt, Tilman Becker, Norbert Pfleger, and Jan Alexandersson. 2002. Making sense of partial. In *Proceedings of the sixth workshop on the semantics and pragmatics of dialogue (EDILOG 2002)*, pages 101–107, Edinburgh, UK, September.

Susan Luperfoy. 1992. The Representation of Multimodal User Interface Dialogues Using Discourse Pegs. In *Proceedings of ACL-92*, pages 22–31.

L. Nigay and J. Coutaz. 1993. A design space for multimodal systems: Concurrent processing and data fusion. In *Proceedings of INTERCHI-93*, pages 172–178, Amsterdam, The Netherlands.

Norbert Pfleger, Jan Alexandersson, and Tilman Becker. 2003. A robust and generic discourse model for multimodal dialogue. In *Workshop Notes of the IJCAI-03 Workshop on "Knowledge and Reasoning in Practical Dialogue Systems"*, Acapulco, Mexico, August.

Norbert Pfleger. 2002. Discourse processing for multimodal dialogues and its application in smartkom. Master's thesis, Universität des Saarlandes.

Norbert Reithinger, Jan Alexandersson, Tilman Becker, Anselm Blocher, Ralf Engel, Markus Löeckelt, Jochen Müeller, Norbert Pfleger, Peter Poller, Michael Streit, and Valentin Tschernomas. 2003. Smartkom - adaptive and flexible multimodal access to multiple applications. In *Proceedings of ICMI 2003*, Vancouver, B.C.

Stuart Russel and Peter Norvig. 1995. *Artificial Intelligence: A Modern Approach*. Prentice Hall, Englewood Cliffs, NJ.

Susanne Salmon-Alt. 2000. Interpreting referring expressions by restructuring context. In *Proceedings of ESSLLI 2000*, Birmingham, UK. Student Session.

Wolfgang Wahlster, editor. 2000. *VERBMOBIL: Foundations of Speech-to-Speech Translation*. Springer.