Using mobile Communication Devices to Access Virtual Meeting Places

Thomas Rist
DFKI, Saarbrücken, Germany

Extended Abstract

1. Working background

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 project Magic Lounge* we are developing a number of new communication services which are to overcome communication barriers imposed by the use of heterogeneous communication terminals. The name "Magic Lounge" also stands for a virtual meeting place in which the members of a geographically dispersed communities can come together, chat with each other and carry out joint, goal-directed activities. For example, to plan a travel they may jointly explore information sources on the WWW, such as time tables of flights and public transportation services. 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 place via exactly the same communication channels. Rather, we imagine a scenario as illustrated in Fig. 1, where two users are equipped with fully-fledged standard multimedia PCs, another with a handheld PDA that allows the display of text and graphics on a small medium resolution screen, while the fourth user is connected through a mobile phone that - apart from the audio channel - has a tiny LCD display which can be deployed for displaying minimalistic graphics.

This contribution addresses issues that arise when trying to integrate mobile devices with limited display and interaction capabilities into a platform tele-conferencing and collaborative work. We first report on a recently conducted case study in which we used a simulation environment to investigate how a small group of geographically dispersed users can jointly solve localization and route planning tasks when being equipped with different communication devices (as illustrated in Fig. 1). We then present a system architecture that includes gateway components as an interface between the mobile device and an interactive application that runs elsewhere but is accessible via the internet. Finally, we argue that there is a 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.

* 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, Saarbrűcken Germany; NIS, Odense University, Denmark; LIMSI-CNRS Paris France; Siemens AG, München Germany; and The Danish Isles - User Community, Denmark.

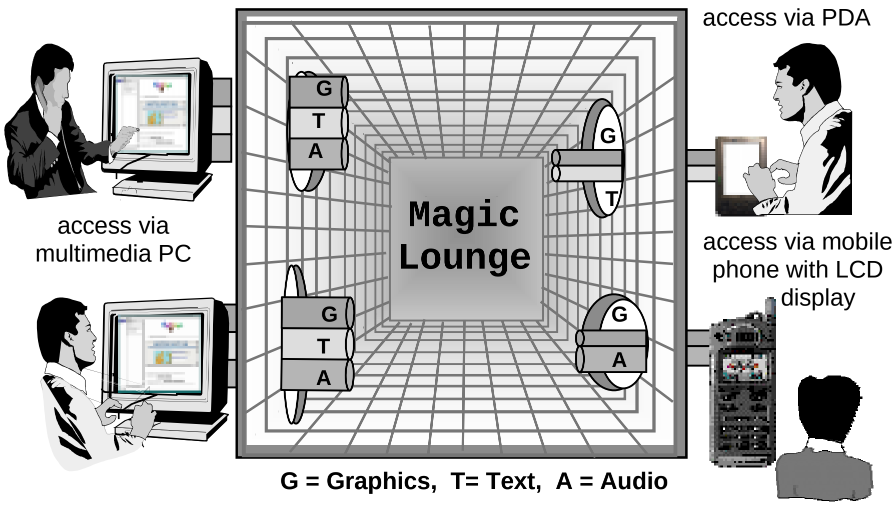

Fig. 1: Accessing the Magic Lounge virtual meeting place via heterogeneous communication devices

2. Solving collaboration tasks over heterogeneous communication channels

To what extend 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 users U1,U2, and U3 meet in the Magic Lounge to discuss the details of an impending trip to downtown Saarbrücken. U1 accesses the virtual meeting place via his PC, U2 via a PDA, and U3 via a phone with a tiny LCD display. An example, how the system supports the users in localization and route planning tasks is illustrated in Fig. 2. 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 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.

Our approach to provide the users with appropriate information displays is to generate presentation variants (i.e. different views) 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. We assume that this representation will be located on a centralized server (the ML Server in Fig.2) which is accessible by all participants. 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.

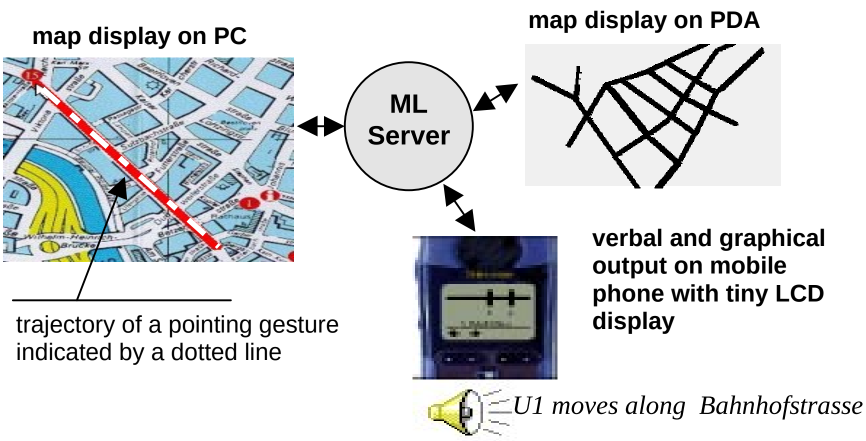

Fig. 2: The central server component provides different views on a shared source of topological data.

The data hierarchy for the "shared map" application is shown in the center of Fig. 3. The uppermost level of this hierarchy is just a list of meaningful domain 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 pixmap, but also all the information of the superior layers. The intermediate layers represent abstractions relative to the information of the layers below. In the map example the intermediate layers correspond to representations that subsequently abstract from geometric aspects such as orientation, shape, and distance between objects. 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).

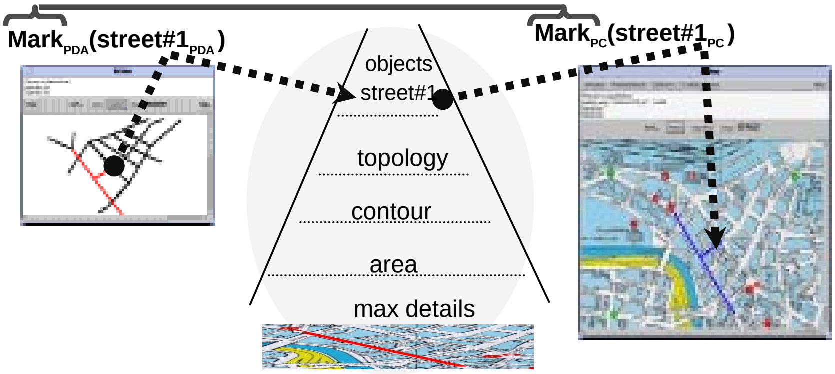

Fig. 3: An object marking on the PDA display is mirrored on the PC display. References to objects are resolved over the hierarchical data representation at the ML Server.

As illustrated in Fig. 3., the mapping involves the following steps: First, the system determines the interaction type that is performed on the source view. In the example of Fig. 3 the user has performed a *Mark*-action on his PDA. Next, the system determines the object(s) which are affected by the interaction (in the example, the graphical representation of street#1 has been marked). The system also determines which action on the PC side corresponds best to the action that has been executed on the PDA display (in the example, the appropriate action is named *Mark_PC*). Finally, the system determines the graphical representation of street#1 on the PC display and applies the action *Mark_PC* to it.

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 5. 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 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.).

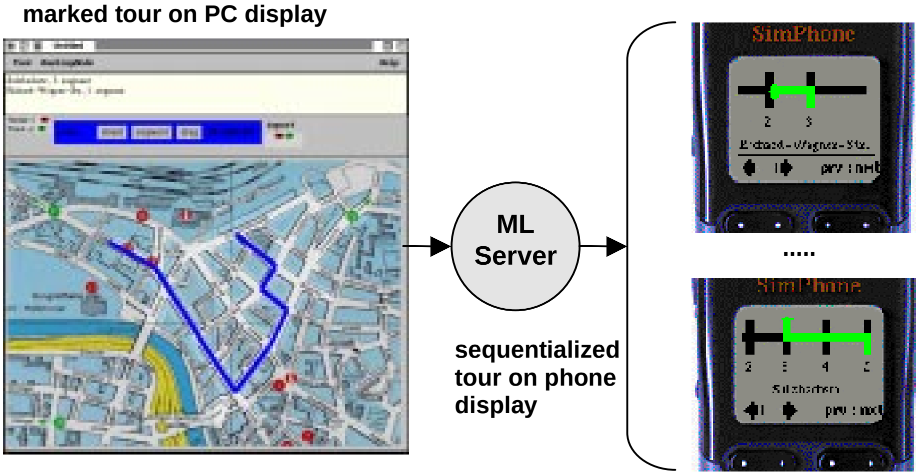

Fig. 4: A marked tour on the PC display is translated into a sequence of displays for the (simulated) phone viewer. To learn about the marked tour, the phone user navigates through the single segments of the tour.

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.

3. Interfacing applications with mobile communication devices

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 Fig.2. We 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 [3]), 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.

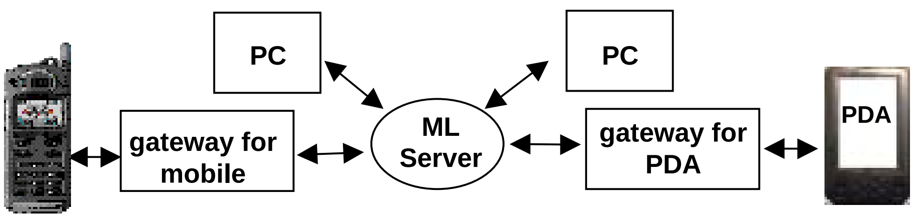

Fig. 5: Mobile devices are connected to the central server (ML Server) via gateway components.

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 [4]. This environment relies on WAP (Wireless Application Protocol [5]) 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 as soon as they become broadly available on the market. 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 Fig. 3 refines the one shown in Fig.2. by adding a so-called WAP phone servlet as well as third party components (drawn as grey boxes). The third party components enable internet access through mobiles. The WAP gateway is a server component that 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 our own 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.

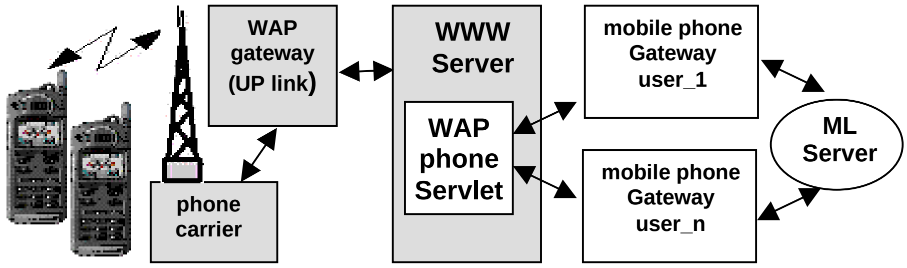

Fig. 6: Refinement of the architecture when relying on WAP. Each mobile is connected the application (running on the ML Server) via a mobile phone gateway A WAP-phone servlet ensures the proper assignments of the devices with their phone gateways.

4. Automated tailoring of information presentations to mobile devices

The case study 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. [6]). 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 the planning of multimedia presentations [7]. 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 enumerations 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. Fig. 7). 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.

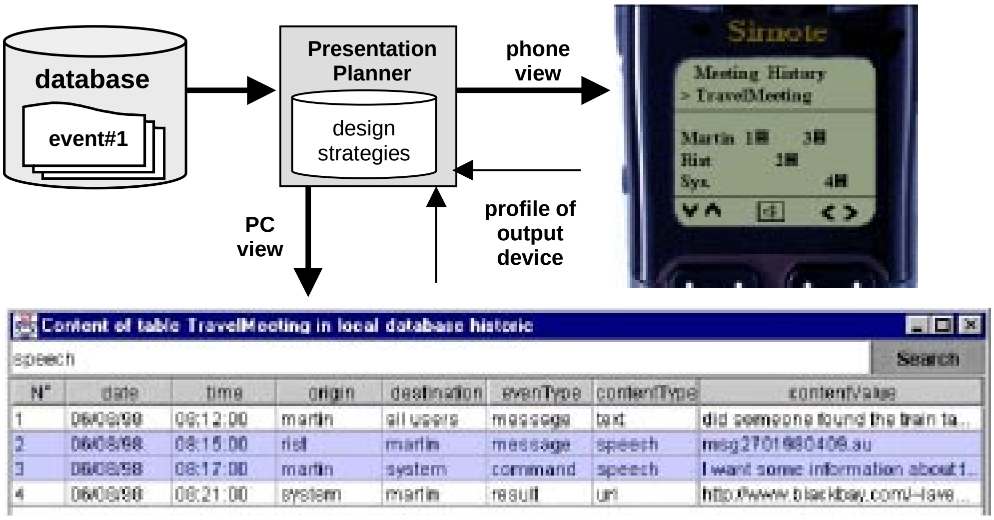

Fig. 7: A knowledge-based presentation planner generates information presentations which are adapted to the specific capabilities and resource limitations of the target devices.

References

[1] Dourish P.: *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] Yong C., Buchanan C.: *Evaluation of Collaborative Classroom Tools: Groupware*. CSIS Department, Kennesaw State College. 1996. Report available under: http://csis3.kennesaw.edu/~groupwre/index.html

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

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

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

[6] Rist T., Faconti G. and Wilson M. (eds): 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. 1997

[7] 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.