Getting the Mobile Users in: Three Systems that Support Collaboration in an Environment with Heterogeneous Communication Devices

Thomas Rist
rist@dfki.de

Patrick Brandmeier
brandmeier@dfki.de

Gerd Herzog
herzog@dfki.de

Elisabeth André
andre@dfki.de

German Research Center for Artificial Intelligence (DFKI GmbH)
Stuhlsatzenhausweg 3, D-66123 Saarbrücken, Germany

ABSTRACT

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 is 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.

Keywords

Mobile communication, collaborative systems, multimedia

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 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 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 devices. After that, we present the Magic Lounge[1] 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.

2. MapViews

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 this paper, 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. The task of MapViews is to support the users in localization and route planning tasks. In order to clarify how to get from a certain location to another, the participants might 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. He 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 cannot be shown, can be provided verbally using a text-to-speech synthesizer on the server side.

2.1 Generating Different Views

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. Our approach to provide the users with appropriate information displays is to generate presentation variants (i.e. different views, see Fig. 1) from a common

[1] Magic Lounge is funded under the European Long-Term Research pro-active initiative i3. Project partners are DFKI, Saarbrücken Germany; NIS, Odense University, Denmark; LIMSI-CNRS Paris France; Siemens AG, München Germany; and The Danish Isles - User Community, Denmark.

formalized representation of the information to be presented. We use a data model in which data are described in hierarchically ordered layers. The details of this hierarchical representation depend on the data which have to be represented.

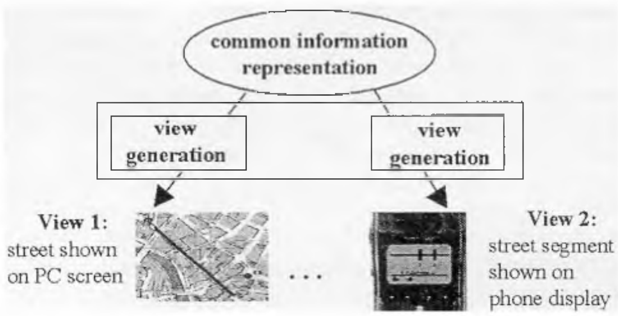

Fig.1: Presenting information through different views

Figure 2 shows such a data description for the "shared" map application as illustrated in the scenario above. 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 Figure 2 shows 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.

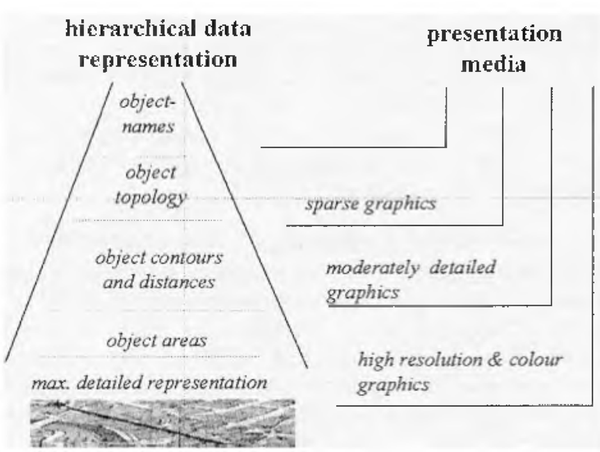

Fig.2: Correlation of data abstraction an presentation media

2.2 Mirroring Interactions on Different Views

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

1. Determine the kind of user interaction that is performed on the source view.
2. Determine the object(s) which are affected by the user interaction. E.g., a PDA user may have marked a graphical element which encodes a certain street.
3. Resolve the object reference, i.e., determine the corresponding domain object in the hierarchical data representation.
4. Look up which action on the PC display corresponds to the action that has been executed on the PDA display.
5. Determine the graphical representation of the object on the PC display and apply the marking operation that corresponds to the marking operation performed on the PDA display.

An obvious problem in collaborative 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. In MapViews a marked path on the PC's map display can be sequentialized into a series of small pages which can be shown on the phone display. 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.).

2.3 Implementation of MapViews

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 three different kinds of viewer components: 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, and a viewer for minimalist graphics which is combined with a map verbalizer for information displays on a mobile phone with a tiny LCD display. For the purpose of the prototype, however, we simulate the physical 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 a small window that simulates the tiny LCD display of a real phone.

2.4 Evaluation of MapViews

We only conducted some informal usability tests with the demonstrator. However, 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 turned 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. 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. 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:

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 [1] 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 [2] 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 [3]. The design of the memory component 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.

Access through heterogeneous devices. As with MapViews we imagine a scenario where some users are equipped with fully-fledged standard multimedia PCs while others use handheld PDA's or mobile phones that apart - from the audio channel – come with tiny LCD displays for restricted textual and graphical display.

3.1 Connecting via Mobile Devices

To use a mobile device as a physical interface to an interactive application (for a single user or a group of users) that runs on a server 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 a client has direct access to this network (e.g. by using a wave-LAN) 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 [4]), 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 (see Fig. 3). 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.

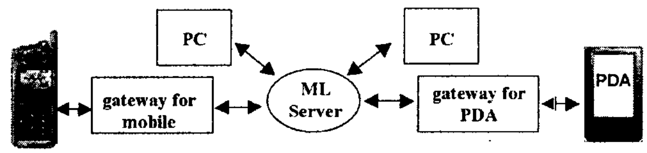

Fig. 3: Mobile devices are connected to the central server (ML Server) via gateway components.

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 WAP (Wireless Application Protocol [5]) for which professional simulation environments are available. Adopting WAP has the advantage, that we can easily switch between the simulation environment and the 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. For example, bitmaps of all graphics must be made available to a WAP server which will send them on request to the mobile devices. The pages must be written in WML (wireless mark-up language). 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. In our system this task is done by a so-called WAP phone servlet that 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.

3.2 Automated Tailoring of Presentations

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. [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. In our current work, we rely on our previously developed component for planning multimedia presentations [7, 8]. This component performs a recursive decomposition and specialisation of a given presentation goal into less complex units which eventually can be solved by the production of a certain media object which will become part of the resulting presentation. To use this component for new target devices like a PDA or a mobile phone, we need (1) to identify suitable presentation types (e.g. small diagram types, list-style enumeration etc.); (2) to identify the elementary units as well as the composition rules of these presentation types; (3) to define design strategies which represent the composition rules and which will be used as operators of the automated presentation planner; (4) 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 display on a mobile phone (see Fig. 4). 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. 4: Mobile phone display of a conversation history

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 the automated presentation generator that delivers a series of WML pages as output (see Fig. 5). 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 annotations.

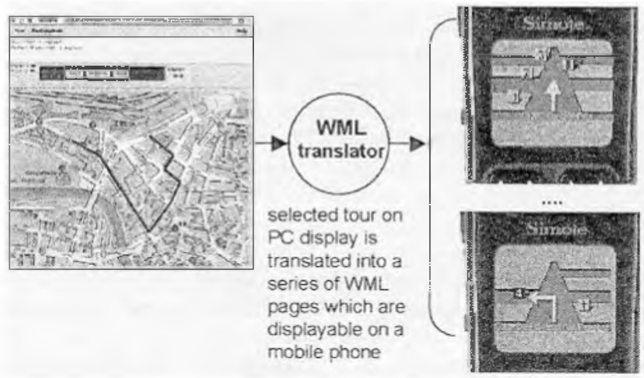

Fig. 5: Translation of a marked tour into a series of WML pages

In contrast to MapViews and 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, is assumed to use a mobile device that has a built-in WAP-browser, such as the Nokia 7110.

5. RELATED WORK

A central aspect of all of the three presented systems is the provision of different views on a shared information store. In CSCW research the concept of providing different views on shared information is also known by the term "relaxed WYSIWIS (What You See Is What I See) as introduced by Stefik et al. [9]. 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 [10] relies on the so-called abstraction-link-view in which 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 [11] comprises a so-called Relational View Language for specifying customized views and their dynamic behaviors during a session. The CSCW toolkit GroupKit [12] 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. [13]. 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. Toolkits like the ones mentioned here release the programmers of CSCW applications 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 also 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. [14] 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 [15]. 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 a 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.

6. REFERENCES

[1] 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 .

[2] 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.

[3] 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.

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

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

[6] 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.

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

[8] 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.

[9] 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.

[10] 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.

[11] 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.

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

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

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

[15] 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.