Orchestrating Output Devices - Planning Multimedia Presentations for Home Entertainment with Ambient Intelligence

Christian Elting

European Media Lab GmbH
Schloss-Wolfsbrunnenweg 33
D-69118 Heidelberg
christian.elting@eml-d.villa-bosch.de

Abstract

In this paper we motivate the use of personalized multimedia presentations involving multiple output devices in a home entertainment environment. First we illustrate our vision and analyze the requirements. Afterwards we present the architecture of our system focusing on the output coordination strategy, which achieves a coordination of multiple output devices by means of an AI planning approach.

Then we present our prototype implementation, which generates movie-related. The implementation consists of a TV set displaying an animated character, a PDA, which acts as a remote control and a 17" digital picture frame, which displays pictures and renders speech. We conclude with an overview over related work.

1. Introduction

In Mark Weiser's vision of ubiquitous computing [1] he foresaw a future in which computing would indeed be pervasive and take place in clothing as well as in the tapestry without us even noticing. The first step towards this vision has already been taken. Embedded systems are nowadays already contained in many everyday devices like pens, watches, cellulars, hifi stereos or TV sets.

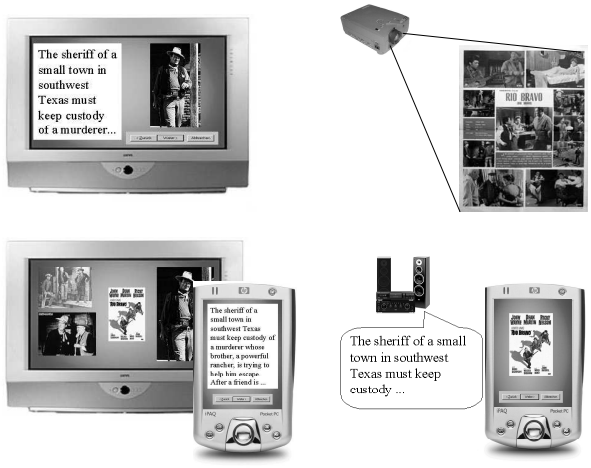

Figure 1: Examples for multi-device presentations of movie media

A similar development seems to take place concerning display technology. Displays are becoming cheaper and larger. They are also becoming more flexible and enter new domains like clothing. The same is true with projectors, which are already present in many homes. Therefore displays are becoming ubiquitous and become increasingly important for the scenarios of ambient intelligence [2].

However what is still missing are ways to coordinate a set of displays or –generally spoken– output devices. Figure 1 gives four examples of how the same content (EPG data for a movie consisting of a synopsis and several stills) can be adapted to different output devices in a home entertainment scenario. The presentations are shown on a TV set, a projector, a combination of a PDA and a TV set and a combination of a PDA and a hifi stereo.

In this paper we present a dialogue system, part of which is an output coordination strategy. This strategy is able to generate presentations of movie media for dynamic and heterogeneous sets of output devices. In order to determine the requirements for such a strategy we illustrate our vision of multi-device presentations in the next section.

1.1. Scenario

Michael and Jean are living in a flat, which has just been equipped with their new home entertainment system. The system comprises a hifi stereo, a TV set, a 17" digital picture frame, and a PDA.

Michael relaxes on the sofa and switches on the PDA. He wants to watch a movie, but he is not sure what is on TV tonight. The PDA displays a list of options including "play music" and "EPG" (electronic program guide). Michael chooses "EPG". After browsing through the program Michael wants more information about the Movie "Gladiator". He selects the title of the movie and requests more information to be presented. The system looks for the title of the movie in a movie database, which contains a synopsis of the movie as well as multimedia like stills, trailers, scenes or tracks from the score of the movie.

On the PDA Michael first receives a short textual synopsis of the movie. However Michael wants to get some orange juice and chooses to let the text be read by a speech synthesis, while he is going into the kitchen. On his way back he switches on the TV set and changes to a special multimedia channel. Automatically the system presents additional pictures, which could not be presented simultaneously with the graphic text on the small 320x240 screen of the PDA.

Additionally an animated character is present on the TV screen. Michael uses the microphone of the PDA for speech input and says "I would like to see more". The character says "I have got a trailer and one track of the soundtrack of 'Gladiator'." "Show the trailer.", answers Michael. The trailer is presented on the digital picture frame, which is able to display MPG video with higher resolution than the 640x480 TV screen. Afterwards Michael says "Now play the soundtrack." Subsequently the track starts to play on the hifi stereo, which provides the best sound quality. In order not to

interfere with the soundtrack the animated character now uses speech bubbles instead of speech.

Jean also wants to use the system to get information about the movie “Notting Hill”, which is shown on TV tonight. However she does not prefer to use the animated character and switches it off. As a result more screen space is available to present stills from the movie.

Afterwards Michael and Jean are well informed about this evening’s TV shows and now only need to decide, which movie will be watched: “Gladiator” or “Notting Hill”: A decision, which they have to make themselves.

1.2. Requirements

From this scenario several requirements arise, which are summarized in figure 2.

The output coordination strategy has to take into account that output devices and services enter and leave the system at any time. E.g., in our scenario Michael turns on the TV set and changes to the multimedia channel thereby activating additional output services. Therefore the output coordination strategy needs to plan presentations with a variable duration and layout complexity depending on the number, types and resources of the available output devices. It is necessary that output devices describe themselves to the system. Such a self-description has to comprise output services (acoustic speech output or graphic video output) as well as resource restrictions (resolution, memory, CPU power and CPU load). In our scenario this information is used by the strategy to switch to graphics after the soundtrack started playing.

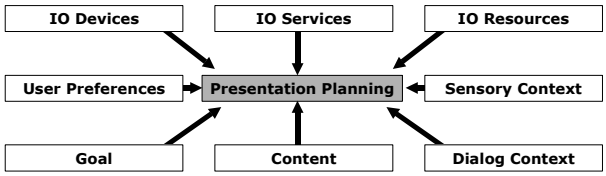

Figure 2: Dependencies of presentation planning

A presentation has to be adapted to the sensory context of the user. If a user chooses to leave the room the output might be redirected to the output devices in the new room. Apart from output devices it is also necessary to model input devices, services and resources. A set of pictures might be presented on the display as thumbnails, which can be accessed via an infrared remote control. This would not be possible if the remote control were unavailable.

The dialogue context has also to be taken into account. This especially includes the output history, which serves to generate future presentations in a style, which is convenient to earlier presentations.

By means of user preferences the content can be customized for different user groups. Some people might prefer presentations including an animated life-like character while others prefer graphics-only interactions.

Additionally a presentation has to be adapted according to the content of the presentation. Graph data can be visualized graphically very well, but only inadequately rendered by means of speech. ASCII text can often be rendered graphically as well as by means of speech.

Finally the goal of the presentation determines the presentation style. A warning might be rendered by means of speech in order to reach the user’s attention. General information might only be rendered by means of a static text information window, which can be read by the user when needed.

2. Architecture

In this chapter we describe the architecture of our system, which is shown in figure 3. For a more detailed description we refer to [3]. In our architecture each device is composed of components, which are situated on three levels: the user interface level, the control application level and the actuators level.

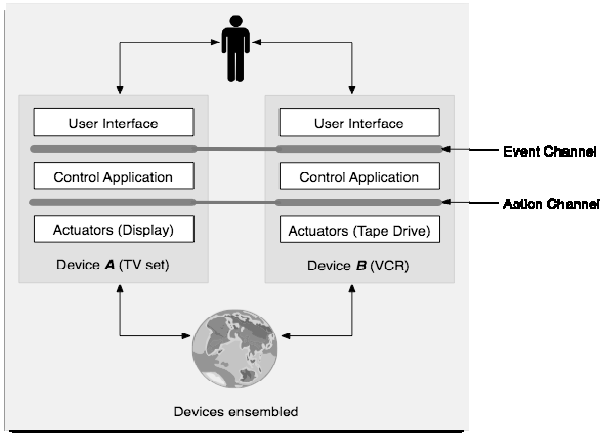

Figure 3: Architecture

For a TV set the set of user interface components consist of a remote control application for input and a graphic user interface (GUI) for output. Input events are passed to the control application level, where a dialogue manager analyzes them.

Between the three levels there are channels, which run across devices. These channels serve to control the flow of information between components. Components never communicate directly but only via channels. On each channel strategies exist, which process events or tasks sent to the channel. Possible strategies are to broadcast the message to all components, which subscribed to the channel. Other strategies are to select the best suitable component for a task or to apply a problem decomposition strategy and distribute a task among a set of components. This paradigm achieves a flexible communication without direct communication. Addressees need not to be known in advance. This is required for dynamic settings, in which devices and components can enter and leave at any time.

Between the user interface level and the control level two strategies are executed. One strategy of the user interface channel is to parse input events from the input level (e.g., GUI input or speech input) and pass them to a proper control component. The other strategy is the output coordination strategy, which we explain in the next section. The task of the output coordination strategy is to process output events from control applications and to distribute them among the set of available output components at the user interface level.

Figure 4 shows a more detailed view of the user interface level, the event channel and the control application level during the generation of a multi-device presentation. For matters of clarification we only display the output part of the architecture and omit the action channel as well as the input components. In this situation the dialogue manager sends a task to the event channel, which is processed by the output coordination strategy. This task contains a set of pictures to be presented on the TV set and was triggered by an earlier request by the user (which is not shown in figure 4).

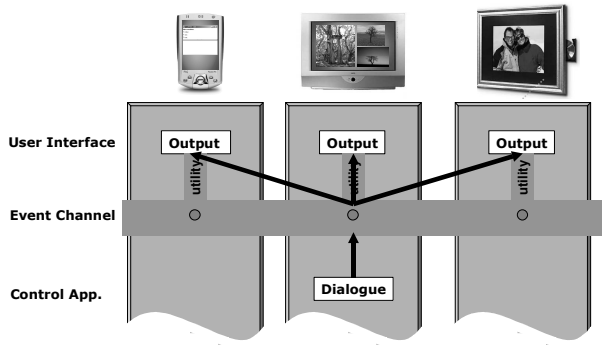

Figure 4: Output coordination strategy on the event channel for three output devices.

The output coordination strategy first asks for available output devices, which subscribed to the event channel.

In this case three output devices are available. A PDA is present with a 320x240 resolution, on which a choice box GUI is located. A TV set is connected, which can display SMIL content [4] and features a 640x480 resolution. The third component is a digital picture frame, which is able to display JPG images and has a 1280 x 960 resolution.

Each output device answers by means of a self-description, which we call utility value. The output coordination strategy collects all those utility values. Based on this information an AI planning approach [5] is used to apply a problem decomposition strategy. As a result each component receives a message containing the content to be displayed.

The first set of pictures is displayed on the TV screen. If possible, more than one picture is displayed at once. Another picture is presented on the digital picture frame. Finally the GUI generates a "continue" option, a "back" option and a "quit" option to allow the user to browse the presentation.

3. Output Coordination Strategy

The content to be presented by the output coordination strategy consists of a movie synopsis given as ASCII text and a set of images. Additionally a dialogue act specifies the goal of the presentation. This makes it possible to distinguish between presentations of general information and presentations of important messages, e.g., error messages. In order to determine the best way to present this content the strategy first looks for available output components, which identify themselves by means of self-descriptions. These self-descriptions are inserted into a presentation planning tool, which determines the best multi-device presentation for the given set of output devices and components.

3.1. Self-descriptions of output components

The self-descriptions of output components consist of two parts. The first part of the self-description is an output component profile, which describes the multimodal output of the component and parameters to be set in an abstract way. E.g., an animated character component may be composed of gestures and lip movements and speech. The output component profile also contains the ID of the physical device on which output is rendered. This makes it possible to identify the available resources as well as other services present on this device. More information about self-descriptions can be found in [6].

Self-descriptions facilitate the integration of new components. Once expressed by the model, an output component of a known type can be integrated without having to adapt any strategies within the presentation planner. For completely new types of output components however new strategies have to be added once for all instances of this type.

3.2. Presentation planning

The output coordination strategy consists of the following steps. First the self-descriptions of the output components and the output device properties are collected from all output components, which subscribed to the event channel. Moreover the event is inserted, which triggered the output coordination strategy and which contains the content to be presented.

Afterwards a hierarchical presentation planning approach [7],[8] is used to decompose the complex output goal into primitive output tasks, which can be executed by each output component. The result of this process is a set of messages to be sent to the output components. Finally a reply containing the result of the presentation planning process is sent to the dialogue manager, which initiated the output coordination strategy.

```
(define-plan-operator
 :header (A0 (build-smil-pres ?rc-id ?im-url))
 :constraints
   (*and* (
      ;; there is an output component of type agent
       (BELP(rc-type ?rc-id-2 agent))
      ;; with speech output
       (BELP(output-unimodality ?rc-id-2 <speech-type>))
      ;; which produced a wav file
       (BELP(output-medium ?rc-id-2 wav ?om-url))))
 :inferiors (
   ;; present a picture with speech
    (A1 (build-img-with-speech ?im-url ?rc-id-2 ?om-url))
   ;; solve constraints and generate smil file
    (A2 (start-mats "pres.smi"))
   ;; send message to output component
    (A3 (send-message ?rc-id "http://myip/pres.smi")))
 :temporal (
    (A1 (m) A2)
    (A2 (m) A3))
  :spatial (
    (centerh A1)
    (centerv A1))
  :start (A1)
   finish (A3))
```

3.3. Presentation strategies

The example in the text box above illustrates the syntax of the planning operators, which define the presentation strategies. The header slot specifies to which terms the strategy can be applied. The constraints slot contains the preconditions for the application of the strategy. The inferior slot contains the acts into which the given act should be decomposed. The temporal slot contains temporal constraints for the application specified by means of Allen interval algebra [9] or quantitative constraints. The spatial field specifies layout constraints for the presentation. The temporal and spatial constraints are resolved by means of a constraint solver [10]. The slots start and finish optionally specify the first resp. last act in inferiors.

The operator in the example serves to generate a SMIL presentation containing an image and a speech file. The image is displayed at the center of the screen. The operator "build-smil-pres" is called with instantiations of the variables rc-id and im-url. The variable rc-id contains the component ID of the SMIL output component, which should play the presentation. The variable im-url contains the image to be presented.

The preconditions in the constraints slot check if a speech synthesis with name rc-id-2 is present, which has output component type agent (i.e., a speech synthesis or an animated character) and produced a WAV file containing speech. The check is conducted by means of the self-description of the output component. The self-description of the speech synthesis is omitted here. If all constraints are satisfied, then the speech file and the image are added to the presentation (action A1), a SMIL file called "pres.smi" is generated (action In the temporal slot of the operator it is assured that first A2) and a message with the URL of the SMIL file is sent to the SMIL player component (action A3).

action A1 is executed, then A2 and then A3 by means of the Allen interval relation "meets". The spatial slot specifies the spatial constraints for the image. In this case the image is centered vertically and horizontally on the screen.

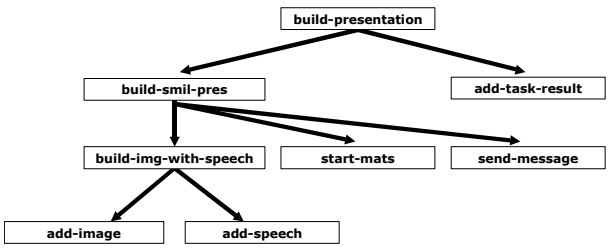

Figure 5: Planning tree of a SMIL presentation.

Now we give an example for a hierarchical decomposition of a presentation task (figure 5). In this example the goal "build-presentation" is decomposed using the presentation strategy "build-smil-pres" shown in the example. Additionally the operator "add-task-result" serves to send a reply to the sender of the task indicating whether all media have been displayed.

4. Implementation

In this section we provide an overview over our demonstrator implementation. First we describe the architecture, devices and output components used in the demonstrator. Then we give examples of system runs.

4.1. Architecture

The current demonstrator implementation (figure 6) comprises four devices: A 3.4 GHz Pentium 4 server with 512 MB RAM running Windows XP, a Digi-Frame DF-1710 17" digital picture frame[1] running Linux, a Compaq PocketPC H3870 running Linux and a Loewe Aconda 9272 TV set[2] with OnlinePlus system running Linux.

Figure 6: TV set with animated character, PDA with GUI and digital picture frame.

Communication between components is achieved via the SodaPop **Erreur ! Source du renvoi introuvable.** infrastructure. The current version is running on the server PC. In the future we intend to replace this local infrastructure with an entirely distributed approach [6].

Figure 7 shows the architecture of the demonstrator including the event channel and the action channel (cf., figures 3, 4). All components as well as the infrastructure were implemented using Java. We implemented the following components. The GUI component on the PDA can be started at any time and is automatically updated according to the current state of the dialogue. It is also possible to start more than one GUI (e.g., on a second PDA) or to use a Via Voice-based speech recognition to control the system [6]. The PDA connects to the SodaPop infrastructure on the server by means of WLAN.

On the TV set we use an output component, which is able to display SMIL content by means of the Xsmiles player[3].

For the digital picture frame we use a remote component on the server, which writes a text file containing a sequence of URLs to WAV files or images to the Samba network drive of the digital picture frame. The digital picture frame automatically loads the new presentation script and displays it. The reason for this is that a Java runtime environment is not yet present on the digital picture frame itself.

We installed an Mbrola-based speech synthesis[4] on the server, which renders ASCII text into a WAV file containing the corresponding spoken text. We also use a character, which is part of the presentation planner software [7] and consists of animated GIF images, which can be synchronously integrated into a SMIL presentation together with speech and pictures. Additionally we are running an HTTP server to share media between devices.

On the actuator level we use an MSAccess-based German movie database[5] containing 11,030 movie synopses to which additional movie stills were added for a limited selection of movies.

[1] http://www.digi-frame.com/products.html
[2] http://www.loewe.de
[3] http://www.xsmiles.org
[4] http://tcts.fpms.ac.be/synthesis/mbrola/
[5] http://www.filmdb.de

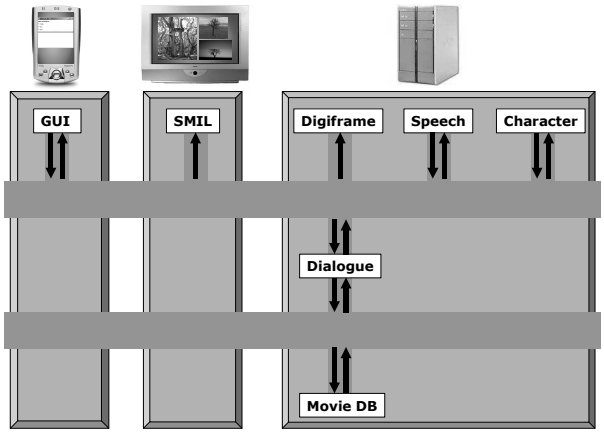

Figure 7: Architecture of the demonstrator.

4.2. Dialogue and presentation strategies

We will now give examples for system runs. The user enters the room and activates the GUI on her PDA. The GUI connects itself to the system and is updated according to the current state of the dialogue.

The movie database is currently connected. Therefore the item "movie information" pops up on the GUI. After selecting it, a list of movies appears from which the user can choose. The same dialogue can also be conducted by means of speech input with a speech recognition component and a wireless radio headset.

When no other device is available for output the synopsis of the movie is presented on the PDA. Due to the restricted PDA layout the text window overlaps the GUI window and has to be closed or minimized to return to the GUI.

If the TV set is switched on and the channel set to a special "home multimedia channel", which shows the X Windows output of the Linux system, the system generates an animated character by means of the SMIL output component. The character reads the synopsis aloud and presents the first image in a newscaster style (figure 8). The animations of the character and the speech from the synthesis are synchronized by means of the SMIL script generated by the presentation planner. Afterwards all pictures are presented in full screen one by one each time the user presses "continue".

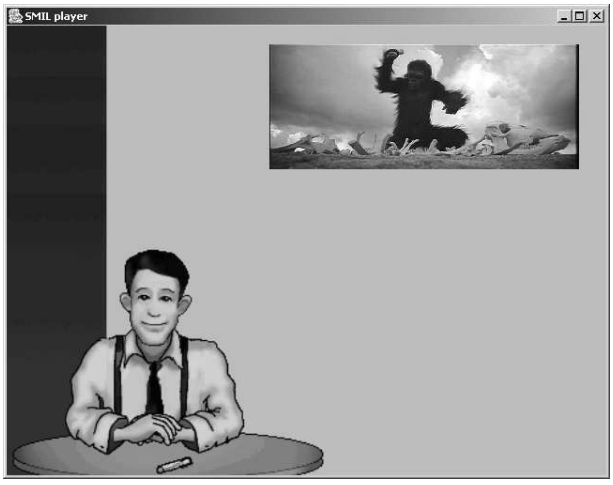

Figure 8: A SMIL presentation containing an animated character and a movie still.

If at any time in the dialogue the digital picture frame is additionally switched on then one additional picture is displayed on it resulting in fewer dialogue steps. If the TV channel is changed or the TV set switched off the voice of the speech synthesis is redirected to the digital picture frame.

In case the speech synthesis component on the server is not started then image-only presentations are generated, because the character cannot be presented without speech. The images are shown in full-screen mode in this case. If the character component on the server is not started then image-speech-presentations are generated.

5. Related Work

In [12] an architecture for multi-device presentation planning is proposed, which generates SMIL presentations for multi-display environments similar to our system. However as a difference to our scenario no acoustic output devices are included in the system.

In the Peach system [13] presentations for multiple PDAs together with a public display are generated. Here the character is able to jump from the PDA to the large public display. The focus of the system is to generate shared multi-user presentations. Instead of SMIL Flash presentations are generated. Due to the museum setting, in which device types are known beforehand, the system does not focus on including heterogeneous sets of IO devices with unknown resources and services.

The Pebbles personal remote controller (PUC) [14] uses self-descriptions of applications to build a PDA-based GUI as well as a speech interface automatically. The applications have to be coordinated manually by the user. However manual coordination is not feasible for output applications, which display complex multimedia content, because the graphic and temporal layout becomes too complex.

[15] describes different types of multi-device user interfaces. The interaction techniques described focus on manually moving data from one device to another and sharing data between devices. We think that those input techniques would be a good addition to our system, in which multimedia content can currently not be moved manually from one device to another.

In the Situated Computing Framework [16] a PDA connects to a set of PCs in a room, on which several applications are running. These applications can export an HTML interface to the PDA or stream media to the PDA similar to our scenario. However it is not possible to combine the display of the PDA with other displays and show different parts of the HTML page simultaneously on different devices.

The system described in [17] dynamically generates presentations involving multiple output devices. However the presentations consist of HTML content and speech, which do not provide the synchronization protocols of SMIL presentations.

The WWICE infrastructure of the Philips Home Lab [18] allows dynamic interactions between mobile devices and TV sets. A similar interaction paradigm is used in the Microsoft Windows Home Concept[1], which connects a tablet PC with different displays of variable sizes.

The use of characters in home entertainment settings is supported by the findings of [19], which show that people respond socially to TV screens and computers. The use of

[1] http://www.microsoft.com/whdc/system/platform/pcdesign/hm-concept.mspx

characters in connection with ambient intelligence is a matter of current research [20].

6. Conclusion and Future Work

We presented the concept, the architecture and the implementation of our output coordination strategy, which generates presentations for a dynamic set of output devices according to the given output services and resources. We presented our prototype implementation running on a TV set, a PDA, a digital picture frame and a server PC.

Next we will conduct a user study, which evaluates different presentation strategies of the prototype implementation and investigate the effects of multi-device presentations. Additionally we intend to couple dialogue management and presentation planning more tightly. We want to enable the user to reference and modify output content ("The image at the upper left in full screen mode"). Moreover we want to integrate user preferences, which allow users to customize the presentations.

We also intend to replace the static movie database with the dynamic EPG database of the TV set, which receives daily updates according to the movies, which are currently shown. Moreover other media like stills or tracks from the score of the movie can be additionally downloaded from the Internet. This would guarantee up-to-date presentations for all movies currently shown on TV.

Acknowledgements

The work presented in this paper was funded by the German ministry for education and research under grant 01 IS C27 B of the project DynAMITE as well as by the Klaus Tschira foundation. The author would like to thank Yun Ding and Ulrich Scholz for comments and proof reading.

References

[1] Weiser, M., The Computer for the 21st Century, Scientific American, 265(3), 94-104, 1991.

[2] Workshop on Ubiquitous Display Environments, Nottingham, England, September 7, 2004.

[3] Hellenschmidt, M. and Kirste, T., A Generic Topology for Ambient Intelligence, European Symposium on Ambient Intelligence, Eindhoven, Netherlands, November 2004.

[4] SMIL, Synchronized Multimedia Integration Language, http://www.w3.org/AudioVideo/.

[5] Ghallab, M., Lau, S., Traverso, P., Automated Planning - Theory and Practice, Morgan Kaufman, 2004.

[6] Elting, C. and Hellenschmidt, M., Strategies for Self-Organization and Multimodal Output Coordination in Distributed Device Environments, Workshop on AI in Mobile Systems 2004, Nottingham, England, September 7, 2004.

[7] André E., Baldes S., Kleinbauer T., Rist. T., CkuCkuCk 1.01 Planning Multimedia Presentations, http://www.dfki.de/imedia/miau/software/CkuCkuCk/manual/manual.html, 2000.

[8] André, E., Concepcion, K., Mani, I. and Van Guilder, L., Autobriefer: A System for Authoring Narrated Briefings. In O. Stock & M. Zancanaro (Eds.), Multimodal Intelligent Information Presentation, pp. 143–158. Springer, 2005.

[9] Allen, J. F., Maintaining knowledge about temporal intervals, Communications of the ACM, 26(11):832–843, November 1983.

[10] Borning, A., Marriott, K., Stucky, P. and Xiao, Y., Linear Arithmetic Constraints for User Interface Applications, ACM Symposium on User Interface Software and Technology, 1997.

[11] Hellenschmidt, M., Distributed Implementation of a Self-Organizing Appliance Middleware, Smart Objects & Ambient Intelligence, Grenoble, France, October, 2005.

[12] Kray C., Krüger A., Endres C., Some Issues on Presentations in Intelligent Environments, European Symposium on Ambient Intelligence, Eindhoven, Netherlands, November 2003.

[13] Kruppa M., The Better Remote Control – Multiuser Interaction with Public Displays, Workshop on Multi-User and Ubiquitous User Interfaces (MU3I), January 13, 2004.

[14] Myers, B. A., Nichols, J., Wobbrock, J. O. and Miller, R. C., Taking Handheld Devices to the Next Level, IEEE Computer, 37(12):36-43, December, 2004.

[15] Rekimoto, J., Multiple-Computer User Interfaces: "Beyond the Desktop" Direct Manipulation Environments, ACM CHI2000 Video Proceedings, 2000.

[16] Pham, T., Schneider, G. and Goose, S., A Situated Computing Framework for Mobile and Ubiquitous Multimedia Access using Small Screen and Composite Devices, ACM Multimedia, Marina del Rey, CA, 2000.

[17] Braun, E., Hartl, A., Kangasharju, J., Mühlhäuser, M., Single Authoring for Multi-Device Interfaces, Workshop "UserInterfaces For All", Vienna, Austria, June 2004.

[18] Baldus, H., Baumeister, M., Eggenhuissen, H., Montvay, A. and Stut, W., WWICE: An Architecture for In-Home Digital Networks". Proc. of SPIE, Vol. 3969, January 2000.

[19] Reeves, B. and Nass, C., The Media Equation: How People Treat Computers, Television and New Media Like Real People and Places, Cambridge University Press, 1999.

[20] Nijholt, A., Rist, T., Tuijnenbreijer, K., Lost in Ambient Intelligence?, Workshop Lost in AmI?, Vienna, Austria, April 25, 2004.