GETTING THE MOBILE USERS IN: THREE SYSTEMS THAT SUPPORT COLLABORATION IN AN ENVIRONMENT WITH HETEROGENEOUS COMMUNICATION DEVICES

Thomas Rist, Patrick Brandmeier, Gerd Herzog, Elisabeth André
DFKI, Saarbrücken, Germany
email: {rist, brandmeier, flint, andre}@dfki.de

Abstract

The wireless integration of small portable devices, such as PDAs or mobile phones with built-in computing power adds another level of complexity on communication and collaborative systems since one has to take into account that the different users may not be equipped equally footed in terms of output and input capabilities. In this paper we present MapViews, Magic Lounge, and Call-Kiosk, three different but related systems that address the integration of mobile communication terminals into multi-user applications. MapViews is a test-bed to investigate how a small group of geographically dispersed users can jointly solve localization and route planning tasks while being equipped with different communication terminals. Magic Lounge stands for a virtual meeting space that provides a number of communication support services and allows its users to connect via heterogeneous devices. Finally, we sketch Call-Kiosk a system that is currently being designed for setting up a commercial information service for mobile clients. All three systems emphasize the high demand for automated design approaches which are able to generate information presentations that are tailored to the available presentation capabilities of particular target devices.

1. Introduction

The increasing quest for mobility together with a large variety of new portable computing and communication devices - including PDA's, Palm computers, and mobile phones with build-in micro computers - add another level of complexity on systems which are to support tele-communication and collaborative work since one has to take into account that the different users may not be equipped equally footed in terms of output and input capabilities. Limited screen real estate, lack of high resolution and color, no support for audio and video are among the typical restrictions on the output site, whereas restrictions on the input site may be due to miniaturized keyboards and GUI widgets, tiny physical control elements, or sparse capabilities for the capture and recognition of gesture, voice and video input. In the context of the European project Magic Lounge[*] we are developing a number of new communication services to overcome communication barriers imposed by the use of heterogeneous communication terminals. In what follows we first present MapViews – a test-bed based on a simulation environment in order to investigate how a small group of geographically dispersed users can jointly solve localization and route planning tasks while being equipped with different communication terminals. After that, we present the Magic Lounge research prototype and show how users can take advantage of new communication support functions. The work on MapViews and Magic Lounge has brought about the concept of an information service for mobile clients with a high potential for commercial exploitation. Call-Kiosk is a prototype that illustrates this kind of information service. Finally, we provide a brief comparison to related work and close with some conclusions and an outlook of further research.

2. MAPVIEWS

2.1 SOLVING COLLABORATION TASKS OVER HETEROGENEOUS COMMUNICATION CHANNELS

To what extent collaborations over heterogeneous communication channels are useful and usable is an issue that is investigated best by means of concrete application scenarios. For the purpose of such a study we have chosen the domain of collaborative travel planning and implemented a demonstrator system. This demonstrator relies on simulations of the user interfaces for both the PDA and the mobile phone which can be run on separate but networked PCs. For example, the phone interface is represented in the simulation through a realistic image. The user can press the depicted number and control buttons. Vice versa, information is presented through combinations of spoken messages and written text and minimalist graphics which are displayed in an area that simulates the tiny LCD display of a real phone.

Consider the situation in which a group of three geographically dispersed users U1, U2, and U3 team up in a virtual meeting space to discuss the details of an impending trip to downtown Saarbrücken. U1 accesses the virtual meeting space via his PC, U2 via a PDA, and U3 via a phone with a tiny LCD display. Figure 1 provides an example that illustrates how the system supports the users in localization and route planning tasks. In order to clarify how to get from a certain location to another, the participants want to consult a map representation. U1 is now in an advantageous position as his PC can easily display even highly colored and detailed maps. But what about the two other communication partners? As far as usability is concerned, it doesn't make much sense to output a

[*] Magic Lounge is funded under the Esprit (now FET) Long-Term Research pro-active initiative i3. The project is one of 13 projects within the i3 initiative. Project partners are DFKI, Saarbr• cken Germany; NIS, Odense University, Denmark; LIMSI-CNRS Paris France; Siemens AG, München Germany; and The Danish Isles - User Community, Denmark.

complex graphics on a small PDA screen. Therefore, U2 receives a more abstract graphical presentation, in this case a network-style map that essentially encodes topological information about the objects in focus. The phone user U3 is certainly in the weakest position as there is only a 100x60 pixels display available on his phone to output graphics. In order to keep all communication partners in the loop, we need a service that provides each partner with an individual view on the underlying information. Taking into account the available communication capabilities of the communication devices the phone user U3 receives on his display a simplified graphical representation of the street which is in the current focus of interest. Other details, such as the street name or attributes which can not be shown, are provided verbally using a text-to-speech synthesizer on the server side.

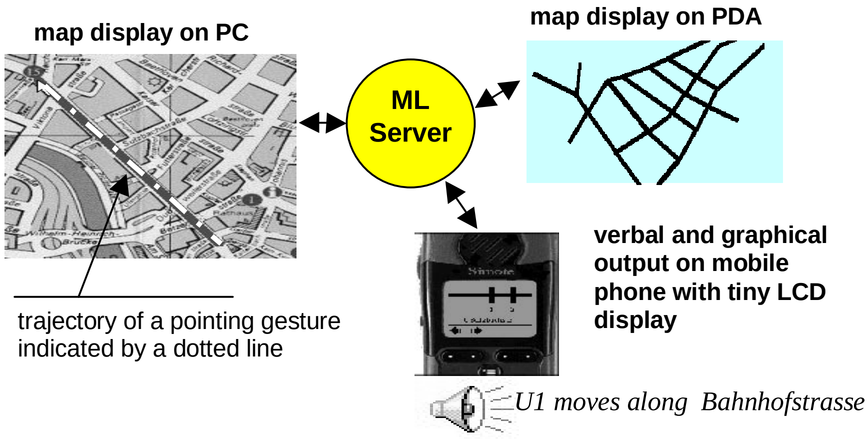

Fig. 1: The central server component provides different views on a shared source of topological data.

2.2 APPROACH AND TECHNICAL CONCEPTION

2.2.1. Generation different views from a common hierarchical data representation

Our approach to provide the users with appropriate information displays is to generate presentation variants (i.e. different views, see Figure 2a) from a common formalized representation of the information to be presented. We use a data model in which data are described in hierarchically ordered layers. Of course the details of this hierarchical representation depend on the data which have to be represented. Currently, this requires some additional efforts for the administrator of the server. In the future, however, we hope that graphical information with semantic annotations will become more common in the world-wide-web so that suitable transformations can be performed automatically. The left part of Figure 2b shows such a data description for the “shared” map application as illustrated in Figure 1. The uppermost level of this hierarchy is just a list of meaningful objects, such as the names of streets, buildings, places, bridges and the like. In contrast to that, the lowest level of the hierarchy comprises a detailed bitmap, but also all the information of the superior layers. The intermediate layers represent abstractions with respect to the information of the layers below. In the map example the intermediate layers may correspond to representations which subsequently abstract from geometric aspects such as shape, orientation, and distance between objects. As shown in the right part of Figure 2b, this hierarchical representation may be used to generate different views on the data. It also indicates a relation between the media which may be used to encode the information available at a certain level. For example, if only the medium text is available, a view may just consist of a street name while a detailed view would require a multimedia PC.

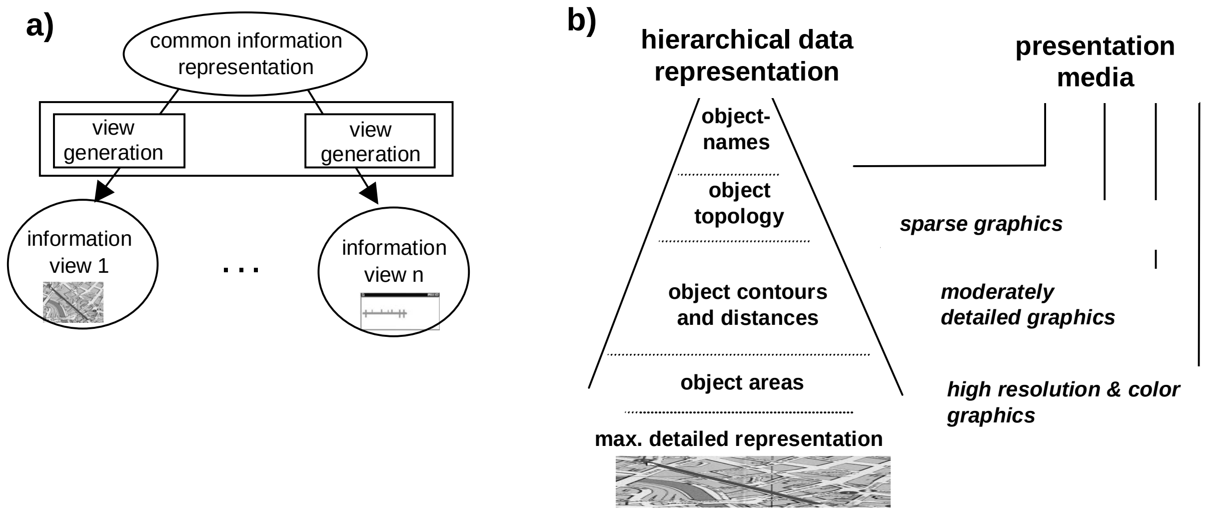

Fig. 2: a) generation of views on a centralized information source
b) hierarchical data representation (left part) which may be used to generate different views on the data. The more powerful graphics is available, the more detailed views may be provided (right part).

2.2.2. Mirroring user interactions on different views

In addition to the generation of different views on geographical data, the data hierarchy plays also a major role for mapping or mirroring of markings and pointing gestures from a source view (e.g., on a PDA) to another target view (e.g., on a PC). The approach for mapping or mirroring markings and pointing gestures from a source view (on a PDA) to another target view (on a PC) is illustrated in Figure 3. The mapping involves the following steps:

- Determine the kind of user interaction that is performed on the source view. In the example of Figure 3, the user has performed a *Mark* action on his PDA display
- Determine the object(s) which are affected by the user interaction. In the example, the user has marked a graphical element. which encodes a certain street.
- Resolve the object reference, i.e., determine the corresponding domain object in the hierarchical data representation. In the example, the object in question is *street#4*
- Look up which action on the PC display corresponds to the action that has been executed on the PDA display. In the example, this action is named *Mark_PC*.
- Determine the graphical representation of the object (*street#4)* on the PC display.
- On the PC display apply the action (*Mark_PC*) to the determined graphical object representation.

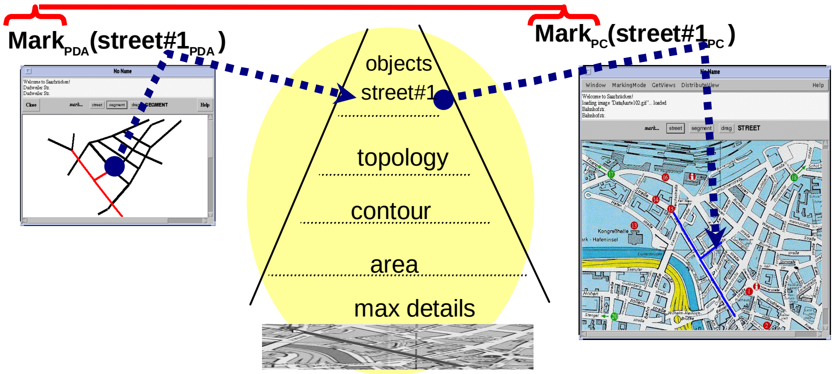

Fig. 3: An object marking on the PDA display is mirrored on the PC display. References to objects are resolved over the hierarchical data representation at the ML Server.

An apparent problem in route planning tasks is the fact that more complex markings on a 2D-map display cannot be mirrored directly on the phone display. One possible approach is to transform the spatial display of the marking into a sequence of phone displays, each showing a yet displayable segment of it. Figure 4 illustrates this approach. The PC user has marked a path on his/her PC display. To view the marked path on the phone display, the centralized server (ML Server) determines the affected street segments converts and generates a sequence of displays. Using the scroll buttons on the phone, the user can navigate back and forth through the sequence of the displays. While this approach does not allow to grasp the marked path at one glance, it may still help the user to build up a mental representation of the path (e.g. how many turns to make, order of turns etc.).

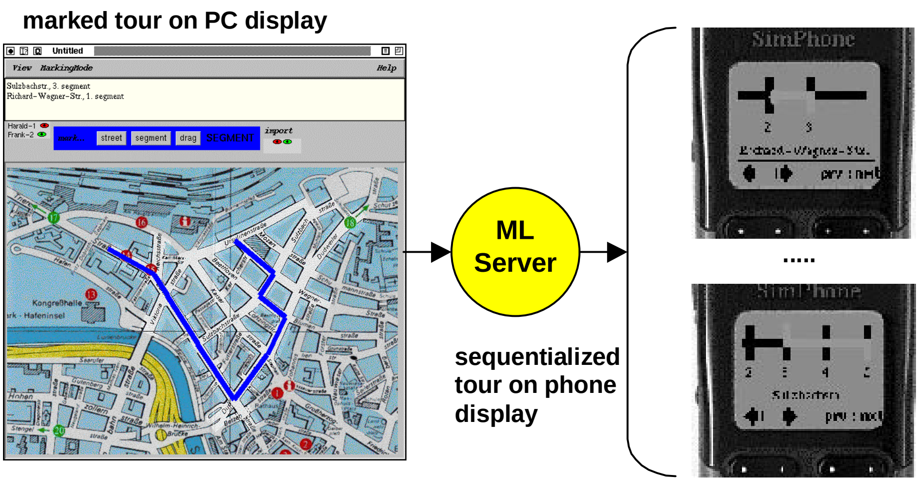

Fig. 4: A marked tour on the PC display is translated into a sequence of displays for the (simulated) phone viewer. To learn about the marked tour, the phone user navigates through the single segments of the tour.

2.3 IMPLEMENTATION OF THE MAPVIEWS TEST-BED

In order to get some first impressions on the feasibility of collaborations over heterogeneous graphical representations, we implemented a test-bed using Java as programming language and CORBA for the exchange of data and commands between the distributed components. In its current version, this test-bed runs on networked PCs and consists of a centralized monitoring module which holds the hierarchical representation of the domain data, and to which one or several instantiations of the following components are connected:

- a viewer for the display of a high resolution map on a PC
- a viewer for the display of moderately complex graphics on a PDA,
- a viewer for minimalist graphics which is combined with a map verbaliser for information displays on a mobile phone with a tiny LCD display. For the purpose of the prototype, however, we simulate the phone by an interactive GUI on a PC. The GUI consists of a realistic image of a mobile phone. The user can press the depicted number and control buttons. Vice versa, information is presented through combinations of spoken messages and written text and minimalist graphics which are displayed in an area that simulates the tiny LCD display of a real phone.

The monitoring module has access to the underlying hierarchical data representation. Taking into account the presentation capabilities of the viewer components, the monitoring component forwards portions of the data to them. For example, for the display of a road net, the diagram viewer will only receive a polygonal representation of the roads, but not a bitmap. A further task of the monitoring component is the synchronization of user interactions. In the current version of the demonstrator, a user of a viewer (map or diagram) can perform marking gestures on displayed road segments. For example, if the PC user moves the mouse cursor along a road segment while pressing a mouse button, the segment will be highlighted on his display. However, the marking action will be recognized by the monitoring component which will in turn inform the other components (PDA viewer and map verbalizer) that a marking occurred on a certain segment. Vice versa, the PDA user may perform a marking action which will lead to a change on the map display on the PC and a verbal notification on the map verbalizer.

2.4 EVALUATION OF MAPVIEWS

So far, we only conducted some informal usability tests with the demonstrator. Based on these observations, we got a clearer idea on the sort of appropriate information displays and the type of collaboration modes which are likely to support the users in solving their tasks. For example, it tuned out that collaborations without any verbal (or textual) communication channel does not work. On the other hand, the availability of graphical representations – though being different from each other – help to facilitate collaborations on localization and route planning tasks. Another observation concerns the way how the users exchange markings among each others. It became apparent that one should prefer a collaboration principle which leaves the decisions to the individual users when and from where they want to "import" other views. Also, it seems not advisable to import markings from several other users at the same time since this can result in confusing displays. Also, to increase and improve the functionality of the demonstrator we need to equip the different viewer components with additional interaction facilities, for example, in order to enable zooming and scrolling of displays.

2.5 DEMONSTRATOR SETUP FOR MAPVIEWS

Since in MapViews mobile devices are simulated only by specially designed GUI's on a PC screen, there are no special requirements for demonstrating the system.

3. MAGIC LOUNGE

The name "Magic Lounge" stands for a virtual meeting space in which the members of geographically dispersed communities can come together, chat with each other and carry out joint, goal-directed activities, such as joint travel planning illustrated in the previous section. The Magic Lounge prototype aims at demonstrating a number of innovative communication services for such virtual meeting spaces, including:

- *Access through heterogeneous devices.* Unlike many other approaches (e.g., chat corners on the web [1] or CSCW platforms [2]) we do not assume that the users enter the virtual space via exactly the same communication channels. Rather, we imagine a scenario as illustrated in Figure 5, where some users are equipped with fully-fledged standard multimedia PCs, but others with handheld PDA's or mobile phones that apart - from the audio channel – come with tiny LCD displays for displaying text and minimalistic graphics.

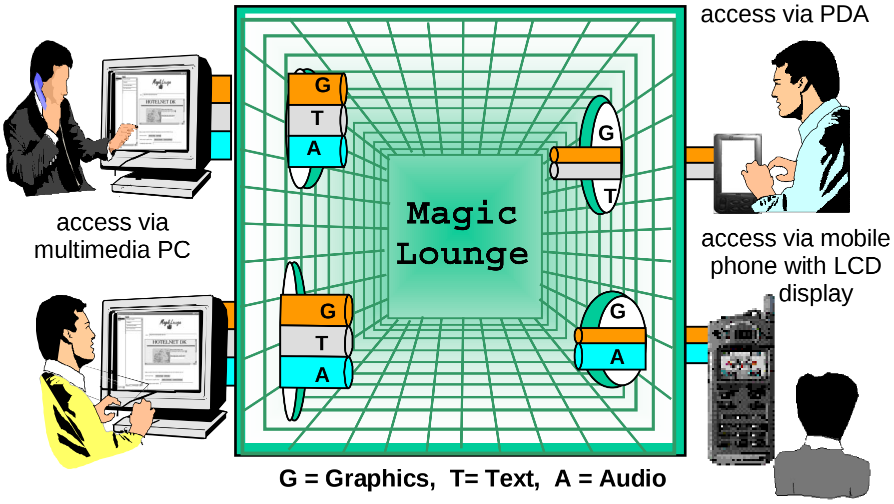

Fig. 5: Accessing the Magic Lounge virtual meeting space via heterogeneous communication devices

- *Memory functions to support conversations in virtual meeting space.* A survey on current teleconferencing and groupware systems as well as project-specific participatory design work [3] with ordinary users revealed a strong need for a structured memory that can be queried by newcomers or latecomers who want to know what has happened in a meeting so far. In existing systems, such a memory is limited to a raw unstructured audio or textual file. The conception of the Magic Lounge memory component [4] foresees the recording of spoken and typed utterances as well as other interaction events, such as the mutual exchange of references to electronic documents, which may be part of the virtual meeting environment. Contributions are recorded in the memory together with structuring information such as the intended and declared communicative act or the conversation to which this contribution belongs. For example, a 'temporal meeting browser" allows users to navigate back and forth through recorded meetings and to inspect individual contributions in a non-linear manner [5]. The conception also comprises a set of different user interfaces for accessing the memory content from specific points of view and by means of different communication devices.

- *Mixed audio/chat communication* It is yet an ongoing discussion whether text-based communication (such as chat or short messaging) will become less important with better services for audio conferencing via the Internet or commercial mobile networks. In Magic Lounge, we provide both communication channels. However, all contributions to a conversation are treated as objects that are recorded in the memory and may be retrieved according to several sorting and structuring criteria, such as temporal occurrence, producer, or relevance to a certain topic.

3.1.1 CONNECTING MOBILE COMMUNICATION DEVICES TO THE MAGIC LOUNGE

We now turn to the question how a real mobile device can be used as a physical interface to an interactive application (for a single user or a group of users) that runs on a server. The approach taken in the context of the Magic Lounge project is illustrated in Figure 6. We assume that the server is connected to a network that runs TCP/IP for the exchange of data between the server and its application clients. In case that a client has direct access to this network and is also computationally powerful enough to run all the required system components (e.g., written in Java and connected with other system components via CORBA [6]), no additional efforts are required. In general, however, such an assumption cannot be made for mobile clients. Therefore, our system architecture foresees a so-called gateway component for mobile clients, such as PDA's and mobile phones. Essentially the idea is to split the application interface into two parts. The gateway component is responsible for mapping output (received from the application) into a format that can be sent to a mobile device and presented on it. Vice versa, the gateway receives input data from the mobile device, maps the data onto application-specific input formats and forwards them to the application. This approach allows to reduce the complexity of the remaining interface components which will run on the mobile devices.

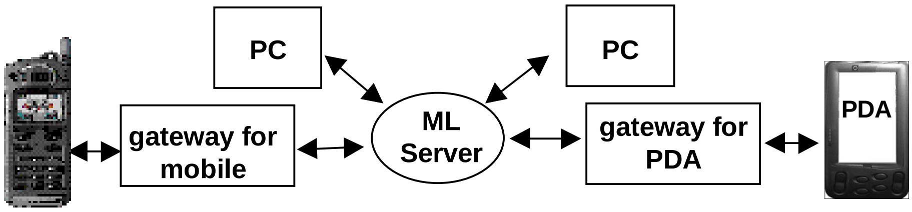

Fig. 6: Mobile devices are connected to the central server (ML Server) via gateway components.

While the demonstrator described in Section 2 was based on a home-made simulation test-bed to emulate the mobile devices, we recently switched to a professional simulation environment [7]. This environment relies on WAP (Wireless Application Protocol [8]) which appears as one of the most promising de facto standards for wireless access to the World-Wide Web. Adopting WAP has the advantage, that we can easily switch to real devices (currently we use a NOKIA 7110 WAP phone for testing). On the other hand, the current WAP version is quite limited with regard to graphics. As in the early days of the WWW, bitmaps of all graphics must be available on a WAP server which will send them on request to the mobile devices. Furthermore, using a WAP browser for the realisation of an application interface imposes the need for an additional module that can manage incoming requests from many phone users simultaneously. The architecture shown in Figure 7 refines the one shown in Figure 6 by adding a so-called WAP phone servlet and some third party components (drawn as grey boxes). The third party components enable internet access through mobiles. In particular, the WAP gateway is a server component (e.g., available from Phone.com or Nokia) forwards pages from an arbitrary web-server to a telecom provider for mobiles. The pages must be written in WML (wireless mark-up language). The WAP phone servlet is part of our own system development. It maintains a table of all connected users and ensures the correct message passing between a particular device and its assigned phone gateway which in turn interfaces with the application.

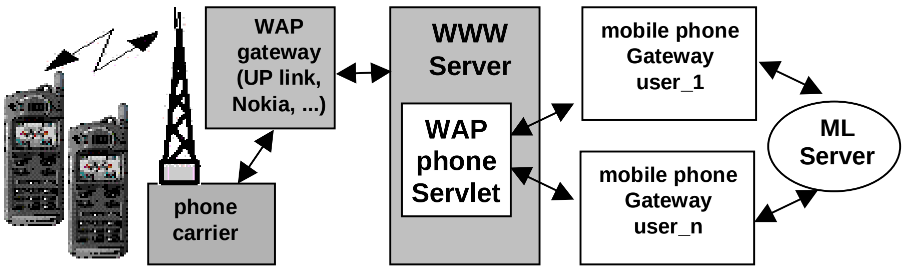

Fig. 7: Refinement of the architecture when relying on WAP. Each mobile is connected the application (running on the ML Server) via a mobile phone gateway A WAP-phone servlet ensures the proper assignments of the devices with their phone gateways.

3.2 AUTOMATED TAILORING OF INFORMATION PRESENTATIONS TO THE CAPABILITIES OF MOBILE DEVICES

The MapViews system presented in Section 2 already illustrated the need for adapting information presentations to different target devices. Currently, many attempts are made to develop transformation mechanisms that take as input arbitrary information sources (e.g. html pages) and deliver information presentations that can be displayed on mobile devices with limited display capabilities. In case of a textual source, a straightforward approach is to fragment the text into displayable chunks. However, whether or not such presentations are still useful and acceptable is another question. Moreover, in the case of visual media such as graphics, animation and video, such a partitioning is often not possible at all. During the last decade, a number of AI researchers worked towards the development of mechanisms which automatically design presentations which are tailored to specific user needs in specific situations (cf. [9]). It is quite natural to wonder whether such approaches can be used in order to flexibly generate presentation variants that take into account the specific display capabilities of the emerging variety of mobile devices. Typically, such approaches take as input an application-specific knowledge- or database together with a presentation goal that refers to a particular portion of the knowledge or a particular data set. Taking into account parameters such as a user profiles and resource limitations (e.g. screen size, available media, and so forth) one and the same presentation goal can lead to very different generation results. Central to many generation approaches are the principles of recursive decomposition and specialisation. That is, a presentation goal is decomposed into less complex goals until a goal can be solved by the production of a media object which will become part of the final overall presentation. In our current work, we rely on our previously developed component for planning multimedia presentations [10, 11]. To use this component for new target devices like a PDA or a mobile phone, we need:

- to identify suitable presentation types (e.g. small diagram types, list-style enumeration etc.);
- to identify the elementary units as well as the composition rules of these presentation types;
- to define design strategies which represent the composition rules and which will be used as operators of the automated presentation planner;
- to define generators that can produce the elementary media objects.

To illustrate this generation approach, consider the task of providing access to a database which stores spoken and written utterances from all conversation partners in a meeting. Especially latecomers are often interested in getting an overview on what has been said earlier in the meeting. In order to access the contents of the database, the repertoire of design strategies comprises strategies for the design of a table as it can be displayed on a PC screen as well as strategies that design a presentation for a mobile phone (cf. Figure 8). Given a presentation task (here the request to present temporally ordered database entries), the presentation planner selects only those design strategies that are compatible with the provided profile information of the target device.

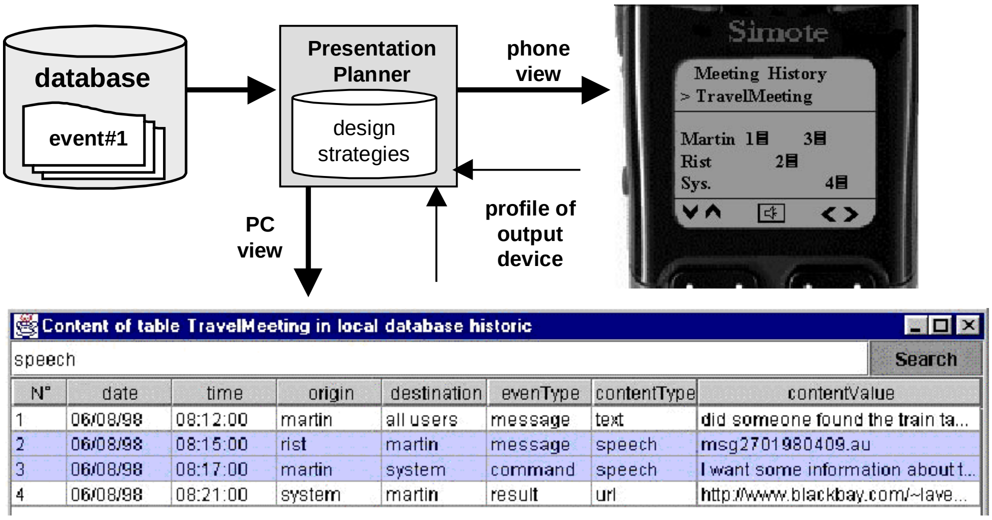

Fig. 8: A knowledge-based presentation planner generates information presentations which are adapted to the specific capabilities and resource limitations of the target devices.

3.3 DEMONSTRATOR SETUP FOR THE MAGIC LOUNGE

A full demonstration of the Magic Lounge system requires tat least two PCs (with full douplex sound boards) and a mobile device with a build-in WAP browser (mobile phone or Palm computer). One of the PCs need to run the Magic Lounge server and should be powerful enough to run a Magic Lounge PC client as well. At a recent project audit, the system has been demonstrated using a Pentium II, 400MHZ and 128K cache as a server, two Labtop computers (Pentium 200) as PC clients, a NOKIA 7110 mobile phone, and a HP Jornada 680 Pocket Computer.

4. CALL-KIOSK

The development of Call-Kiosk aims at a commercially exploitable information service for mobile users. Essentially the idea is similar to that of an information desk at which an information seeking user gets advice. For example, consider the situation of a tourist who arrives in Saarbrücken for the first time. In order to get some basic information on hotels, restaurants, shops, and interesting spots, it is quite natural to go to the tourist office. Since it would be hard to remember all suggestions made by the tourist guide, people usually take away a brochure or a city map often annotated manually to highlight aspects that are of particular relevance to the individual tourist. The Call-Kiosk is designed to serve mobile clients who want to receive similar information via a mobile phone with a WAP-browser. Call-Kiosk is a semi-automated information system. When seeking information, the client dials a certain service number (e.g. the number of the tourist office) to get connected to a human operator (e.g. a professional tourist guide). After having verbally specified the information need, the human operator just selects relevant information bits on a display window. In case of travel information, the operator may see a certain map on his/her display and select hotels, shops, etc. by clicking on them with the mouse pointer. When the selection process is finished, the operator starts an automated presentation generator that delivers a series of WML pages as output. These individualized pages are then stored on a server that can be accessed by the client. The generated series of WML pages serve as substitutes for hardcopy take-away brochures or maps that carry individual notes.

In contrast to MapViews or Magic Lounge, Call-Kiosk is restricted to a two-point conversation. One of the partner (the service provider) is assumed to have a fully-fledged workstation with all necessary support to efficiently find and select information relevant to a customer's inquiry. The customers, on the other hand, may use an arbitrary mobile device that has a built-in WAP-browser. At the time of writing this paper, the implementation of the Call-Kiosk has not yet been finished. However, a demonstrable version will be available by January 2000. A demonstrator setup will require a PC and a WAP-enabled mobile phone only.

5. COMPARISON TO RELATED WORK

A central aspect of all of the three presented systems is the provision of different views on a shared information store. The concept of "views" on internal data representations has been introduced a long time ago in database research to accommodate for the fact that different users may have different information needs and access privileges. In CSCW research, Stefik et al. [12, 13] have coined the term "relaxed WYSIWIS (What You See Is What I See)" in the context of a collaborative system with shared graphical output. In contrast to "strict WYSIWIS, where all collaboration partners obtain exactly the same graphical displays, relaxed WYSIWIS allows to be more selective on what is to be shared by other collaboration partners and how shared information gets individually presented. Later on, several different approaches have been suggested to support flexible (also referred to as tailorable or customizable) views either for particular CSCW systems or as parts of toolkits for the development of multi-user applications. For instance, the Rendezvous system [14] relies on the so-called abstraction-link-view [15] in which a the possibly different views are linked to a shared data abstraction of the application semantics via constraints. During a multi-user session the different views are dynamically updated in a way so that the specified constraints remain satisfied. The toolkit Weasel [16] comprises a so-called Relational View Language for specifying customized views and their dynamic behaviors during a session. The CSCW toolkit GroupKit [17] relies on an open protocol approach that allows application programmers to implement and coordinate tailored views on shared data abstractions. The notion of so-called tailorable user display agents has been introduced by Bentley et al. [18]. Sitting between the shared data representation and the display of a user, the agent is responsible for the (possibly) user-specific style of information display and also for the continuos update of the display during a session based on a filed mechanism for event-handling. There is no doubt that the use of such toolkits will release the programmer of a CSCW application from low-level socket-programming and event-handling issues. The reason why we did not base our own developments on any available toolkit is that we are especially interested in collaborations that involve very different communication terminals including mobile phones. Assumptions such as the availability of a common operating or windowing system could not be made. Abstracting from implementation details, however, similarities with other systems become more apparent. For example, similar to the Rendezvous system view generation in our systems is also done at a centralized server component and we, too, rely on an event-driven approach for coordinating the information content in different views. On the other hand, a unique feature of our approach is that we apply automated presentation planning for the generation of a certain view. Our main motivation for this is that the distinctive heterogeneity of our communication terminals imposes a much higher need for presentation flexibility. So far, the involvement of heterogeneous communication devices has been rarely addressed in CSCW research or in research on mobile computing. For example, Myers et al. [19] describe a collaborative environment in which PDA's are used as remote control devices for the manipulation of a display that is shared by all users. A combination of multiple PDA's with a shared large-scale whiteboard is also used by Greenberg and Boyle [20]. In their SharedNotes system, the PDA's are used as input devices for private notes but which can also be made public by sending them to the shared whiteboard. While they have focused on an the distinction between private and public information in a multi-user application, our work addresses the issue of tailored information presentations in a collaboration environment with heterogeneous devices. For applications like accessing a meeting memory (Magic Lounge) or presenting a selected portion of an information store (Call-Kiosk) we generate presentations through a goal driven refinement process that takes as additional generation parameters the device-specific output capabilities/limitations.

Acknowledgements

We would like to thank Matthias Hüther for his implementation work on the MapViews system.

References

[1] Dourish, P. (1998) *Introduction: The state of play. Special Issue on Interaction and Collaboration in MUDs.* Computer Supported Cooperative Work: The Journal of Collaborative Computing 7(1-2), pp. 1-7. 1998

[2] Coleman, D., Khanna, R., (Eds.) (1995) *Groupware* Technology and Applications. Upper Saddle River, NJ: Prentice Hall.

[3] Masoodian, M., and Cleal, B., User-Centred Design of a Virtual Meeting Environment for Ordinary People. Conference Proceedings of HCI International'99, 8th International Conference on Human-Computer Interaction, Vol. 2, 528-532, Munich, Germany, August, 1999

[4] Rist, T., Martin, J.-C., Néel, F.D., and Vapillon, J.: On the Design of Intelligent Memory Functions for Virtual Meeting Places: Examining Potential Benefits and Requirements. To appear in the Journal Le Travail Humain.

[5] Luz, S. F. and Roy, D. M.: Meeting browser: A system for visualising and accessing audio in multicast meetings. In Proceedings of the International Workshop on Multimedia Signal Processing. IEEE Signal Processing Society, September 1999.

[6] Object Management Group. *The Common Object Request Broker Architecture (CORBA).* Framingham, MA, U.S.A. Online resource: http://www.omg.org/corba/

[7] Unwired Planet Inc.: UP.SDK Software Development Kit. Software and documentation available under: http://www.uplanet.com

[8] Wireless Application Protocol Forum Ltd.: The Wireless Application Protocol. Specification available under: http://www.wapforum.org/

[9] Rist, T., Faconti, G., Wilson, M. (Eds.) (1997) Intelligent Multimedia Presentation Systems. Special Issue of the International Journal on the Development and Application of Standards for Computers, Data Communications and Interfaces. Volume 18, Number 6 and 7.

[10] André, E., Rist, T. (1995) Generating Coherent Presentations Employing Textual and Visual Material, in: AI Review 9, pp. 147-165.

[11] André, E., Müller, J., Rist, T.: WIP/PPP: Knowledge-Based Methods for Fully Automated Multimedia Authoring. In: EUROMEDIA'96, London, UK, pp. 95-102, 1996.

[12] Stefik, M., Bobrow, D.G., Foster, G., Lanning, S., Tatar, D. (1987) Beyond the Chalkboard: Computer Support for Collaboration and Problem Solving in Meetings ACM Communications (30). 32-47.

[13] Stefik, M., Bobrow, D.G., Foster, G., Lanning, S., Tatar, D. (1987) WYSIWIS Revisited: Early Experiences With Multiuser Interfaces, ACM Trans. on Office Information Systems 5, 2, pp. 147-167.

[14] Hill, R.D., Brinck, T., Rohall, S.L., Patterson, J.F., Wilner, W. T. (1994) The Rendezvous Architecture and Language for Constructing Multiuser Applications. ACM Transactions on Computer-Human Interaction, Vol. 1, No. 2, June 1994. p81-125.

[15] Hill, R.D., (1992) The Abstraction-link View Paradigm, Using Constraints to Connect user interfaces to applications, Proc. of CHI '92 (Monterey, CA, May) pp. 335-342.

[16] Graham, T.C.N., Urnes, T. (1992) Relational views as a model for automatic distributed implementation of multi-user applications. In Proc. of CSCW'92, pp. 59-66.

[17] Roseman, M., Greenberg, S. (1996). Building Real-Time Groupware with GroupKit, A Groupware Toolkit. ACM Trans. on Human-Computer Interaction, pp. 66-106.

[18] Bentley, R., Rodden, T., Sawyer, P., Sommerville, I. (1994) Architectural support for cooperative multiuser interfaces IEEE Computer, pp. 37-46.

[19] Myers, B.A. Stiel, H., Gargiulo, R. (1998) Collaboration Using Multiple PDAs Connected to a PC. In Proc. of CSCW'98, pp. 285-295.

[20] Greenberg, S., Boyle, M. and LaBerge, J. (1999). PDAs and Shared Public Displays: Making Personal Information Public, and Public Information Personal. Personal Technologies, Vol.3, No.1, March. Elsevier.