# SmartKom - Adaptive and Flexible Multimodal Access to Multiple Applications

Norbert Reithinger, Jan Alexandersson, Tilman Becker, Anselm Blocher, Ralf Engel, Markus Löckelt, Jochen Müller, Norbert Pfleger, Peter Poller, Michael Streit, Valentin Tschernomas

DFKI GmbH – German Research Center for Artificial Intelligence

Stuhlsatzenhausweg 3

D-66123 Saarbrücken, Germany

+49 681 302 5346

sk-info@dfki.de

## ABSTRACT
The development of an intelligent user interface that supports multimodal access to multiple applications is a challenging task. In this paper we present a generic multimodal interface system where the user interacts with an anthropomorphic personalized interface agent using speech and natural gestures. The knowledge-based and uniform approach of SmartKom enables us to realize a comprehensive system that understands imprecise, ambiguous, or incomplete multimodal input and generates coordinated, cohesive, and coherent multimodal presentations for three scenarios, currently addressing more than 50 different functionalities of 14 applications. We demonstrate the main ideas in a walk through the main processing steps from modality fusion to modality fission.

## Categories and Subject Descriptors
0.9: *Intelligent multimodal interfaces; 01. Adaptable and adaptive multimodal interfaces; 02. Anthropomorphic interfaces and life-like characters; 06. Fusion and fission techniques*

## General Terms
Algorithms, Design, Human Factors

## Keywords
Intelligent multimodal interfaces, multiple applications, system description.

## 1. INTRODUCTION
Intelligent multimodal interfaces empower users to navigate in information spaces or to control their environment [10,17]. As shown by Oviatt and Cohen [14] the combination of multiple modalities on the input and output side of an interactive system reduces errors in the communication.

With SmartKom (www.smartkom.org) we look into the combination of multiple modalities for input and output to reach these goals [21]. SmartKom features the situated understanding of possibly imprecise, ambiguous, or incomplete multimodal input and the generation of coordinated, cohesive, and coherent multimodal presentations. While most mono- or multimodal dialogue systems provide access to one functionality like map-based services [8] or a related group of functionalities like the command posts of the future [12], our goal in SmartKom was to come up with a processing approach that covers a wide variety of applications, and modalities under the umbrella of a uniform backbone system.

In this paper we do not cover the application specific functionalities, but present the basic ideas of the system's design, the interaction metaphor, and the design principles. Then we present a tour through the multimodal processing in the dialogue backbone, which is responsible for the processing of the ongoing interaction with the user.

## 2. ONE INTERACTION METHAPHOR - THREE SCENARIOS
The research goal of SmartKom was to come up with a uniform multimodal dialogue interface to many applications, ranging from consumer electronics control to mobile services. Our fundamental hypothesis - which we call the situated delegation oriented dialogue paradigm - is that the user gets a homogenous and pleasing interaction experience through an anthropomorphic personalized interaction agent, called Smartakus, to whom the user delegates the task to be solved (see figure 1). Both communication partners collaborate during the problem solving process in which the personalized agent accesses the background services. The agent may ask for more information and finally presents results on the output channels.

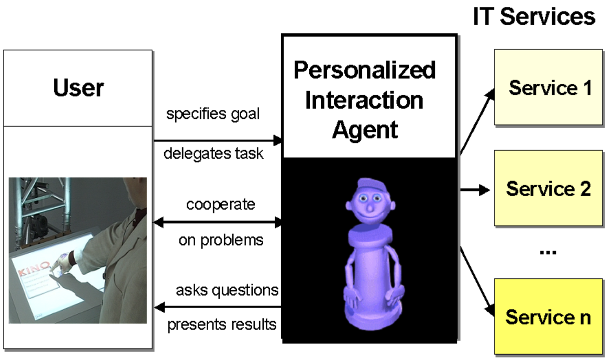


**Figure 1. The situated delegation oriented dialogue paradigm**

The modalities available to the user are mainly speech, processed by a user-independent speech recognizer, and natural gestures that are recognized by scenario-specific hardware (see below). The system presents visualizations of domain information like maps or cell phones which are addressed during the dialogue. The agent whose appearance is adapted to the screen size moves between these objects and uses gestures to focus on interesting information. For speech output we use a unit-selection based speech-synthesis developed specifically for this project [19].

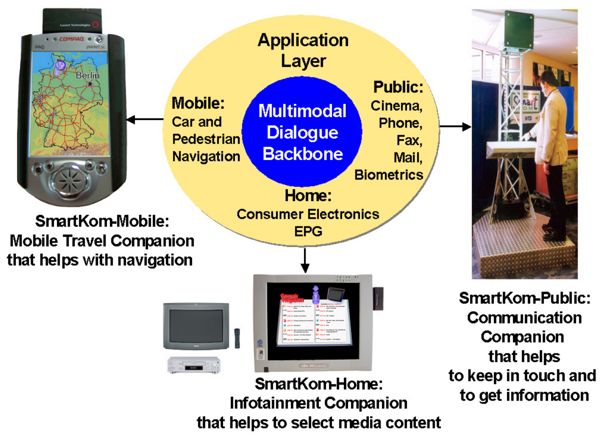


**Figure 2. The three scenarios of SmartKom**

Figure 2 shows the three scenarios of SmartKom which realize the interaction metaphor. The public information booth on the right side realizes a touch free interaction using gesture recognition with an infrared camera. The visual context is projected on the table in front of the user. The user can act like in real life and use natural gestures to perform possibly imprecise referential actions. Applications are cinema information, including interactions with a city map of Heidelberg, different types of biometry, and communication services like phone, fax, and e-mail.

In the scenario SmartKom-Home a portable touchpad captures the gesture input. The user can control her home theater using SmartKom and has access to electronic programming guide (EPG) information. Especially in this scenario it is important to respect the modalities available to address the user: In the so-called *lean-forward* mode, the user pays attention to the visual content, while in the *lean-backward* mode the user, e.g., watches TV and can be addressed by speech only.

The mobile system uses a PDA or the navigation system screen of a car as visualization hardware for car or pedestrian route interactions. Maps and accompanying information for points of interest are presented on the screens. Note that the agent is realized only as a talking head to make economically use of the limited screen estate. Also, only one application visualization is available at one time thus respecting the limitations of the output device. The targeted city of our system is Heidelberg with a detailed map of the city and a wide variety of tourist-related information.

Table 1 shows the 52 functionalities the user can currently address in the three scenarios with their different applications. All functionalities, modality combinations and technical realizations – including a wide variety of hardware options for the periphery – are addressed by the same core dialogue system with common shared knowledge sources. This is possible because we strictly follow two principles:

The first principle is that we do not implement special solutions for special problems. There are no shortcuts or application specific procedural dialogue processing steps in the dialogue core system. All functionalities are treated coherently using a knowledge based approach.

The second, related principle is to adhere to the rule "No processing and presentation without representation!". For all multimodal inputs and outputs we use a common representation approach that allows us to use generic interaction models. The interaction processing is based on the uniform **M**ulti**M**odal **M**arkup **L**anguage (M3L). It is formulated as a set of XML schemata [7]. The main schema, describing the intentions of both the user and the system, is defined off-line in an ontology, using the OIL [13] representation and notation framework. The OIL-based knowledge source is automatically transformed into an XML schema [6]. Currently, our ontology comprises more than 700 concepts and about 200 relations, which describe the abstract objects needed to communicate about the whole set of functionalities. We apply a closed world reasoning – everything the user and the system can talk about is encoded in the ontology.

The technical realization is based on the MULTIPLATFORM testbed [7], an integration platform with a distributed component architecture. The platform is implemented on the basis of the scalable and efficient publish/subscribe approach that decouples data producers and consumers. Software modules communicate via so-called data pools that correspond to named message queues. It is an extension of the integration middleware that was developed in the project Verbmobil [20] and is successfully used also in other projects. The underlying architecture allows for the easy integration of system modules and provides a wealth of debugging and test tools. The XML based interfaces assure a clear and well-defined syntax for data exchange that can be validated easily. Figure 3 shows the control screen of the system with all major modules of SmartKom.

**Table 1. Multimodal addressable functions in SmartKom**

| | | | |
|---|---|---|---|
| **Home** | | | |
| EPG (Electronic-Programming Guide) | General program<br>Channel selection<br>Channel information<br>Selection based on genre | Information for one broadcast<br>Time-based operations<br>Help functions for genres | 7 |
| TV | On/off | Channel selection | 2 |
| VCR control | On/off<br>Record<br>Play<br>Pause | Wind/rewind<br>Programming using EPG and the calendar | 6 |
| Lean-Forward/<br>Lean Backward | Select Lean-Backward<br>Deactivate Lean-Backward | Context aware presentations | 3 |
| **Total Home** | | | **18** |
| **Public** | | | |
| Telephone | Manipulative key operations<br>Telephony functions | Audio handling<br>Address book | 4 |
| Hand contour biometry | Selection of biometry type<br>Hand biometry | Presentation and camera control<br>Address book (see above) | 3 |
| Voice biometry | Presentation and audio control<br>Voice biometry | Address book (see above)<br>Selection of biometry type (see above) | 2 |
| Signature biometry | Presentation and tablet control<br>Signature biometry | Address book (see above)<br>Selection of biometry type (see above) | 2 |
| Fax | Presentation and interaction<br>Fax handling | Address book (see above)<br>Camera control | 3 |
| E-Mail | Presentation and interaction<br>E-Mail handling | Address book (see above)<br>Camera control (see above) | 2 |
| Cinema | General program<br>Movie information | Seat reservation<br>Cinema location | 4 |
| **Total Public** | | | **20** |
| **Mobile** | | | |
| Car navigation | Selection of start und goal city<br>Route type selection<br>Car route computation | Selection of parking garage<br>Information about parking garages | 5 |
| Pedestrian navigation | Selection of map type<br>Selection of start und goal<br>Route computation | Selection of points of interest<br>Information for points of interest<br>Integrated car and pedestrian route planning | 6 |
| Map manipulation | Resize<br>Help functions for map interactions | Change viewpoint | 3 |
| **Total Mobile** | | | **14** |
| **Total System** | | | **52** |

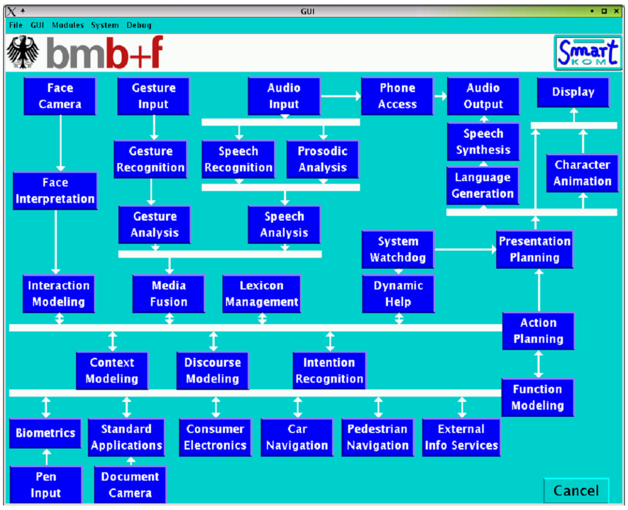


**Figure 3. The control screen with the main modules**

# 3. THE INTEGRATED MODALITY PROCESSING

## 3.1 The modality analysis and fusion

Our tour through the system starts with the processing of the input. Assume that the user requested the TV program and it is displayed on the screen (see figure 9 below for an example). Since we require that everything that is visualized is also encoded in our domain representation, the system has a description in terms of the intentional representation of the content. If the user now utters a sentence like "*Who is acting in this movie?*" [1] accompanied by a pointing gesture, the modality analyzers, the discourse memory and the modality fusion work together to come up with the intention of the user's input.

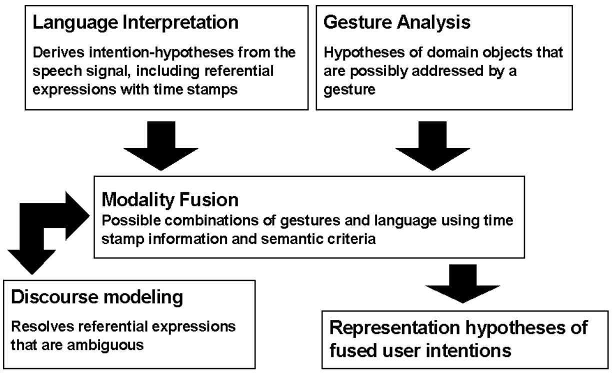


**Figure 4. The modality analysis and fusion**

Figure 4 shows the information flow between these modules. Since the user-independent speech recognition still has a significant error rate, we use a robust template based analysis approach that parses the word lattice as produced by the speech recognizer [5]. The output of the language interpretation is a list of hypotheses of user intentions that comprise of ontological representations of the input, time stamps, and scoring information. The same is true for the output of the gesture analysis. This module takes the coordinates and gesture type information of the gesture recognizers, like pointing, encircling or tarrying, and computes the most probable hypotheses of referenced objects addressed by a gesture, using the representation of the objects visible on the screen. The fusion then combines both lists and with the help of the discourse memory comes up with the list of merged hypotheses. During the merging process many unimodal hypotheses are discarded if the combination is impossible due to the underlying knowledge source. Important information at this module's disposal besides timing information are referential expressions in the language hypotheses.

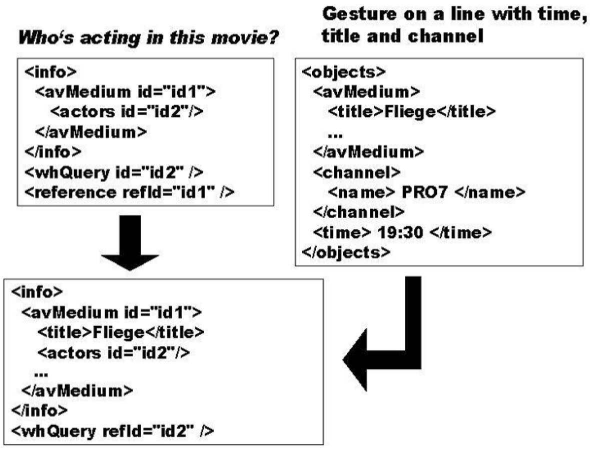


**Figure 5. An example for the multimodal fusion**

Figure 5 shows an example for the fusion process. For the example utterance the robust parser created an empty intention hypothesis with a request for an avMedium object that has no further information. Gesture analysis provides, amongst others, a specific object of this type and fusion can merge the information from both modalities.

## 3.2 The discourse memory

The central repository for representing discourse information is the discourse memory [15]. We have merged and extended the approaches of LuperFoy [9] and Salmon-Alt [18] to represent discourse beyond standard approaches in computational linguistics. The basic structure is a three-tiered representation model where for each object in any input and output modality we store a so-called modality object for that particular event. Such an object refers to a discourse object located at the discourse level. Input and output from multiple modalities can therefore contribute to the same object at the discourse level. The discourse objects are linked to respective descriptions in the ontology.

Figure 6 shows a snapshot of the memory, where the system presented a list of information on the screen. Our approach enables cross-modal referential expressions. Thus, the user can address, e.g., the elements of the presentation using, for

[1] The example is translated from German to English, as are all the other example inputs and outputs throughout this paper.

example, a definite numerical expression, such as "*the first*", for referring to the an object in the sequence of discourse objects.

Besides the support of the reference resolution processes the main task of the discourse module is to merge incoming new information to the already existing background knowledge. This is done with a default unification operation called Overlay [2]. This includes enrichment, type-coercion of hypotheses as well as their validation. The processing respects expectations from the action planner (see below) about the possible discourse continuations. To guide the search, our discourse memory also encompasses global and local focus structures.

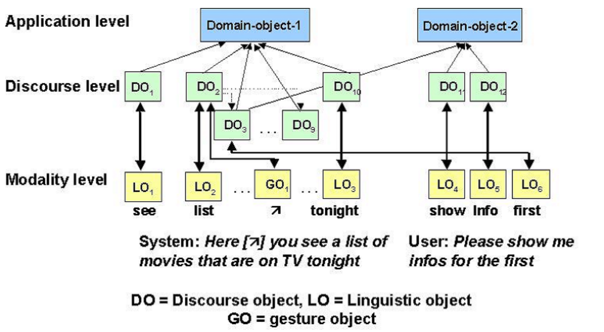


**Figure 6. An example for the multimodal discourse memory**

## 3.3 The action planner

After intention processing which rescores hypotheses, the intention list is passed on to the action planner.

The action planner has the task to coordinate the actions the dialogue system has to perform. This firstly means to identify the action the user wants the system to do on the basis of the incoming intentions. Then the planner has to interact with the various applications: Select the appropriate application for the user's request, address it correctly and finally forward the content that has to be realized by the presentation module. Separate modules for function modeling and information coercion, which hide application specific details to a great extent from the action planner mediate the communication with the applications. Therefore, the planner's interface is (almost) exclusively based on the modeling of the intentions as defined in the ontology. Also, the output to the presentation module is expressed in terms of this representation.

This results in a uniform approach in planning, namely performing intention based communicative games [4]. The input and output structures in all channels, be it from and to the user or from and to applications or internal functionalities like lexicon updates are based on a uniform representation: The planner uses dialogue acts like *request*, *response* and *inform* to describe these games. At the heart of the module is a backward chaining, non-linear regression planning approach.

In figure 7 we show the dialogue game for the TV program overview. The top-level goal is created when the user asks for the TV program. The response is the resulting information that is presented to the user. However, to get the information, the system has first to ask for additional information about a time constraint, which creates an expectation. This expectation information is forwarded to the discourse memory module. If the user utters an elliptical expression, the system can match this response with the open expectation.

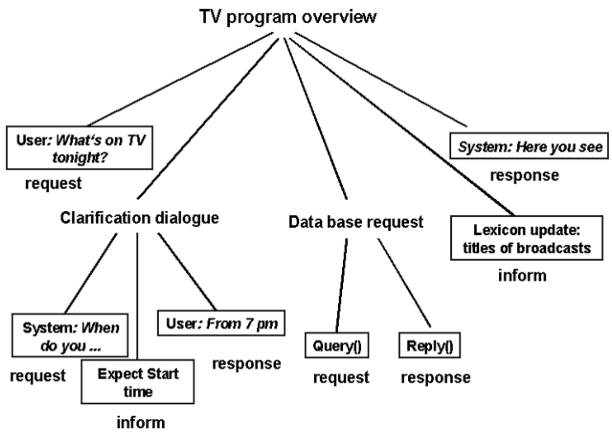


**Figure 7. An example for a communicative game**

The next step is a dialogue game with the electronic programming guide database that provides the information about tonight's TV program. The communicative game continues with the lexicon module of SmartKom. The result of the database query contains words that are not in the lexicon of the speech recognizer or the speech synthesis. Those modules have to be informed about possible words the user or the system might utter when presenting or addressing the titles, actors and other information. The lexicon module creates the phonetic transcriptions – since titles are often in foreign languages it has to provide pronunciation variants – and updates the speech modules. New words that are added to the recognizer and the synthesizer belong to a list of fixed word classes – mostly names – that have corresponding descriptions in the object layer of the ontology. Therefore, we can process them under the closed world assumption employed in SmartKom. Finally the action planner initiates the response of the system.

The games are stored as plan operators with pre- and postconditions and actions. They can be executed synchronously or asynchronously. Especially important is that the user can easily deviate from the games and can start new sub-dialogues, with the current game still pending.

## 3.4 The dynamic help and watchdog

The integrated modality processing is supplemented by a component that handles problematic events, e.g. internal system errors or malfunctions of modules, problems in the progress of the dialog, changes of emotional states of the user, underspecified requests that cannot be resolved by discourse analysis, or indirect and vague help requests and information queries on the meta level. Most of these events have a negative property in common, namely that there is no obvious specification of an application task that the user wants to perform. For analyzing these events a thorough awareness of the system's capabilities, processing state and dialog situation is required.

In any case the processing gets stuck – which may be caused by any software or hardware problem in the system – the system's watchdog is able to detect this condition and to react with an appropriate feedback. The system monitoring is also used to provide the presentation module with information about the current state of the internal processing. The interface agent signalizes this state by e.g. turning away from the user and typing on a laptop, thus signalling that he's busy and that the user should be patiently waiting for the results.

The dynamic help module performs a deep analysis of the situation and the context. It has a layered architecture with a planning component on top. The top-level predominantly guides the behaviour of the module in an abstract way. It basically determines how lower levels must compute specifications from knowledge sources, also from imported ones of other modules, and from the situation.

We demonstrate the type of reasoning that is performed with the example of an emotional statement expressed by e.g. "*Yes I like this"*. We defined a knowledge source of potential likes and dislikes for the applications. They refer to rather abstract states of affairs that have to be substantiated by context. To make the search efficient, the eliciting conditions are organized as a decision network. Utterances are analysed by reading eliciting conditions backwards in an abductive manner. In a second level, specifications of reactions are computed that may overcome a negative attitude or that draw consequences from a positive attitude for the further behaviour of the system. Important methods are query relaxation resulting in a modified database query or publishing user preferences, that may be used as presentation parameters. For the example above one consequence is to forward a message to the presentation planner to sort information in a cinema or EPG presentation according to the user's preferences.

## 3.5 The multimodal presentation

### *3.5.1 The presentation pipeline*

In SmartKom, the action planner and the dynamic help module provide the input for the presentation planner. The decision how this presentation goals are to be realized depends on several layers of conditions, starting from the selection of an appropriate distribution across the output modalities of the system to restrictions about available modalities.

Figure 8 shows the data flow of the presentation modules. First, realized as a set of XSLT style sheets, the presentation description is transformed into the internal language of the presentation planner [11]. This step also inserts important information about the dialog situation. Some examples for the presentation parameters are the current scenario, the display size whose range is from 1024x768 pixels for the public and home scenarios to 240x320 pixels for the mobile scenario, available user interfaces, and user preferences, e.g. the preferred modality. Information about preferred and available modalities is provided by the interaction modeler module, which analyzes scenario and user specific modality preferences. For example, the user can tell the system in the home scenario that she is no longer paying attention to the information on the screen. The information the system wants to convey must then be expressed verbally.

The planner applies predefined presentation strategies that decompose the complex presentation goal into presentation tasks. For example, the media fission strategy decides whether a description is to be uttered verbally or graphically. It is based on constraints in the strategies and the data in the context. The result is a hierarchical plan that is transformed into a presentation script for the display manager. A special sub-task of this planning process is the generation of animation scripts for the presentation agent that have to be related and coordinated with the graphical presentations. The main task is the selection of gestures that supplement the graphical output appropriately.

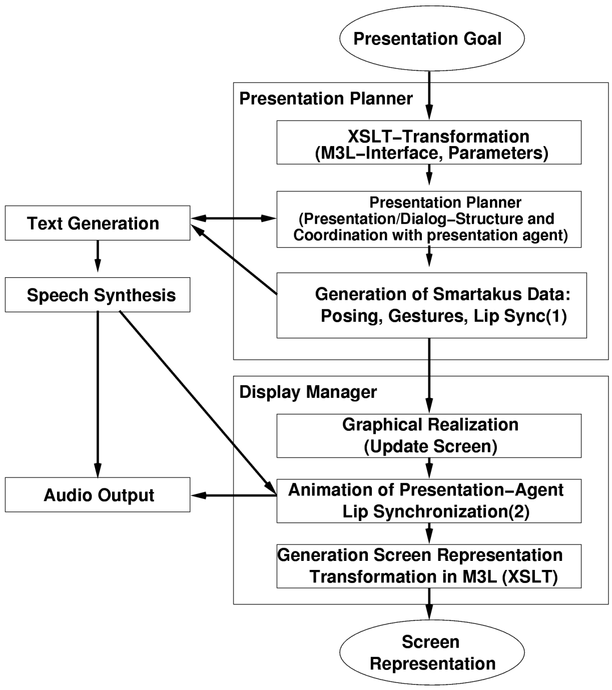


**Figure 8. The presentation pipeline**

The display manager determines the layout for the objects using different predefined graphical layout elements, e.g., for a TV program, for hand contour biometry, for maps, or for a cell phone. They are filled dynamically by content elements from the action planner's input, e.g., movie titles, channel logos, or maps. Since many objects can be shown at the same time on the display, the layout manager re-arranges the objects on the screen and removes objects, if necessary.

The second task of the display manager is the integration of the animated agent into the presentation and the coordination of its animations with the graphical and the speech output. It defines a fixed time point at which the audio signal has to start to synchronize the lip animation.

In the media fission step of the planning process the presentation activates the natural language generation component. It follows a template-oriented approach, based on Tree Adjoining Grammars [1]. It starts with high-level communicative goals and object descriptions based on the ontology of SmartKom. The module either applies templates for simple, fixed sentences, which convey fixed intentions, or uses more elaborate reasoning for complex intentions and descriptions to be generated, combining templates and grammar rules to generate natural language. The output consists of text annotated with conceptual structures for a concept-to-speech synthesizer. The synthesizer not only generates the audio signal, but also a phoneme transcription that is passed back to the animation of the agent, to allow for a lip-synchronous visualization while talking.

The generation of synchronous lip animations is based on a mapping of phonemes to a predefined set of only 8 static unmoved mouth position pictures (visemes). The synchronized audio-visual output itself uses a special viseme displaying routine that continuously matches the elapsing time against the time it takes to display a viseme, to speak the phoneme and the time to perform the match itself [16]. In the first evaluation this feature was noted by naïve subjects and significantly added to the acceptance of the presentation agent.

### *3.5.2 User-adapted modality allocation*

The importance of user-adapted modality allocation is obvious especially in the home scenario. We defined two modi, depending on the user's attention. In the *lean-forward* mode the user interacts with the touchpad, paying attention to the presentation, listening to the spoken output and reacting using pen gestures and speech. Figure 9 shows the result of a query like "*Show me tonight's TV program!"*. Since the screen is the best modality to present large quantities of data, all details of the program are presented in tabular form. Speech output only repeats the details of the query itself, thus giving the user concise feedback about the system's understanding of the input.


**Figure 9. Lean-forward presentation**

In the *lean-backward* mode the user turns away from the screen, e.g. when she watches TV and she tells the system not to interrupt. Since the visual modality is not available, and the TV data is presentable verbally – unlike, e.g., the details of a map – the user's attention is not called to the screen, but SmartKom generates exclusively spoken output. Since there are too many shows in total, only the most relevant ones are mentioned verbally and a hint about further on information is given: "*For example,....*" (see figure 10). On the display, the presentation agent remains visible to show that the system is still active. If the user wants to get more information, she can ask the system for the next titles in the list.

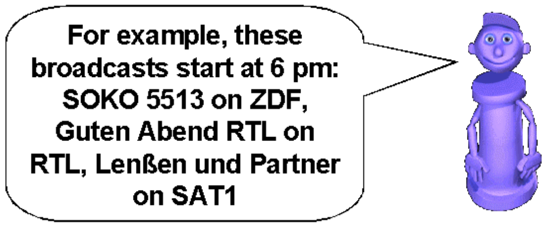


**Figure 10. Lean-backward presentation**

## 4. CONCLUSIONS

In this paper we presented a short overview of the dialogue backbone at the heart of the multimodal dialogue system SmartKom. It enables the dialogic interaction with multiple applications using a coherent interface metaphor. Once the basic approach of using a common, OIL-based representation was in place, and using the solid foundation of the middleware and its communication approach, adding new functionalities was very straightforward. With the modular, knowledge-based approach is was also fairly easy to replace the German speech recognizer and synthesis with English versions, and to rewrite the knowledge sources for the parser and generator to English for the mobile scenario.

During the development of the system, all interactions with the system reached a high, three-digit number. Evaluations using the PROMISE method were performed on two intermediate systems [3]. While the first system lacked stability and robustness, a clear improvement was reached for the next milestone system. Users especially liked the interaction with the interface agent and the coherent interface metaphor. The German and the English systems – predecessor versions and the final one – were presented at various conferences and at other public opportunities over the last three years.

## 5. ACKNOWLEDGEMENTS

The system was developed in the context of the "Human Computer Interaction" research program funded by the German Federal Ministry of Education and Research from 1999 to 2003 under grant 01 IL 905. The responsibility for the content is with the authors. We would like to thank our project partners at DaimlerChrysler AG, European Media Laboratory GmbH, Friedrich-Alexander University Erlangen-Nürnberg, Internatio-nal Computer Science Institute, Ludwig-Maximilians University

München, MediaInterface GmbH, Philips GmbH, Siemens AG, Sony International (Europe) GmbH, and University of Stuttgart.

## 6. REFERENCES


[1] Abeillé, A. and Rambow, O. Tree Adjoining Grammars: Formalisms, Linguistic Analysis and Processing. CSLI, Stanford, USA, 2000.

[2] Alexandersson, J. and Becker, T. The formal foundations underlaying Overlay. In: Proceedings of the Fifth International Workshop on Computational Semantics (IWCS-5), Tilburg, The Netherlands, 2003

[3] Beringer, N., Kartal, U., Louka, K., Schiel, F., and Türk, U. PROMISE: A Procedure for Multimodal Interactive System Evaluation. In: Proceedings of the 3rd Language Resources & Evaluation Conference (LREC), Las Palmas, Gran Canaria, Spain, 2002.

[4] Carlson, L. Dialogue Games. Reidel Publishing Company, 1983.

[5] Engel, R. SPIN: Language Understanding for Spoken Dialogue Systems Using a Production System Approach. In: Proceedings of the 7th International Conference on Spoken Language Processing. Denver, Colorado, USA, 2002.

[6] Gurevych, I., Merten, S., and Porzel, R. Automatic creation of interface specifications from ontologies. In: Proceedings of the HLT-NAACL Workshop on Software Engineering and Architecture of Language Technology Systems (SEALTS), Edmonton, Canada, 2003.

[7] Herzog, B., Kirchmann, H., Merten, S., Ndiaye, S., Poller, P., and Becker, T. MULTIPLATFORM Testbed: An Integration Platform for Multimodal Dialog Systems. In: Proceedings of the HLT-NAACL Workshop on Software Engineering and Architecture of Language Technology Systems (SEALTS), Edmonton, Canada, 2003.

[8] Johnston, M., Ehlen, P., Bangalore, S., Walker, M., Stent, A., Maloor, P., and Whittaker, S.. MATCH: An Architecture for Multimodal Dialogue Systems. In: Proceeding of 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, USA, 2002.

[9] LuperFoy, S. The Representation of Multimodal User Interface Dialogues Using Discourse Pegs. In: Proceeding of 30th Annual Meeting of the Association for Computational Linguistics, Newark, USA, 1992.

[10] Maybury, M. and Wahlster, W. (eds.). Readings in Intelligent User Interfaces. Morgan Kaufmann, 1998.

[11] Müller, J., Poller, P., and Tschernomas, V. Situated Delegation-Oriented Multimodal Presentation in SmartKom. In: Workshop Notes of the AAAI Workshop on Intelligent Situation-Aware Media and Presentations (ISAMP), Edmonton, Canada, 2002.

[12] Myers, B.A., Malkin, R., Bett, M., Waibel, A., Bostwick, B., Miller, R.C., Yang, J., Denecke, M., Seemann, E., Zhu, J., Peck, C.H., Kong, D., Nichols, J., and Scherlis, B. Flexi-modal and Multi-Machine User Interfaces. In: [17].

[13] OIL. Ontology Inference Layer. http://www.ontoknowledge.org/index.shtml (Last access: 10. August 2003).

[14] Oviatt, S.L. and Cohen, P.R. Multimodal Interfaces That Process What Comes Naturally. Communications of the ACM, Vol. 43, No. 3, pp. 45-53, 2000.

[15] Pfleger, P., Alexandersson, J., and Becker, T. A robust and generic discourse model for multimodal dialogue. In: Proceedings of the 3rd IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems, Acapulco. Mexico, 2003.

[16] Poller, P. and Müller, J. Distributed Audio-Visual Speech Synchronization In: Proceedings of the 7th International Conference on Spoken Language Processing. Denver, Colorado, USA, 2002.

[17] Proceedings of the 4th International Conference on Multimodal Interfaces. Pittsburgh, USA, 2002.

[18] Salmon-Alt, S. Référence et dialogue finalisé: de la linguistique à un modèle opérationnel. PhD Thesis, Université H.Poincaré, Nancy, 2001.

[19] Schweitzer, A., Braunschweiler, N., Klankert, T., Möbius, B., and Säuberlich, B. Restricted unlimited domain synthesis. In: Proceedings of Eurospeech 2003 Geneva, Switzerland.

[20] Wahlster, W. (ed.). Verbmobil: Foundations of Specch-to-Speech Translation. Springer, 2000.

[21] Wahlster, W., Reithinger, N., and Blocher, A. SmartKom: Multimodal Communication with a Life-Like Character In: Proceedings of Eurospeech 2001, Aalborg, Denmark.