Situated Delegation-Oriented Multimodal Presentation in SmartKom

Jochen Müller, Peter Poller, Valentin Tschernomas
German Research Center for Artificial Intelligence GmbH (DFKI)
Stuhlsatzenhausweg 3
66123 Saarbrücken, Germany
E-Mail: {jmueller,poller,tscherno}@dfki.de

Abstract

One of the major scientific goals of SmartKom is to design a new human-machine interaction metaphor for a multimodal dialog system that combines speech, gesture, and mimics input with speech, gesture and graphics output. The system behavior is based on the new situated delegation-oriented dialog paradigm (SDDP) in the sense that a communication assistant, realized as an animated life-like character, serves as the communication partner of the user. In this paper we focus on the multimodal output of SmartKom, showing how these highly ambitious tasks of the SmartKom system are managed and realized on the output side, i.e., how the communication assistant adapts its behavior to the available output media and modalities with respect to the current dialog situation.

Introduction

One of the most prominent tasks for dialog systems today is the switch from monomodal spoken dialog systems to multimodal systems that permit gestures, graphics, and even mimics both in input and output in order to get closer to natural multimodal human-machine interaction.

SmartKom (www.smartkom.org) is a multimodal dialog system that supports the situated understanding of possibly imprecise, ambiguous, or partial multimodal input and the generation of coordinated, cohesive, and coherent multimodal presentations (Wahlster, Reithinger, & Blocher 2001). Interactions in SmartKom are managed based on representing, reasoning, and exploiting models of the user, the domain, the task, the context, and the media and modalities themselves.

One of the major scientific goals of SmartKom is to design new computational methods for the seamless integration and mutual disambiguation of multimodal input and output on a semantic and pragmatic level. SmartKom is based on the situated delegation-oriented dialog paradigm (SDDP, (Wahlster, Reithinger, & Blocher 2001)), in which the user delegates a task to a virtual communication assistant, visualized as a life-like artificial character on a graphical display.

According to SDDP, SmartKom breaks with the traditional desktop interaction metaphor that is based on WIMP (windows, icons, mouse pointer) interaction. We radically reduce the content of the graphical user interface to only those elements (e.g., graphics) that are relevant to the user. These are presented on a black background. Thereby, our communication assistant also gets capabilities that humans don't have to act like a magician on a stage with no background behaving as a butler of the user. In this sense we investigate a new interaction metaphor within a multimodal dialog system aiming to reach new fields of human-machine interaction by taking advantage of the unlimited virtuality of the communication assistant. In this sense we go beyond the various aspects of human-human conversation in virtual models of real worlds described in (Cassell *et al.* 2000) by extending them to new multimodal areas that are unreachable in reality.

There are three different instances/scenarios of SmartKom (see figure 1) with different applications/hardware. SmartKom PUBLIC is a system which can be used as a multimodal information kiosk. The gestures of the user are tracked by the Siemens Virtual Touch screen (SIVIT) and the display is projected on a screen. In the SmartKom HOME scenario, a Tablet PC is used to show the visual output of SmartKom. Here, the system can be used as an information system at home, e.g., as an EPG and for device control (TV set, video cassette recorder). SmartKom MOBILE is an instance of the SmartKom system with its display on a PDA (Compaq iPAQ) that is used for mobile tourist information and navigation (car and pedestrian).

In order to reduce the complexity of modules and representations, SmartKom can be divided into a set of version-specific output modules that have to be adapted to the concrete applications and hardware devices while a so called multimodal dialog backbone, i.e., a set of modules that are responsible for analysis, dialog management and presentation, works independently of the current applications and hardware devices. Figure 1 also illustrates this distribution.

In the rest of the paper, we focus on the multimodal output generation in SmartKom within the SDDP metaphor described above. The general problem are the various situations (and presentation parameters imposed by them) that all constrain the generation of appropriate natural multimodal system output in heterogeneous aspects. To mention some prominent presentation constraints in SmartKom: First, the user communicates with a virtual communication assistant which requires the generation of natural and appropriate an-

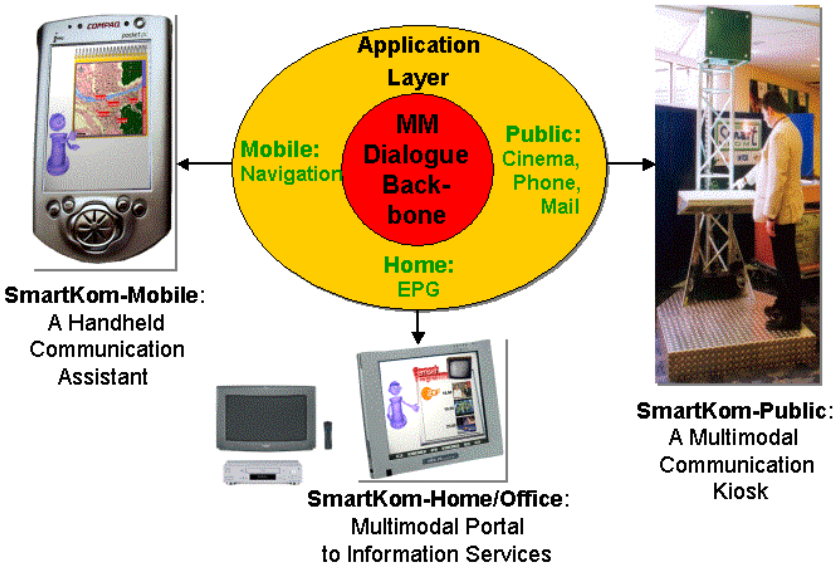

Figure 1: SmartKom backbone and scenarios

imations including lip synchronization for speech output. Second, there are three different versions of Smartkom that differ in application scenarios and hardware devices at hand that impose at least different time and space requirements for graphics, animations and speech. Third, the distribution of presentations to the available media and modalities (speech, graphics, gestures) also depends on user states (e.g., anger, no access to the display).

In this paper, we describe the methods and the realization of the multimodal output generation in SmartKom in more detail, focusing first on the distribution of the relevant computational steps in a modular architecture, and then on the computational behavior of these modules including their interfaces and interactions and dependencies in the different scenarios.

The Modular Presentation Architecture

In this section we describe the modular architecture of the components involved in the output generation of SmartKom leaving aside input modules and the dialog management.

In SmartKom, the input for the Presentation Planner is provided by the Action Planner module, which decides what is to be presented to the user (so called presentation goal) while the decision how that is done is the task of the Presentation Planner which also contributes the distribution of presentations to the available output media and modalities.

Another source of presentation goals can be the Dynamic Help module which permanently supervises the global state of the overall system. In case of problems, e.g., speech recognition difficulties or incomplete user intentions it also produces presentation goals which are intended to give hints to the user about what the problem was and possibly how it can be solved by sending appropriate presentation goals for that to the presentation planner.

Figure 2 shows a simplified data flow architecture of the output modules of SmartKom focusing on the Presentation Planner and the Display Manager. The Presentation Planner cooperates with a Text Generator and a Speech Synthesizer in the sense that speech presentation tasks are published to them for generation and synthesis while the synthesizer in turn publishes a phonetic representation of the synthesized audio signal. The Display Manager is responsible for performing the multimodal output presentation thereby having access to the Audio Output module to trigger the speech output signal. The details of Presentation Planner and Display Manager are the subject of the following sections.

After the creation of a detailed presentation plan within the Presentation Planner, the plan is evaluated in order to generate all the data that are necessary to perform the multimodal presentation itself. To do so, we employ several generators:

- Generator for the script for the presentation agent Smartakus (PERSONA generator) (Müller 2000)
- Text Generator (Becker 2002)
- Speech Synthesizer
- Graphics Generator
- Display Manager

Once all these parts are realized, corresponding representations are passed to the Display Manager in order to perform the multimodal presentation on the available output de-

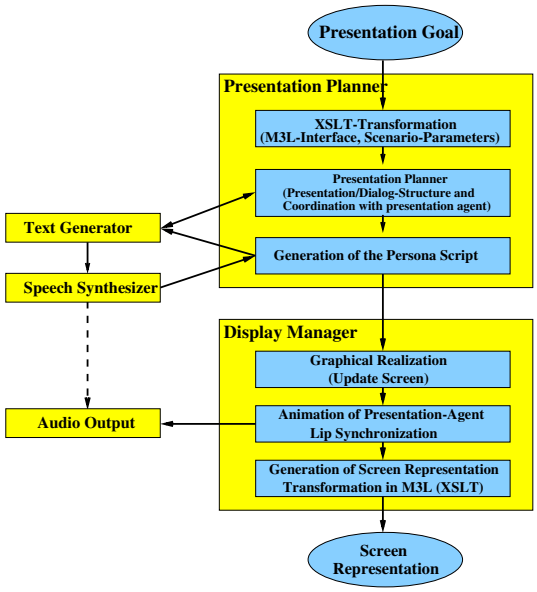

Figure 2: Presentation-Pipeline in SmartKom

vices. The details of the Display Manager are described in section .

The Gesture Analysis module of SmartKom needs a representation of all objects that are currently visible on the screen in order to relate gesture input of the user to the current presentation on the screen. Thus, the display manager finally generates a representation of all objects on the screen, including coordinates of their bounding boxes and links to the world objects they represent. This document is finally send to the gesture analysis module.

Presentation Planning

In this section we describe the Presentation Planner in more detail. All information that flows between modules is represented as an XML-document conforming to the schema-based "Multi-Modal Markup Language - M3L" which was developed in SmartKom. The input of the Presentation Planner (see figure 3 for an example) is sent by the Action Planner and consists of a M3L-document that contains an abstract representation of the system intention to be presented multimodally to the user.

Currently, the input consists of data which are pre-divided into sections for graphics (<graphicalPresentationGoal>) and for speech output (<speechPresentationGoal>). This division is not fixed for the presentation (e.g., the movie titles could also be presented by speech). We currently adapt the interface between Action Planner and Presentation Planner such that the M3L input representation for the Presentation Planner gets modality independent.

The M3L-document in figure 3 consists of a graphical presentation goal which represents the intention to <inform> the user by giving an <overview> about a <broadcast> (among others) with the identifier "ap_22" (i.e., the movie "Asso" that is shown from 20:15 to 22:20 on <channel>

```
<presentationTask>
<subTask goalKey="ap_14">
  <graphicalPresentationGoal>
// ``means: inform the user by giving''
    <inform>
      <concreteResult>
        <informFocus>
          <graphicalRealizationType>
// ``an overview''
            overview
          </graphicalRealizationType>
          <content idReference="ap_22"/>
             ...
        </informFocus>
      </concreteResult>
    </inform>
    <abstractPresentationContent>
      <taskSpecification>
        ...
      </taskSpecification>
      <result>
// ``about a broadcast with identifier ap_22''
         <broadcast id="ap_22">
          <beginTime>
            <time>
              <function>
                <at>
// ``that begins at:''
                  2000-12-13T20:15:00
                </at>
              </function>
            </time>
          </beginTime>
          <endTime>
            <time>
              <function>
                <at>
// ``and ends at:''
                  2000-12-13T22:20:00
                </at>
              </function>
            </time>
          </endTime>
          <avMedium>
            <title>
// ``has the title:''
              Asso
            </title>
          </avMedium>
          <showview/>
          <channel>
            <name>
// ``and is telecasted on channel:''
              SAT1
            </name>
          </channel>
        </broadcast>
          ....
      </result>
    </abstractPresentationContent>
  </graphicalPresentationGoal>
  <speechPresentationGoal>
// ``comment on the graphical presentation goal above''
       ....
  </speechPresentationGoal>
</subTask>
</presentationTask>
```

Figure 3: M3L-input document for Smartkom Presentation System

"SAT1"). The speech presentation goal not shown in detail here consists of a comment on that.

General Approach

The first computational step being performed on the M3L input document is a transformation of the document into the special input format of the core planning component PrePlan ((André 1995),(Tschernomas 1999)) by application of an appropriate XSLT-stylesheet (see figure 2).

The use of stylesheets ensures flexibility with respect to the input structure of the Presentation Planner. The only change that is needed to process syntactically different input formats is the adaption of the XSLT-stylesheet such that the entire planning process inside PrePlan remains unchanged. Similarly, different situation-dependent XSLT-stylesheets that reflect different dialog situations are used. These dialog situations impose different input parameter settings for the Presentation Planner. Some examples for the presentation parameters are:

Current scenario: There is Smartkom MOBILE, SmartKom HOME and SmartKom PUBLIC.

Display size: Currently we have one display with the resolution 1024x768 pixel for use with the SIVIT and the Tablet PC. The display size of the used Compaq iPAQ is 240x320 pixels.

Language: The SmartKom system currently supports German and English.

Available User Interface Elements: The planner must be informed what graphical elements can be used on the display (e.g., lists for TV or cinema movies, seat-maps, virtual phone or fax devices)

User Preferences: Does the user prefer spoken output (i.e., while using SmartKom MOBILE in a car) or graphical output ?

Style: We support different design styles.

The stylesheet transformations additionally add scenario-specific or language-specific data to the knowledge base of the presentation planner. For example, translations for labels and a logical description of available screen-layout elements are inserted. The Presentation Planer decides what to present and how to present. For this presentation task, a simple XML-transformation by using XSL-stylesheets is not sufficient (Wilcock 2001).

The Presentation Planner starts the planning process by applying a set of so-called presentation strategies which define how the facts are presented in the given scenario. They define the knowledge how to decompose the complex presentation goal into primitive tasks.

For example, they determine whether a description is given verbally or graphically or which graphical elements like lists or slide-shows are used in the presentation. The decision is based on the constraints of the strategies and the data in the knowledge base. The presentation goal is decomposed into less complex ones until all goals have been expanded to elementary presentation acts. The result of this process is a hierarchical plan which reflects the structure of the presentation. An example of a presentation strategy is given in figure 4.

```
(define-plan-operator
 :HEADER (A0 (Show-Sights ?sightlist))
 :CONSTRAINTS
   (BELP (layout ?layout canshow picture))
 :INFERIORS (
    (A1 (SetLayout ?layout))
    (A2 (seq-forall (?url) with
      (BELP (objectpicture ?sightlist ?url))
        (AI (Add-Picture-URL ?layout ?url))))
    (A3 (Add-Description ?layout ?url))
    (A4 (Persona-Speak))
 ))
```

Figure 4: A presentation strategy in SmartKom

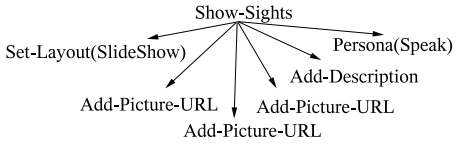

Figure 5: A part of an instantiated presentation plan in SmartKom

The example shows, how a collection of touristic sights should be included in the presentation. The procedure to present a touristic sight is as follows:

1. Choose a suitable layout. The layout should be able to show some pictures and short textual descriptions. Currently, we use a slide show for the pictures.
2. Add pictures to the current presentation (here we assume that the pictures are available by accessing an URL).
3. Add the textual description to the presentation.
4. Instruct the presentation agent Smartakus to speak a short introduction.

The result of the planning process is a planning tree (figure 5).

This tree is evaluated in order to create a presentation script that is finally passed to the Display Manager as an agenda of the resulting presentation to be performed on the available output media and modalities.

The resulting presentation for such a presentation goal according to the presentation strategy above is shown in figure 6.

Figure 7 shows an example presentation for a TV program in the PUBLIC scenario. Smartakus moved to the left of the TV graphics and then gives a short comment about the data being presented by the graphics.

A special sub-task of this planning process is the generation of animation scripts for the presentation agent Smartakus that have to be related and coordinated with the graphical presentations. The main task of the Presentation Planner here is the selection of gestures that supplement appropriately the graphical output. Details about the gesture knowledge base and the gesture selection criteria are presented in

Figure 6: A presentation in SmartKom MOBILE on a PDA

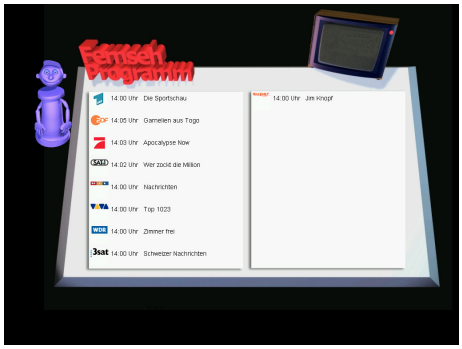

Figure 7: Presentation of the actual TV program in SmartKom PUBLIC.

section . In our implementation, the Presentation Planner constructs a special script for Smartakus that is based on the script-language of the PERSONA system (Müller 2000) on an abstract level. The PERSONA system has two components: the PERSONA Generator and the PERSONA Player The PERSONA Generator decomposes the abstract commands written in the persona script into elementary commands that can be handled by the PERSONA Player. While the generator is integrated in the Presentation Planner, the player is integrated in the Display Manager.

In conjunction with these animations, the lip synchronization data are prepared. A part of the presentation is usually performed by speech output. To generate it, an M3L-representation of the intended meaning is passed to the text generation module (Becker 2002). It generates corresponding utterances which are passed to the speech synthesis module. The speech synthesizer produces two output representations, the audio signal to be spoken and a special representation containing all the phonemes of the speech output and the exact time points at which they occur in the speech output. The latter representation is the input of the lip synchronization sub-module whose task is the generation of appropriate synchronized mouth movements. The result is a lip synchronization script that is also passed to the Display Manager. More details about lip synchronization procedure are described in section .

After all these presentation scripts are generated, they are sent to the Display Manager. Since in the current SmartKom system, the Presentation Planner and the Display Manager are two separate modules, which can run on different machines, the scripts are first converted to a corresponding M3L-document before they are sent to to the Display Manager.

The Display Manager evaluates these scripts and performs the planned actions accordingly. After the end of the presentation, a representation of the current screen content is generated and send as a M3L-document to other SmartKom modules for further processing (i.e., gesture recognition). These procedures are described in section .

Gesture Generation

The animations of the presentation agent Smartakus have to be related to the graphical output for pointing gestures and to the speech output for synchronous mouth movements.

Gestures

In SmartKom, gesture animations have to be aware of two kinds of synchronization in oder to realize a natural behavior of the communication assistant on the output screen:

- All deictic gestures have to be “graphically” synchronized with the display of the corresponding graphical output they are related to.
- The acoustic speech output has to be synchronized by appropriate lip movements of the communication assistant Smartakus.

Behavior Catalog

The knowledge source for Smartakus' behaviors and gestures is a catalog of predefined GIF animations. Smartakus is statically modeled in 3D as a life-like character with 3D-Studio-Max. But for efficiency reasons the deeply 3D-modeled gestures are rendered as animated GIF's. These categorized GIF's form the behavior knowledge base of the presentation sub-system of SmartKom.

The animations themselves are further subdivided into preparation phase, stroke phase and retraction phase.

On the topmost level we distinguish the following kinds of behaviors in accordance to the “situations” in which Smartakus might be and which are performed appropriately by the Display Manager as soon as corresponding information arrives. Each of the categories below contain at least one category-specific behavior:

idle: Idletime-gestures signal that the system awaits new user input.

Figure 8: A Pointing Gesture of Smartakus

listening: Gestures indicating that Smartakus is currently listening to the speech input of the user (indicated by moving a hand behind an ear).

mediating: In the telephone application the user talks to a third party which should be signaled by a disappearing Smartakus.

moving: Gestures that permit Smartakus to move around the screen.

presenting: Gestures that are used for the multimodal presentations, e.g., pointing gestures to interesting regions on the screen or eye movements with a similar intention.

start-stop: Gestures that are used at the beginning and at the end of dialogs to let Smartakus appear or disappear, i.e., by minimizing Smartakus to an 'I' and vice versa.

stranded: Gestures that are used to indicate that the system had problems to process the user input successfully.

working: Gestures to indicate that the system is currently the processing the last user input.

lipsync: Visemes used for the lip synchronization (see below)

Figure 8 shows an example of a pointing gesture to a visible output object. Smartakus presents a phone to the user, points to the phone and comments it by saying "Please dial a number!"

In this example the script for the presentation consists of a surface visualization of a cellular phone, a Smartakus motion to the left of the phone, a pointing gesture and a lip synchronized animation for multimodal speech output.

Lip Synchronization

The Display Manager is responsible for the realization of visible output while the Audio Output module realizes speech output. Thus, audio output and visual output are performed independently from each other and can even be processed and realized on different machines. When presenting both modalities, we have to merge them to get lip synchronized output.

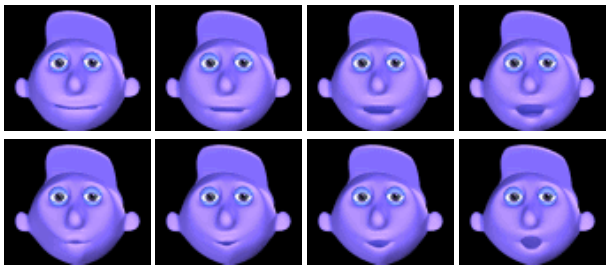

Figure 9: Smartakus' Visemes

Knowledge Sources The lip synchronization is based on an underspecified mapping between acoustic and visual units (so called phonemes and visemes). In our system a viseme is defined as a specific mouth position picture (www.whatis.com: generic facial image).

However, closely cooperating with the speech synthesis group at IMS, University of Stuttgart (Schweitzer, Dogil, & Poller 2001), we found that due to the cartoon-like character of Smartakus (neither tongue nor teeth are visible) only a limited variety of mouth/jaw positions or movements are possible at all. Consequently, the only parameters that we found relevant to describe mouth positions are the lip rounding and the jaw opening. Lip rounding is a binary parameter, because lips are either rounded or not. On the other hand, for jaw opening we identified four different values being reasonable and sufficient especially with respect to the precision that is needed in SmartKom in all three scenarios. Figure 9 shows the 8 different visemes that are currently used in the PUBLIC scenario. They resulted from several optimization procedures involving our designer and the speech synthesis group of SmartKom. The first row shows the visemes with unrounded lips and 4 different opening degrees, the second row the corresponding rounded lips.

Then, again in cooperation with the synthesis group at IMS, we developed an underspecified mapping of phonemes to the identified visemes. We found that almost every phoneme has a corresponding viseme, while only a few of them (plosives and diphtongues) have to be mapped to at least two visemes to visualize their articulation appropriately (Schweitzer, Dogil, & Poller 2001). Thus, in such cases the mapping partly becomes a one-to-many mapping in the sense that one phoneme can be mapped to more than one viseme. Furthermore, the mapping has to be partly underspecified in lip rounding and jaw opening as well to be able to take coarticulation effects into account.

Since the Audio Output module and Display Manager are two separate modules in SmartKom that in principle work independently from each other, the idea to synchronize lip movements with speech output is to synchronize the individual time points at which corresponding acoustic and visual

events occur as exactly as possible.

The Algorithm

The speech synthesis module in SmartKom does not only produce audio data but also a detailed representation of the phonemes and their exact time points (in milliseconds) inside the audio signal. Based on this representation and the phoneme-viseme mapping mentioned above the Presentation Planner generates a lip animation script for Smartakus that is then executed by the Display Manager during speech output.

The animation script is generated by a stepwise procedure iterating over the phonemes that consecutively specifies the concrete viseme(s) and their exact time points at which they have to be shown in one stream. The determination of a viseme considers the neighboring visemes and follows the following criteria to fix an underspecified viseme

- avoid "extreme" mouth opening degrees whenever possible
- prefer similar visemes to avoid that consecutive visemes differ to much

In terms of figure 9 the procedure always tries to select a viseme that has a common borderline with the previous viseme whenever possible (also by inserting intermediate visemes if necessary).

The concrete time points of the visemes do not coincide with the beginning time of phonemes. Currently, the viseme times are decremented by a constant time (20 ms) because lip and jaw movements take place first before the articulation starts (e.g., when articulating an ' m' , the lips have to be closed first before the articulation is possible at all). We are currently investigating the question whether the time shift is the same for all phonemes/visemes or is phoneme/viseme dependent. At this point, we benefit from the extremely small amounts of time (milliseconds) exceeding the recognition capacity of the human eye.

Display Management

The first task of the Display Manager is to determine a layout for the the objects to be presented to the user. The knowledge base of the display manager is a set of different graphical layout elements/frames (e.g., a frame for a TV program as shown in figure 7, a frame for maps as shown in figure 6, a cellular phone as shown in figure 8). Then the frame has to be filled dynamically by content elements from the input (e.g., movie titles, channel logo, ...) with respect to space restrictions of the selected frame. On the other hand, the cellular phone is an example of a completely fixed layout element that virtually models a physical unit and the gesture interactions with it. Immediate feedback of key-pressing events is given by inverting the key visualization as well as showing the digit on the small display of the phone.

The second task of the Display Manager is the integration of Smartakus into the presentation and the coordination of its animations with the graphical output and the speech output. Graphical presentations are referred to by Smartakus. Thus, they are presented first. Then the Smartakus animations including pointing gestures are executed. Finally, the speech output is initiated by the display manager. It defines a fixed time point at which the audio signal has to start and itself starts the lip animation at the same time point by using the PERSONA player.

Finally, the Display Manager computes a screen representation in M3L-format that is published to the SmartKom system in order to permit the gesture analysis component to relate gestures on screen objects to the world objects they represent.

Implementation

All three versions of SmartKom are fully implemented as demonstrator systems. SmartKom is realized as a multi-blackboard system that supports several programming languages for Linux and Windows NT by providing corresponding integration API's. Furthermore, the system architecture permits the distributed execution on different machines which is necessary because some modules run under Linux only while other run under Windows NT only.

The Presentation Manager and the Display Manager are implemented in Java. So, both modules run under either Windows NT or Linux which ensures the flexibility to integrate them into the SmartKom system.

Conclusion and Current Work

We presented major parts of multimodal presentation in the multimodal dialog system SmartKom. The output subsystem of SmartKom is controlled by the Presentation Planner that is responsible for the planning of multimodal presentations and the Display Manager whose task is the multimodal execution of the presentation plan. A key feature of both modules is the strict separation of knowledge sources and programming code which significantly facilitates the adaption to extended input structures as well as the integration of them into other environments.

Currently, we are working on the improvement of gesture generation. The gesture generation is based on a structured predefined but inflexible gesture catalog. In the current demonstrator version it is not yet implemented, that Smartakus performs two different types of synchronized gestures simultaneously, e.g., pointing during speech based on gesture-speech-aligned data. This could be achieved on a basic level by dividing Smartakus graphically into two autonomously operating body elements that are synchronized individually, e.g., the head and the body. Since our speech animation concerns the head only, the body could perform deictic gestures (e.g., pointing to a specific graphic output) as long as the head is not involved in these gestures. Otherwise we would have to copy the 8 visemes for different head positions. Beyond this, our goal is to develop models and techniques that allow for dynamic gesture generation.

Acknowledgments

This work was funded by the German Federal Ministry of Education, Science, Research and Technology (BMBF) in the framework of the SmartKom project under Grant 01 IL 905 K7. The responsibility for the contents lies with the authors.

We'd like to thank Tilman Becker and Norbert Reithinger for their invaluable and helpful comments on earlier versions of this paper.

References

André, E. 1995. *Ein planbasierter Ansatz zur Generierung multimedialer Präsentationen*. Ph.D. Dissertation, Universität des Saarlandes.

Becker, T. 2002. Practical, Template–Based Natural Language Generation with TAG. In *Proceedings of TAG+6*.

Cassell, J.; Sullivan, J.; Prevost, S.; and Churchill, E. 2000. *Embodied Conversational Agents*. Cambridge, MA, USA: The MIT Press.

Müller, J. 2000. *Persona: Ein anthropomorpher Präsentationsagent für Internet-Anwendungen*. Ph.D. Dissertation, Universität des Saarlandes, Saarbrücken.

Schweitzer, A.; Dogil, G.; and Poller, P. 2001. Gesture-speech interaction in the smartkom project. Poster presented at the 142nd meeting of the Acoustical Society of America (ASA). Ft. Lauderdale, FA, USA, *http://www.ims.uni-stuttgart.de/˜schweitz/documents.shtml*.

Tschernomas, V. 1999. *PrePlan Dokumentation (Java-Version)*. Deutsches Forschungszentrum für Künstliche Intelligenz, Saarbrücken.

Wahlster, W.; Reithinger, N.; and Blocher, A. 2001. Smartkom: Multimodal communication with a life-like character. In *Proceedings of Eurospeech 2001, 7th European Conference on Speech Communication and Technology*, volume 3, 1547 – 1550.

Wilcock, G. 2001. XML for Natural Language Generation. In *Proceedings of the 1st NLP and XML Workshop, co-located at the NLPRS2001*.