The SmartKom Architecture: A Framework for Multimodal Dialogue Systems

Gerd Herzog and Norbert Reithinger

DFKI GmbH, Stuhlsatzenhausweg 3, D-66123 Saarbrücken
{herzog,reithinger}@dfki.de

Summary. SmartKom provides an adaptive and reusable dialogue shell for multimodal interaction, which has been employed successfully to realize fully-fledged prototype systems for various application scenarios. Taking the perspective of system architects, we will give a review of the overall design and specific architecture framework being applied within SmartKom. The basic design principles underlying our approach will be described and the different instantiations of the conceptual architecture will be presented to illustrate the adaptiblity and flexibility of the generic framework.

1 Introduction

The next generation of user interfaces aims at intelligent systems which are able to adapt to common forms of human dialogues and hence provide more intuitive and natural ways of interaction (see [2, 3, 13, 15, 16, 21]). This ambitious goal, however, poses new challenges for the design and implemention of interactive software systems. Compared with traditional user interface software these kind of intelligent interfaces, which are capable to employ natural language as well as other modalities for communication, require more sophisticated architecture models.

The development of a complex multimodal system like SmartKom, which realises three different application scenarios, requires an architecture that accomodates various, sometimes conflicting goals:

- Natural interaction behaviour: Facilitate timely, user adapted interaction.
- Flexibility: Handle a large number of modalities and applications.
- Openness: Allow for different processing approaches.
- Manageability: Support distributed development.

In this contribution, we will elucidate the architectural design of SmartKom, which provides a practical framework for the succesful realization of effective multimodal dialogue applications. The following section discusses essential requirements for multimodal dialogue systems and presents a more elaborate high-level architecture for intelligent interfaces. Based on these design foundations, a detailed description of the specific architecture framework that has been developed within the SmartKom project will be provided in Sect. 3. Following the introduction of the underlying framework the different instantiations of the generic SmartKom architecture are presented in Sect. 4 to highlight the flexibility and adaptibility of our approach.

2 Architectural Requirements for a Multimodal Interaction System

Conceptual architectures like the ones described in [1, 4, 7, 8, 11, 14] contrast the system design of traditional handcrafted interfaces with intelligent user interfaces. Within the classical Seeheim reference model [17] the internal structure of the user interface component that mediates between the end user and the underlying application system is partitioned into three distinct conceptual tiers:

- The *presentation* level or display management determines the appeareance and low-level behaviour of the user interface.
- *Dialogue control* constitutes the intermediary processing stage which manages the flow and order of interaction.
- The *application interface* as the third logical component provides the semantic model of the application which is needed to access the functional core of the system.

This coarse modularisation, however, is not sufficiently detailed to capture the basic structure of multimodal interfaces and to serve as a starting point for the development of the modular structure of a complex multimodal system.

Figure 1 provides a sketch of a more refined architecture at a high level of granularity. The conceptual architecture is based on the functional blocks of a generic multimodal dialogue system and decomposes the processing task into the following parts

- Sensor specific input processing: This layer of functionalities comprises all technical input devices like microphones and gesture recognition technologies. It provides a standardised, hardware independent signal- or symbol-level interface for further processing.
- Modality-specific analysis: This processing level consist of the recognisers that transform sensor signals from the environment into symbolic information and analysers that provide a meaning description for these signals.
- Modality fusion: This stage merges the meanings recorded in different modalities to one coherent, unified meaning representation. A core aspect of modality fusion is the mutual disambiguation of multimodal input on semantic and pragmatic levels.

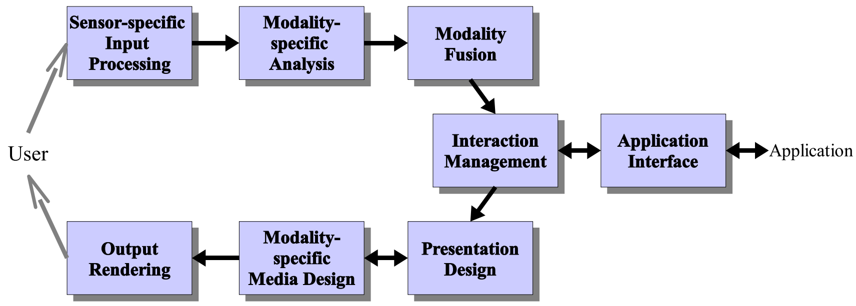

Fig. 1. A high-level architecture for multimodal interaction

- Interaction management: Here resides the "mind" of the machine. This part of the system identifies the intention of the user, determines the next steps to be taken by the system and addresses the various application functions.
- Application interface: This layer defines an abstract interface for the interaction with the background applications. It hides application-specific technical details and thus provides a generic way to control the applications.
- Presentation design: Given a system intention, the information to be conveyed needs to be transformed into coordinated multimodal output. This functional block constitutes the first stage of modality fission and deals with the overall organisation of the intended multimodal presentation. It includes in particular sub-tasks like content selection, media and modality allocation, layout design, and coordination.
- Modality-specific media design: In close cooperation with the presentation design stage, the processing on this layer is specifically concerned with the conversion of abstract content structures into presentable media objects.
- Output rendering: The final function block comprises the technical means to present the coordinated system reaction in the defined media channels like screens, loudspeakers, or force-feedback devices.

Of course, this break-down of the architecture can and does not define sharp lines between the functional blocks. For example, one could argue about the output interface of sensor-specific input processing. For speech, it could either be a standardised audio stream after echo compensation or the output of the speech recogniser, if that component already incorporates all lower level signal processing functionalities. In practice, the exact boundaries between the processing steps may depend on the constraints imposed by available components that are to be used for building a multimodal system.

In addition to the processing steps mentioned above, a rich variety of knowledge sources that are not shown in the diagram are also vital within the system and may require shared access from multiple components. Important knowledge sources for intelligent multimodal interaction include explicit models of the user, domain, context, task, and discourse as well as the different media and modalities. Consider for example, reference resolution processes, which need access to data from discourse context storage. This information has to be provided by the output components and is delievered on demand to all interested parties.

The architecture sketch highlights the major flow of information through the system, leading to a linear structure that resembles a processing pipeline. However, this depicts only the main and principal succession of processing steps including analysis, execution, and generation. It does not rule out additional interconnections. The individual components of the system could be regarded as communicating processes rather than just pipeline stages.

If the system designer now refines this high-level architecture, various additional aspects have to be considered [5], including

- Declarative data models: The benefits of a flexible and configurable knowledge-based approach can only be exploited if system design strictly follows the principle that there should be no processing and presentation without explicit representation.
- Time-coordinated parallelism: Modality-specific analysers and generators run in parallel. Therefore the analysis and generation of events must be coordinated and the architecture must support this behaviour. Especially important are time constraints on processing as well as the provision of timing information within data formats.
- Standardised interfaces supporting the handling of probabilistic information: Modality recognisers usually provide alternative interpretation hypotheses augmented with information about the recognition quality. The design of the integration platform has to support the maintenance and handling of multiple, competing hypotheses.
- Incremental processing: It cannot be assumed that only complete representations are passed between the function blocks in the system. Increments of information, either originating in the user's input style or caused by processing methods must be handled properly in the system.
- Iterative development: A complex multimodal system usually starts from a limited core of modalities and functionalities. Step-by-step, the coverage of the system is extended, with new input/output modalities being added. Therefore, the design must be extensible, not limiting additions by hard-wired interaction and control paths. This requires accessible interfaces and extensive support for debugging and monitoring.
- Distributed development: Depending on specific organisational and technical constraints, software modules from various sites and with different background have to be integrated. For instance, it is very typical that off-the-shelf recognisers are to be used together with bespoke analysis components.

In the next sections, we will show how the SmartKom system realises these requirements in three different application scenarios.

3 SmartKom as an Adaptive and Reusable Shell for Multimodal Dialogues

SmartKom represents a new generation of multimodal dialogue systems, dealing not only with basic modality integration and synchronization but addressing the full spectrum of dialogue phenomena arising in multimodal interaction [19, 22]. With a clear focus on reusability, SmartKom has been designed as a transmutable system that can engage in many different types of tasks in various usage scenarios [18, 23].

3.1 Distinguishing Features of the SmartKom Approach

The SmartKom system relies on the so-called *situated, delegation-oriented dialogue paradigm* [24] to provide an anthropomorphic and affective user interface. This interaction metaphor is based upon the idea, that the user delegates a task to the virtual communication assistant which is visualised as a life-like character. The interface agent recognises the user's intentions and goals, asks the user for feedback if collaboration is necessary, accesses the various application services on behalf of the user, and presents the collated results in an appropriate form. The interaction style being employed within SmartKom targets in particular non-desktop scenarios, such as smart rooms, kiosks, or mobile application contexts.

An interesting property of the SmartKom dialogue shell is its ability to integrate multiple services, i.e. different application functions which have been developed independently from each other, into a coherent, value-added system.

Another pecularity of the system is a build-in self-monitoring capability which continueously determines the overall processing state using status information from the individual components. In the sense of a reflective architecture [12], this kind of metalevel reasoning and control can be exploited to provide the user with appropriate feedback concerning system activity and potential error situations.

3.2 Anatomy of the SmartKom Core System

The SmartKom kernel shown in Fig. 2 provides the core building blocks for a multimodal interactive system. The diagram documents the software architecture using a component-and-connector viewtype as defined in [6]. Each component represents one of the principal processing units of the executing system and the highlighted components are being reused in all application cases. Since SmartKom has been implemented as a distributed system (see

[10] for details), every component corresponds to an independent process. The connectors illustrate the pathways of interaction. The underlying architectural style is based on communicating processes, more specifically on the publish-subscribe style. In this style, the interaction is event-driven using directed communication links between a message sender, who acts as data producer or event source, and a set of recipients, the data consumers.

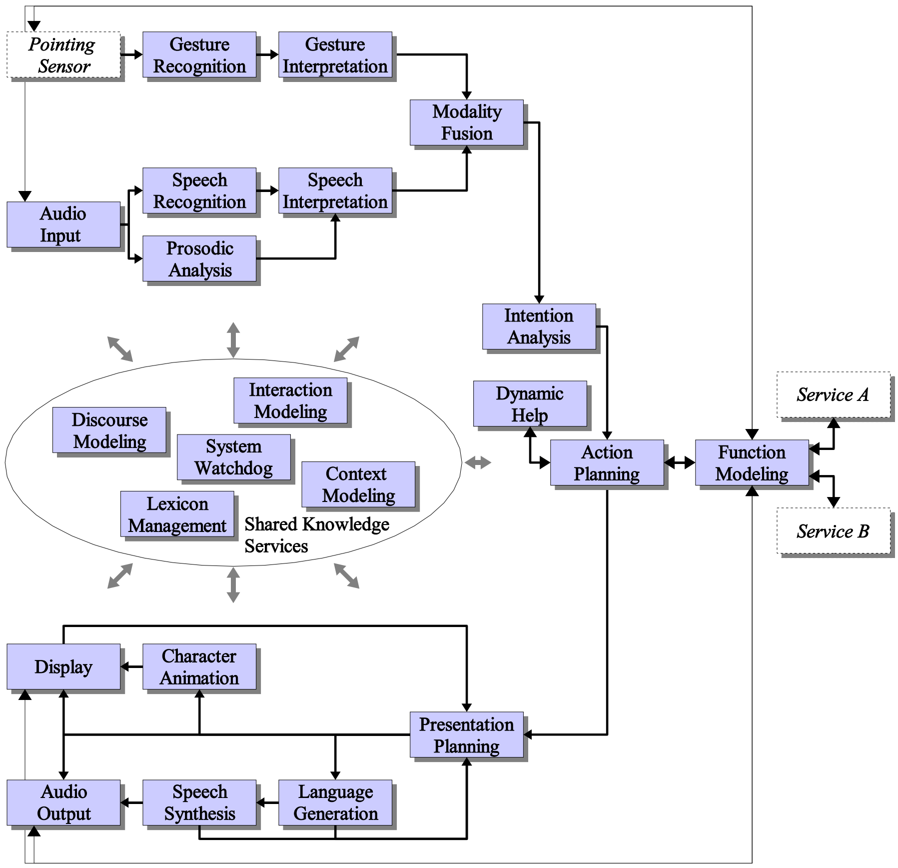

Fig. 2. SmartKom architecture blueprint

To provide a clearer outline of the system structure, the connectors chosen for Fig. 2 denote major information flows and may combine several publish-subscribe communication links. The various publish-subscribe connections between the shared knowledge sources and the other components of the dialogue backbone have only been sketched.

A basic SmartKom system supports multimodal input using speech plus gestures and integrates some application-specific services. The specific sensor type for gesture input depends on the technical setup of the given application scenario. Practical options that have been used for the SmartKom prototypes include either pen-based pointing or a vision system for hand tracking.

The separation of sensor-specific input processing into distinct components, which connect some technical device to the system and which encapsulate access to the underlying hardware, constitutes a first decomposition principle of the architecture framework. In order to provide maximum flexibility for system deployment, the same approach is being followed with respect to the output devices that are needed for the rendering stage.

Modality-specific input analysis can be modularized into separate components and it has proven useful to differentiate further between modality-specific recognition, i.e. processing of sensory data on the lexical and syntactical layer, and subsequent semantic interpretation of the derived symbolic representation.

Interaction management within SmartKom comprises several components. The main task of intention analysis is to select the most plausible input interpretation from the given set of hypotheses. Action planning constitutes the heart of dialogue control and is backed by a supplementary help component that is activated whenever difficulties occur during the interaction or if additional help is needed. The main processing stages are being supported by different components that actively maintain shared knowledge sources:

- The multimodal discourse model is utilized for the semantic and pragmatic interpretation during input and output processing. It is dynamically updated as system output progresses and performs contextual reasoning and scoring.
- Contextual information, as it is needed to handle references to situative parameters like current place and time, is provided by the context model.
- Interaction modeling is concerned with different aspects like available modalities and user preferences for specific forms of communication as well as the affective state of the user. The interaction model allows to dynamically adapt the communicative behaviour of the system.
- The lexicon is a dynamic knowledge source, which is updated with additional lexical entries depending on dynamic application data as it is received from the external information services. Lexicon updates are propagated to all components that process natural language input and output.
- The system watchdog monitors the processing status of all individual components to offer up-to-date information concerning the system state. This is used to provide immediate feedback or to initiate helpful reactions in case of processing problems.

The function modeling component realizes the application interface and in addition controls all input and output devices to coordinate access depending on an explicit state model. The application layer itself is modularised into multiple service components, each of which realizes some application-specific functionality.

The remaining elements of the component ensemble deal with modality fission. A second planning component is responsible for the presentation design. Part of the modality-specific output design is carried out by the character animation component which realizes the perceivable behaviour of the embodied conversational agent.

In addition to the integration of application-specific service components, the architecture framework provides other extension points to expand the core dialogue components with even more modalities. A practical example for an optional input modality is the interpretation of facial expressions to obtain information about the affective state of the user. The SmartKom prototypes focus mainly on audiovisual sensor input and presentations. With respect to other human senses, it would be feasible to include haptic sensations as well, e.g. using a force-feedback device. A limited form of tactile interaction is already provided when using pen-based input.

3.3 Representing Data with Multimodal Markup Language

The interaction between the distributed components within the SmartKom architecture is based on the exchange of structured data through messages. The external data format for the representation of structured information employs XML notation. XML, the *extensible markup language*, provides a flexible standard for the definition of specific data exchange formats that ensures interoperability between heterogeneous software components.

In the context of the SmartKom project we have developed M3L (Multimodal Markup Language) to cover all data interfaces within the entire multimodal dialogue system in a declarative way. Instead of using several unrelated XML languages for the various information flows, we aimed at an integrated and coherent language specification, which includes all sub-structures that may occur on the different communication links.

The basic information flow from user input to system output continuously adds further processing results so that the representational structure will be refined step-by-step. A typical example of such a complex M3L expression is shown in Fig. 3. The partial listing contains a so-called intention lattice that represents the interpretation result for a multimodal user input which combines a verbal information request ("Tell me more about this film.") with a pointing gesture.

Conceptual taxonomies provide the foundation for the representation of domain knowledge as it is required within a dialogue system to enable a natural conversation in the given application scenario. In order to exchange instantiated knowledge structures between different system components they need to be encoded in M3L. The corresponding parts of the M3L definition are derived from an underlying ontology that captures the required terminological knowledge [9]. For example in Fig. 3, the representation of the event

```xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<intentionLattice>
 <latticeStart>2003-01-24T11:01:55.000</latticeStart>
 <latticeDuration>PT3.02S</latticeDuration>
 <hypothesisSequences>
  <hypothesisSequence>
   <score>
    <source>acoustic</source><value>0.82</value>
   </score>
   <score>
    <source>gesture</source><value>0.73</value>
   </score>
   <score>
    <source>understanding</source><value>0.89</value>
   </score>
   <hypothesis>
    [...]
    <discourseStatus>
     <discourseAction> set </discourseAction>
     <discourseTopic>
      <goal> epg-info </goal>
    [...]
    <event>
     <informationSearch id="dim769">
      <pieceOfInformation>
       <broadcast id="dim763">
        <avMedium>
         <avMedium id="dim767">
          <avType> featureFilm </avType>
          <title> Cassablanca </title>
 [...]
 </hypothesisSequences>
</intentionLattice>
```

Time stamp
Confidence in speech recognition result
Confidence in gesture recognition result
Confidence in speech understanding result
Planning act
Object reference

Fig. 3. Excerpt from an M3L structure

structure inside the intention lattice originates from the ontology. M3L can easily be updated as the ontology evolves while novel application domains are being modeled.

4 Instantiations of the Generic Architecture

The SmartKom architecture framework is designed to support a wide range of collaborative and multimodal dialogues, that allow users to intuitively and efficiently access the functionalities needed for their task. The SmartKom prototypes address three different application scenarios which exhibit varying characteristics that influence the architectural design.

4.1 Public Information and Communication Kiosk

SmartKom Public realizes an advanced multimodal information and communication kiosk for train stations, airports, or other public places. It supports users seeking for information concerning movie programmes, offers reservation facilities, and provides personalized communication services using telephone, fax, or electronic mail.

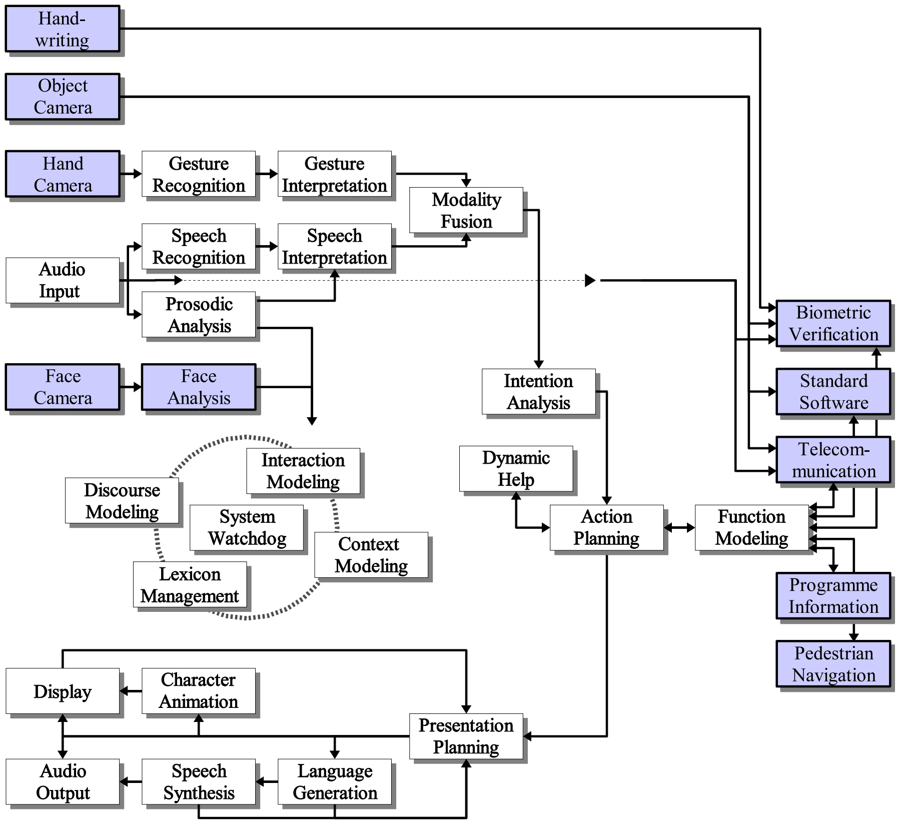

Fig. 4. Component architecture of SmartKom Public

Figure 4 provides an overview concerning the specific software architecture of SmartKom Public. Scenario-specific components are highlighted using grey boxes.

Gesture input is based on an infrared camera system that tracks the user's hand as it moves across a video projection of the graphical output. This kind of vision-based sensor does not require direct physical contact with the horizontal projection surface and hence allows for more natural hand gestures.

SmartKom Public integrates facial expressions as an additional input modality. In combination with acoustic indicators derived from prosodic analysis, the interpretation of facial expressions is used to recognize the current affective state of the user.

The kiosk system is equipped with two application-specific input components for pen-based signature input and an additional camera system that takes digital images of objects placed on the projection panel. In the given application scenario, the following functions are made available through the main service components:

- Biometric authentication using either voice, signature, or hand contour;
- Access to standard applications like address book and email transfer;
- Telephone connections and fax transmission;
- Cinema programme and reservation service;
- Provision of maps with cinema locations and marked routes to the desired destination.

The function model of the application interface coordinates the interplay of services to carry out specific tasks. The need for flexible application-specific use of input and output components is the reason why the basic control of all shared devices has been allocated to the function modeling component. The corresponding control lines have been omitted from Fig. 4.

Depending on the current application context, audio input can be re-routed to the biometrics component or to the telecommunication service. Input from the object camera is used to send documents (via email or fax) and the input component can also record the hand contour for biometric verification.

4.2 Infotainment Assistant for the Living Room at Home

In its basic configuration, SmartKom Home aims to provide a multimodal portal to home entertainment services. Using a portable tablet PC, the user is able to utilize the system as an electronic programme guide or to easily control consumer electronics devices like a television set or a digital video casette recorder.

Figure 5 highlights the specific components of the SmartKom Home prototype. The system supports the basic modalities that are associated with the core multimodal system and the user can employ a pen for stylus-drawn gestures on the tablet PC's screen. In the context of the Home scenario, however, two different styles of interaction modes are supported. In so-called *lean-forward* mode, coordinated speech and gesture input can be used for multimodal interaction with the system. Alternatively, the user is able to put away the tablet PC and change to so-called *lean-backward* mode. This interaction style is constrained to verbal communication. To compensate for the unavailable graphical view in this case, the presentation design needs to dynamicly adapt its presentation strategies using more elaborate natural language output. The user can easily request the activation or deactivation of the display to switch between the two modes.

SmartKom Home shares two of its service components with the Public scenario but employs different application functions from these services. The programme database is primarily used to obtain detailed information concerning upcoming televison broadcasts and the calender management utility from the standard software component is utilized to schedule off-air recordings of

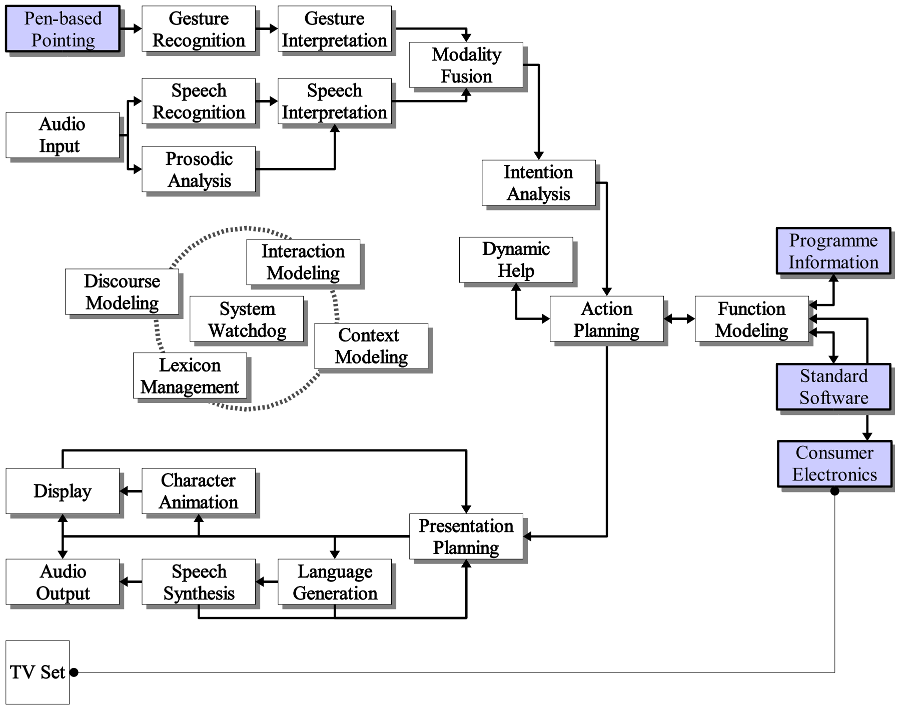

Fig. 5. Component architecture of SmartKom Home

selected programmes. The third service component provides a high-level interface to various appliances. In particular, the consumer electronics service provides remote control of a TV set, a VCR, and a satellite receiver. The TV screen can be regarded as an application-specific output device, which is not directly available for system output that is routed through the interface components.

SmartKom offers a very flexible and open approach for application integration. Services constitute connector components that provide a well-defined link to some application-specific functionality and often encapsulate complete and complex application-specific subsystems. A typical case is the integration of commercial off-the-shelf software. The standard software component in SmartKom, for example, constitutes a bridge to external standard applications provided by a separate installation of the IBM Lotus software product. The service component for consumer electronics reuses already available software modules from the related Embassi project [7]. It is basically a wrapper for a complex multi-agent system, which in fact relies on a different kind of middleware solution than the publish-subscribe style used to interconnect the SmartKom components.

4.3 Mobile Travel Companion

SmartKom Mobile provides a travel companion that can be used inside the car and while walking around. This application scenario comprises typical services like integrated trip planning and incremental route guidance through a city via GPS. Outside the car, the user operates the system using a handheld computer as a front end. Inside the instrumented car, the on-board equipment of the vehicle is used for displaying visual information and for audio input/output. A pecularity of the mobile travel companion is the seamless transition from one device to the other without loosing the current dialogue context.

Figure 6 shows the basic components of SmartKom Mobile. Multimodal interaction is possible using pen-based input on the handheld device. The in-car display, however, is not touch-sensitive.

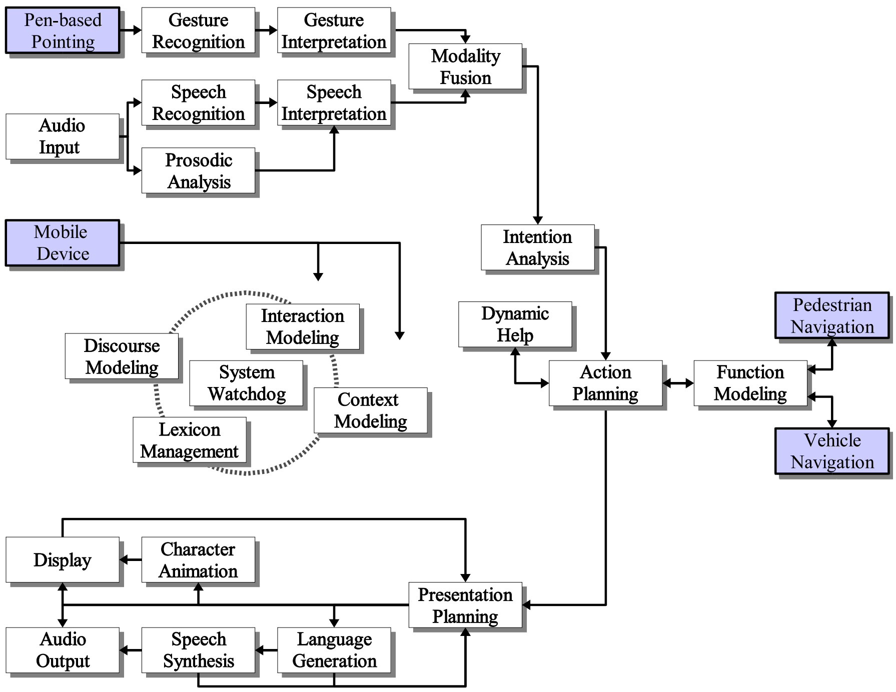

Fig. 6. Component architecture of SmartKom Mobile

The hand-over between alternative devices constitutes one of the central technical challenges imposed by this specific application scenario. The component named *mobile device* provides a generic solution to manage different remote front-end devices. In order to avoid time-consuming component restarts and shutdowns during system runtime, audio input and output as well as the display component have been enhanced with streaming capabilities for input and output redirection. RTP, the Internet-standard protocol for the transport of real-time data, is used for bi-directional audio connections and VNC [20] enables remote access to the display component. The mobile device component serves as a special kind of input device. After pressing the push-to-activate button on the handheld or in the car, the characteristics of the new front-end device—including also address parameters for video and audio redirection—are propagated by the mobile device component to dynamically update the interaction model. The significant difference in screen format of the handheld computer and the in-car display enforces the presentation design to automatically adapt the visual output accordingly.

The vehicle navigation component includes commercial off-the-shelf navigation software to generate map displays with marked vehicle routes. It also provides information concerning parking locations. The pedestrian navigation service offers feature-rich functions for map generation, touristic information, pre-trip planning, and incremental route guidance in Heidelberg. The component encapsulates a complex sub-system that is realized in a modular fashion, using a multi-agent framework.

SmartKom Mobile introduces an important new aspect for the dialogue management. In addition to user inputs, the system also needs to handle external events since navigation instructions are generated on the fly by the navigation service to provide incremental route guidance.

4.4 Other System Configurations

The SmartKom prototypes for the three sample scenarios use German for speech input and output. A special English system is based on a subset of SmartKom Mobile, excluding incremental route guidance. SmartKom English uses adapted versions of language-specific recognition, analysis, generation, and synthesis components.

In addition to the basic prototypes, many other plug-and-play variations of system setups with extended or restricted capabilities can be defined easily, using a flexible configuration mechanism for system instantiations. For example, for demonstration purposes, the SmartKom reference installation is often used as an extended Public system, which integrates all available service components.

5 Conclusion

Multimodal dialogue systems hold the promise of providing more intuitive and natural ways of human-computer interaction. In order to construct such advanced user system interfaces an effective architectural design is of utmost importance. SmartKom provides an adaptive and reusable dialogue shell, that has been employed successfully to realize fully-fledged prototype systems for various application scenarios.

SmartKom is based on an open architecture for multimodal dialogue systems that is flexible and adaptive. All available components can be reused directly and modular organisation into different service components simplifies the integration of application-specific functions. With respect to the dialogue backbone, the realisation of new application scenarios primarily becomes a knowledge engineering task instead of an extensive programming exercise. In order to extend the domain-specific dialogue behaviour, mainly the component-specific declarative knowledge sources have to be adapted.

The generic SmartKom architecture with its core components and their interconnections promises to provide a solid base structure even in cases where a customized multimodal dialogue system may require more advanced adaptations and extensions.

References

1. J. Allen, D. Byron, M. Dzikovska, G. Ferguson, L. Galescu, and A. Stent. An Architecture for a Generic Dialogue Shell. *Natural Language Engineering*, 6(3):1–16, 2000.
2. N. O. Bernsen and L. Dybkjær. Is Speech the Right Thing for your Application? In *Proc. of the Int. Conf. for Spoken Language Processing, ICSLP'98*, pages 3209–3212, Sydney, Australia, 1998.
3. M. M. Blattner and R. B. Dannenberg, editors. *Multimedia Interface Design*. ACM Press, New York, 1992.
4. M. Bordegoni, G. Faconti, S. Feiner, M. T. Maybury, T. Rist, S. Ruggieri, P. Trahanias, and M. Wilson. A Standard Reference Model for Intelligent Multimedia Presentation Systems. *Computer Standards & Interfaces*, 18(6–7):477–496, 1997.
5. H. Bunt, M. T. Maybury, and W. Wahlster. Coordination and Fusion in Multimodal Interaction. Dagstuhl Seminar Report 325, IBFI, Wadern, Germany, 2001.
6. P. Clements, F. Bachmann, L. Bass, D. Garlan, J. Ivers, R. Little, R. Nord, and J. Stafford. *Documenting Software Architectures: Views and Beyond*. Addison-Wesley, Boston, 2003.
7. C. Elting, S. Rapp, G. Möhler, and M. Strube. Architecture and Implementation of Multimodal Plug and Play. In *Proc. of the 5th Int. Conf. on Multimodal Interfaces*, pages 93–100, Vancouver, Canada, 2003. ACM Press.
8. S. Feiner. An Architecture for Knowledge-based Graphical Interfaces. In Sullivan and Tyler [21], pages 259–279.
9. I. Gurevych, S. Merten, and R. Porzel. Automatic Creation of Interface Specifications from Ontologies. In H. Cunningham and J. Patrick, editors, *HLT-NAACL 2003 Workshop: Software Engineering and Architecture of Language Technology Systems (SEALTS)*, pages 59–66, Edmonton, Canada, 2003. Association for Computational Linguistics.
10. G. Herzog, A. Ndiaye, S. Merten, H. Kirchmann, T. Becker, and P. Poller. Large-scale Software Integration for Spoken Language and Multimodal Dialog

Systems. *Natural Language Engineering*, 10, 2004. Special issue on Software Architecture for Language Engineering.
11. W. Hill, D. Wroblewski, T. McCandless, and R. Cohen. Architectural Qualities and Principles for Multimodal and Multimedia Interfaces. In Blattner and Dannenberg [3], pages 311–318.
12. P. Maes and D. Nardi, editors. *Meta-Level Architecture and Reflection*. North-Holland, Amsterdam, 1988.
13. M. T. Maybury, editor. *Intelligent Multimedia Interfaces*. AAAI Press, Menlo Park, 1998.
14. M. T. Maybury and W. Wahlster. Intelligent User Interfaces: An Introduction. [15], pages 1–13.
15. M. T. Maybury and W. Wahlster, editors. *Readings in Intelligent User Interfaces*. Morgan Kaufmann, San Francisco, 1998.
16. S. Oviatt. Multimodal Interfaces. In J. A. Jacko and A. Sears, editors, *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications*, pages 286–304. Lawrence Erlbaum, Mahwah, NJ, 2003.
17. G. E. Pfaff, editor. *User Interface Management Systems: Proceedings of the Seeheim Workshop*. Springer, Berlin, 1985.
18. N. Reithinger, J. Alexandersson, T. Becker, A. Blocher, R. Engel, M. Löckelt, J. Müller, N. Pfleger, P. Poller, M. Streit, and V. Tschernomas. SmartKom: Adaptive and Flexible Multimodal Access to Multiple Applications. In *Proc. of the 5th Int. Conf. on Multimodal Interfaces*, pages 101–108, Vancouver, Canada, 2003. ACM Press.
19. N. Reithinger, G. Herzog, and A. Ndiaye. Situated Multimodal Interaction in SmartKom. *Computers & Graphics*, 27(6):899–903, 2003.
20. T. Richardson, Q. Stafford-Fraser, K. R. Wood, and A. Hopper. Virtual Network Computing. *IEEE Internet Computing*, 2(1):33–38, 1998.
21. J. W. Sullivan and S. W. Tyler, editors. *Intelligent User Interfaces*. ACM Press, New York, 1991.
22. W. Wahlster. SmartKom: Fusion and Fission of Speech, Gestures, and Facial Expressions. In *Proc. of the 1st Int. Workshop on Man-Machine Symbiotic Systems*, pages 213–225, Kyoto, Japan, 2002.
23. W. Wahlster. SmartKom: Symmetric Multimodality in an Adaptive and Reusable Dialogue Shell. In R. Krahl and D. Günther, editors, *Proc. of the Human Computer Interaction Status Conference 2003*, pages 47–62, Berlin, Germany, 2003. DLR.
24. W. Wahlster, N. Reithinger, and A. Blocher. SmartKom: Multimodal Communication with a Life-Like Character. In *Proc. of Eurospeech'01*, pages 1547–1550, Aalborg, Denmark, 2001.