Using spatial Co-location for Coordination in Ubiquitous Computing Environments

Michael Beigl

Telecooperation Office (TecO), University of Karlsruhe, Vincenz-Prießnitz-Str. 1,
D-76131 Karlsruhe, Germany
michael@teco.edu

Abstract. A problem in Ubiquitous Computing environments is the coordination of a multitude of different devices. This paper presents the RAUM[1]-system that provides the basis for communication between devices in a Ubiquitous Computing environment. Such communication considers the spatial order of objects in the environment similar to the way humans do. The RAUM-system uses this order to establish communication connections between objects in the environment and gives applications the possibility to react according to the spatial order. This paper proposes that in Ubicomp environments such spatial-dependent location-aware communication is superior to other communication strategies. Also three example set-ups are presented indicating that applications for Ubicomp environments benefit from the RAUM-system in various ways.

1 Introduction

Ubiquitous Computing (Ubicomp) environments can be described as places populated with a myriad of different computerized and interconnected devices. One problem in such environments is the coordination of and the interaction with a multitude of devices. Coordination of objects and applications in Ubicomp depend on conditions of the physical world, especially of surrounding objects. Some applications use for example the range of an infrared or RF transmission device to restrict the spatial range they are responsible for. Here technical restrictions are used to assign a space of interest to an Ubicomp application.

The system presented here uses spatial layout as it is perceived by a human as a strategy for structuring and restricting the space of interest for an application. Human cognition attaches importance to spatial distribution [17], using spatial order of things as a help to carry out a task. For instance, a receiver placed on a telephone indicates that calls are possible (ready for call), papers in the waste-paper-basket indicate that they could be thrown away. Another example is people talking to each other: They stand in visual and audible reach to communicate. Communication and order of objects in an environment are bound to the spatial layout, binding behavior and

[1] Depending on the context the German word “Raum” could be translated as room, space, area or scope

relation of objects in the *virtual world* to the behavior of associated objects in the *physical world*. A transfer of physical relations into the virtual world is shown in Figure 1. If two objects are close together the spatial order has a meaning that is recognized in the same way both by the human and by the system (Fig.1 circle). This leads to the following thesis:

Thesis: The communication of (computerized) objects in Ubiquitous Computing environments should, like human cognition, be spatially organized.

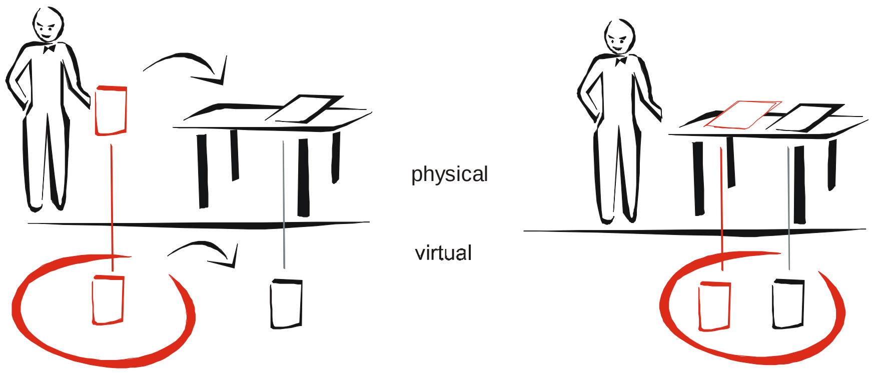

Fig. 1. Matching of physical and virtual world: If two objects (here two pieces of paper) come close in the physical world, the corresponding objects in the virtual world also come close. If they are close enough the left object recognizes the right object. They are close enough when the scope around the left object overlaps the right object.

In this paper I argue that in Ubicomp environments spatial-dependent location-aware communication is superior to other communication strategies when performing locally restricted actions. This claim is backed up and illustrated by three examples that show that applications for Ubicomp environments benefit from the proposed system.

This paper presents the RAUM system that handles communication between objects according to their spatial organisation in the environment. Therefore the RAUM concept introduces a new metaphor how communication between devices in a Ubicomp environment should be performed. The system contains the conceptual RAUM model as a theoretical base and the RAUM architecture that puts this concept in a concrete form.

In the next section an example for a spatial dependent application should clarify the problem and application area. Then the RAUM-system itself is presented. Details of the RAUM concepts and the architecture will be described next, followed by construction and implementation of the RAUM architecture, which together make up the system. The system itself will be illustrated by three example applications and related work is discussed in the end.

2 Example: ElectronicManual

The scenario "ElectronicManual" is introduced as an example to help to understand the model presented in the following sections. Purpose of the ElectonicManual application is to provide a common platform for accessing manuals of devices in the environment and for controlling these devices. The ElectronicManual scenario (Fig.2) consists of a program, the Browser, running on a handheld device and a surrounding Ubicomp environment built on top of the RAUM system. In the electronic manual example devices in the environment can communicate with each other and with the handheld device. When a manual of e.g. a video recorder located in the environment should be accessed the manual is downloaded onto the handheld device; it is then possible to read the manual and to control the video-recorder with the handheld. If multimedia content in the manual should be made available the handheld device facilitates surrounding output devices like a TV set or a HIFI in the environment to make them visible or audible. Spatial order is used in two ways in this example: First, the device that should be controlled is in a certain spatial order to the handheld device, here in front of the device. Second, the handheld device makes use of devices that are located around in a certain distance to help displaying multimedia content.

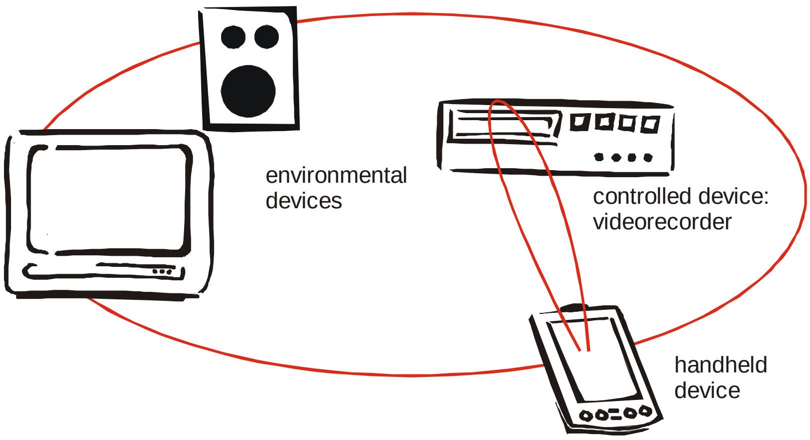

Fig. 2. The ElectronicManual scenario: A handheld device running the main application defines two scopes of interest. The first scope is to the device that should be controlled (here a video recorder). The second scope contains all devices in front in a certain distance. This second scope is used to ask these devices to assist the handheld device by displaying multimedia content.

3 Model

This section explains basic concepts of the RAUM-system and defines the RAUM term. Communication in RAUM is compared with other communication concepts.

3.1 Basic concepts

The way humans communicate with (computerized) objects and with each other in the physical world (source domain) is taken as a metaphor for communication between computerized objects in the virtual world (target domain). The proposed spatial structure for ordering physically existent interaction objects influences the possibility and the form of the communication. Communication is restricted to the spatial location of the object, but the object at the same time possesses the same image of the spatial environment as a user. According to the thesis the image of the distribution of objects existing in the virtual and in the physical world is equally spatially ordered. Then communication in the virtual world (i.e. between programs) occurs in the same way according to spatial structures (Figure 1). As we will see below, communication depending on the spatial layout should limit the scope of communication.

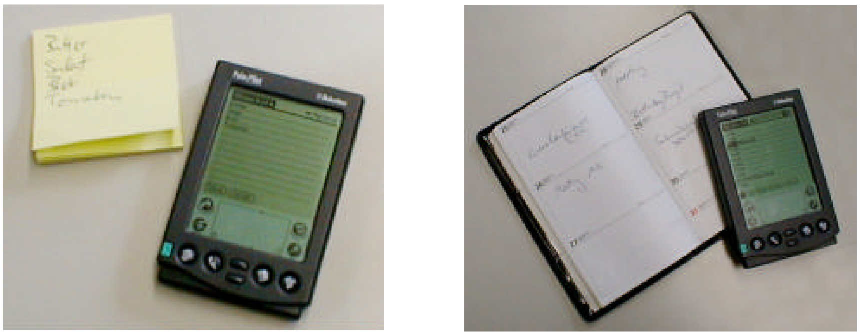

Fig. 3. Electronic applications and their paper counterparts. When a user carries out a task he recognizes the application, not the device. Using conventional interfaces these applications match to a physical object that helps to carry out the task (left: MemoPad, right: Calendar). To save space a computerized object (here a PDA) integrates several applications, but a user recognizes only the running one.

Application Objects. The term application objects (short: *objects*) stands for the combination of a physically available item (short: *item*) and a *program* serving a special purpose. This definition refers to the “obvious” function of the object to a human, the function a human recognizes as the purpose of the object at the time in question. Some items can have more than one purpose, sharing different functions over the time. In this case there is more than one object located on such an item. The set containing all objects located on one item is called an *object group*. According to the definition all objects have a physical presence when selected and a virtual presence. The virtual presence exists always in the virtual world independent of the fact whether the corresponding function is selected on the item or not.

For example a Personal Digital Assistant (PDA) handheld computer contains several programs (e.g. a memo pad and a calendar, Figure 3). In terms of the model, the item PDA hosts an object group that contains the objects “memo pad” and “calendar”, because both are obvious functions when a human selects these applications on the PDA.

3.2 RAUM

The RAUM concept introduced in this paper allows modeling of objects based on a spatial order structure supporting and controlling communication. The RAUM-system defines a "location-based ***R***elation of ***A***pplication objects for communicating ***U***bicomp event ***M***essages". This relation defines the communication space containing objects that communicate among each other with the help of a spatial aware and spatial-dependent communication system.

Definition: A *RAUM* (short for RAUM scope) is defined by the relation of all *objects o*, under the restriction that the corresponding physical items lay in a *scope S:*

$$o \rho S \text{ where } o \in \{\text{all objects}\} \text{ and } S \in \text{Powerset(location description)}$$

Definition: A *scope* is a spatially defined area of the physical world that matches an according area of the virtual world.

The order of objects in the virtual world matches the order of corresponding items in the physical world. Because objects are ordered the same way in the physical and virtual world a scope defines the corresponding area in both worlds.

Scopes are defined according to a location description determining the boundaries of the scope or according to a relative range e.g. in relation to an item. Scopes are physical spaces that can be either described symbolic, geometric or hybrid; the RAUM-system follows a description method similar to [19].

Objects in RAUM communicate with each other using message events (short: *events*). Objects that take part in a RAUM receive all message events from all other objects in the same RAUM regardless of their capabilities. RAUMs are initiated by one or more objects. During initiation a scope is defined cooperatively. This scope can change over time, depending on the need of the object programs. The initiating objects describe the scope of the RAUM; objects that are in the scope of the RAUM after the definition are automatically included in the relation.

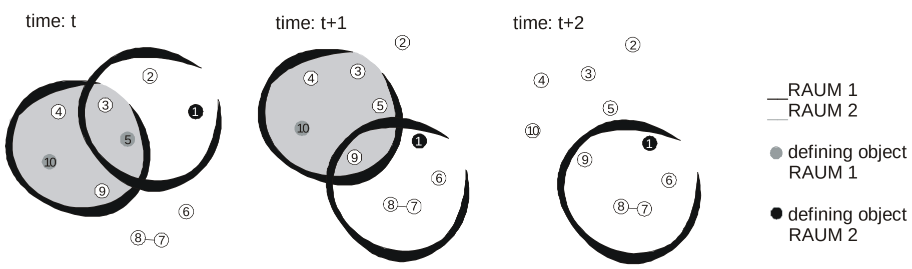

Fig. 4. Example how RAUMs are working. 3 states of a room containing 2 RAUMs: RAUM 1 with defining object 1 is moving downwards, while in RAUM 2 defining objects are giving up. As a consequence RAUM 2 vanishes

Figure 4 shows two intersecting relative RAUMs containing several objects that reflect the location of the corresponding physical items. RAUM 1 is defined by object

1; Objects 5 and 10 cooperatively define RAUM 2. Both RAUMs have objects in common. RAUM 1 is defined relative to Object 1; therefore, Object 1 is called *defining object.* As object 1 moves, RAUM 1 moves with the object (Fig.4 middle), leaving some objects behind and including others. While in Figure 4 (left) RAUM 2 is defined relative to Object 5 and Object 10, in Figure 4 (middle) Object 5 gives up his function as a defining object. Later Object 10 gives up its function as a defining object (Fig. 4 right). At the same time as Object 10 gives up its function RAUM 2 vanishes.

Every RAUM is in one of three possible states: initializing, running or dissolving. RAUMs are cooperatively created by the defining objects. The objects define the parameters of a RAUM and set the spatial definition. The definition determines indirectly which other objects take part in a RAUM. If new objects should be included as a defining object all defining objects have to agree on this.

Objects can be part of more than one RAUM. An object can even be a defining object in more than one RAUM, if the program running on the object requires this. In this case, events from one RAUM can not be seen in the other RAUM with the exception of objects located in the intersection of both RAUMs.

In the ElectronicManual example two RAUMs are defined: The first RAUM is in front of the handheld device and consists of the handheld and the device that should be controlled and whose manual should be displayed. Both devices together define this RAUM. The second RAUM is in a circle around the handheld; this second RAUM is used for dispatching multimedia content in the environment and is defined only by the handheld device.

3.3 Communication in RAUM

Communication in RAUM is *location oriented* and *spatial-dependent:* Communication partners are selected according to their spatial arrangement. In the ElectronicManual example the handheld device communicates with a radius of 2 meters in front of the device (Figure 1). This RAUM does not depend on the provided services of the objects. If a multimedia content is to be shown, an event is dropped and the appropriate device capable to handle the request picks up the event and shows the content.

On the other hand, in distributed or mobile computing systems (and also in Internet based services like the Web) communication partners are selected according to the service they provide, abstracting from the (geographical) location of the service. Here, these communication networks are referred to as *service-oriented* networks. The request for a service - e.g. a database or a printing service - as a selection criterion for the communication partner is typical for service-oriented communication. After finding this service (mostly via explicitly entering the network address as a virtual location by the user) communication is established between the service client and the service server. Physical location is not of any concern in service-oriented networks. Either location is not of interest or the assignment of virtual addresses to physical location is left to the user.

The concept of spatial-dependent communication is complementary to service-oriented communication. While spatial oriented communication abstracts from services of the communication partners, service-oriented communication abstracts from location. When creating worldwide service systems, abstracting location is a good choice as long as the service can be handled in the virtual world. In Ubicomp environments many services are performed in the physical world. Spatial-dependent communication as used in the RAUM-system supports exactly this kind of working environment: highly flexible communication directly coupled to human-computer interaction tasks. The spatial restriction of communication does not only direct communication automatically to the objects that are of interest in the local Ubicomp application, but also takes away the cognitive load from the human to imagine abstract distant devices or objects.

Communication in the RAUM-system is realized via event messages. Inside every RAUM, event messages are distributed (almost) without restriction between the objects taking part in the RAUM. The structure of events is generally freely defined. To ensure a communication basis a small number of events are predefined. Events are classified according to communication patterns. Patterns are then classified in pattern classes. Some of the patterns have to be understood by every object taking part in the RAUM-system. The common patterns cover functionality for creating, destroying, joining and dividing RAUMs. New functionality is introduced by an object through defining a pattern and ordering this pattern into the pattern class structure. In the ElectronicManual example requests for displaying multimedia content are defined as such a pattern; events built according to this pattern must be understand by devices that should take part in the ElectronicManual system. Apart from the communication pattern every event message also contains the senders location and the class of the pattern that this event belongs to.

3.4 Implications of the RAUM-system

In the following sections the major implications of the RAUM-system are described. Some principle implications concern services, network organization, communication cost, context and privacy. These implications also clearly indicate the advantages of the RAUM-system in Ubicomp environments.

Services. Services are implicitly provided in RAUM. As every object can drop an event (can be modeled as a service request) and every object can pick up events (can be modeled as a server) from the view of an object the whole RAUM reacts as one multi-purpose server. Because services in the system are bound to physical items the availability and disappearance of services can be followed by the human making applications more transparent.

Self-organizing network environment. The administrative burden of maintaining hundreds or thousands of services is taken away by the RAUM-system through restricting the number of servers via restricting the location and the use of information appliances [24] as services servers. In RAUM based systems the network and service structure is self-organizing and does not need additional environment servers or set-ups.

Reduced communication cost. To indicate the potential of the RAUM-system communication costs are compared to communication costs of service-oriented networks. The comparison is performed in a Ubicomp scenario with *n* objects located in a given area e.g. an office room. Let there be n_{or} objects on an average in a RAUM, and the object *o* should take part in n_{ro} RAUMs. Also let *d* be a correcting factor (because some objects are in more than one RAUM) defined as

$$d = \frac{\sum^{n} \frac{\sum \#\{objects \in RAUM \mid o_n \in RAUM\}}{\# \bigcup \{objects \in RAUM \mid o_n \in RAUM\}}}{n} \,. \tag{1}$$

For comparison communication costs to request a service, which is a typical communication in service-oriented systems, should be regarded; the results are similar when regarding other communication scenarios. Service based communication (e.g. Jini) normally uses a register server instance to register services and to inform other objects about services in a certain area. So in service-oriented communication first a call to such a register server is needed to request a service. Asking a register server first requires a broadcast to find the server. Next, a message is sent back to the requesting object, and then a message is sent from the requester to the actual server. Assuming all message cost as 1 the overall service call costs are:

$$cost_{service} = n_b + 2 \quad \text{for service-oriented communication.} \tag{2}$$

In the RAUM system communication costs depend on the number of objects that share a RAUM with the requesting object. The message must be sent to all these objects, therefore the cost is the cost for a multicast *c* to $n_{or} \cdot n_{ro} \cdot d$ objects:

$$cost_{RAUM} = n_c \text{ for RAUM communication.} \tag{3}$$

In the case the underlying network supports multicast and broadcast, the cost for both multi- and broadcast is 1 and there are *m* service request messages the overall costs for both communication methods are

$$3\,m \quad \text{for service-oriented} \qquad m \quad \text{for RAUM communication.} \tag{4}$$

indicating a clear advantage for RAUM communication.

If the underlying network does not support broadcast and multicast communication the cost increases in both cases (note: to find the register server for every service call one broadcast is needed. Due to the mobility of almost all objects in ubicomp environments objects frequently leave areas of a register server.).

$$m\,(n_{or}\ x\ d + 2) \quad \text{(service-oriented)} \qquad m\,(n_{or}\ n_{ro}\ d) \quad \text{(RAUM).} \tag{5}$$

where x is the number of RAUMs which must be crossed by the broadcast to find the location server neglecting the costs to compute x.

We now assume a 5 x 5 m^2 room with 70 RAUMs and an average of 10 objects per RAUM while every object takes part in about 3 RAUMs and a correction factor of d=0.7. Figure 5 shows a comparison of the cost for both communication methods. The cost for service based communication is significantly higher than the cost of RAUM communication when more than 2 RAUMs (~1.5 m^2) are needed for a broadcast to find the register server.

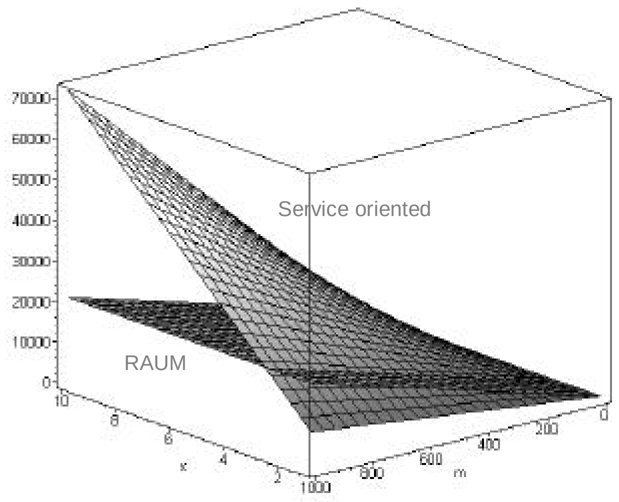

Fig. 5. Comparison of communication cost between service-oriented communication and RAUM communication. In Ubicomp environments with many objects and a high message rate RAUM communication is significantly superior to service-oriented communication

Context. Context is an important factor in Ubicomp systems. Beside location, context is needed as a decision base for many applications [26]. Context can be derived from the environment in two ways: either by equipping every object with appropriate hard- and software or through exchanging detected and preprocessed contexts among a set of objects. In a RAUM-system context is available inside the RAUM scope from all objects taking part. Every object, e.g. equipped with special sensors can transmit its findings to every other object in scope. In that way the community of objects provides a common context to every object in scope. Communication between objects is not only used for notification and command messaging but also contains sensed information. This way, the RAUM-system reduces the complexity of context detection introducing a system of shared context events.

Location based authorization. The assurance of privacy is of major concern in Ubicomp [5, 20]. Actual proposals suggest a feedback mechanism [5] so that the user knows when he is watched allowing him to move to a non-watched room, or suggest the possibility to switch off the system [15]. Both ideas are not really satisfying, because the system functionality is lost when the users do not want to be watched.

Spatial-dependent RAUM-system communication enables privacy with two concepts without loosing functionality of the system. It prohibits distribution of events e.g. movements of persons detected by sensors outside the wanted scope and allows anonymous access to sensitive data without reducing security.

Spatial limitation of communication prohibits the global access to information and prevents the set up of "Orwellian" scenarios. Communication is only allowed in specified boarders e.g. in a room. Furthermore this kind of communication allows an exact definition of the scope of events and therefore hinders supervision of persons - e.g. in a company - without reducing functionality of the system.

In spatial-dependent communication personal authorization for access can be dropped, because objects (belonging to a person) that are allowed to enter an area are allowed to use devices and data in this area. Therefore, only the location not the identity of the using object must be verified. This location based authorization metaphor is derived from daily life: Is someone authorized to enter a room, e.g. because he possesses a key for the room, he can read all documents in this room without additional identification.

4 Implementation

A couple of objects belonging to a RAUM make up a set that is modeled as a tuple space [12] with some additional operators. This tupple space operates on the RAUM scope, or short RAUM. Tupple spaces are a widely used basis for event communication. Examples for other systems based on tupple spaces are Jini [30] and T Space [18]. Tupple spaces are used to implement a RAUM communication scope. Every tupple space contains objects that are within the scope of a RAUM relation. A RAUM is therefore implemented as a tupple space using special operators for the setup of the RAUM and the definition of the scope. Additional operators are needed to take practical conditions of Ubicomp environments into consideration. For example, one operator set allows the determination of objects in a RAUM at a certain moment with a certain probability. Others allow asynchronous communication or the union and disunion of RAUM spaces.

4.1 Architecture

Events of an object that takes part in the RAUM are distributed to all other objects in the RAUM. If an object is a member of several RAUMs all objects in all these RAUMs receive events from the object. In principle, all objects in a RAUM have the same rights so dropping and accepting events is possible by all objects.

From the program's point of view object communication is handled by getting and putting event messages to the RAUM. All objects running on the same item put their event messages to the assignment level of the 3 level RAUM-system stack. At this level, events are assigned to the RAUMs the object is member of. As a result a set of RAUM-event pairs are then passed to the RAUM-system administration level. This is the level where the RAUM-system is managed, which for example decides about creation or dissolution of a RAUM. The network layer then assigns events of RAUMs to events to Objects, if needed by the underlying network. For example, if an IP stack of an object does not support multicast addressing, IP addresses of all RAUM-objects have to be computed at this level and events must be sent out object by.

5 Examples

Three example set-ups should illustrate the RAUM concept and application areas of location dependent communication. In the UbicompBrowser and ElectronicManual scenario surrounding computerized objects and information appliances in the environment are cooperating to enhance and simplify Human-Computer Interaction. The AmbientTelepresence system shows the integration of new kinds of computerized objects of the everyday life into application scenarios and the combination of RAUM and service-oriented communication.

5.1 UbicompBrowser

The UbicompBrowser [2] is a system that applies Ubiquitous Computing to the World-Wide Web extending the Web into our everyday environments. It extends the browser concept by replacing the standard Web user interface with a handheld access and control device and surrounding output and controllable devices. This ubiquitous user interface is determined dynamically based on the location of the handheld control. The UbicompBrowser improves Web accessibility by realizing a ubiquitous environment-based user interface, and by extending accessibility to environment-specific resources.

The UbicompBrowser setup consists of the UbicompBrowser program running on a PDA, as shown in Figure 6. This object defines a RAUM scope, which is bound to a local physical space around the PDA. When a PDA user accesses a document of a media type that can not be shown at the PDA, e.g. a movie or audio stream, the UbicompBrowser puts an event message to the RAUM asking for an object in scope to display the document. This event is taken from the RAUM by a device that matches the condition for displaying the document. Next the displaying device retrieves the document from the Web and puts out the content while synchronizing with the UbicompBrowser object via RAUM communication.

5.2 ElectronicManual

The principle of the ElectronicManual used as an example throughout the paper. The ElectronicManual [4] is build on top of the UbicompBrowser. The ElectronicManual gives users better assistance in understanding and using their devices. There are two fundamental contributions to the ElectronicManual: First the uniform access to information over devices that follows a single metaphor instead of forcing the user to rethink about the form and access method with every new device. Second the ubiquitous access to the current information, possibly enhanced with multimedia contents, learning programs and support channels. The ElectronicManual setup shows that Ubicomp environments can bring real advantages for everyday tasks. Although electronic manuals are not new, the ubiquitous access to these informations and the possibility to integrate control and information access in one user interfaces significantly improves usability.

5.3 Ambient Telepresence

The Ambient Telepresence System [3] is set up to demonstrate the connection of two or more distance rooms with the help of RAUM technology. Ambient Telepresence itself is introduced as a method to give someone the feeling that someone else is present while he is not. In contrast to other telepresence approaches, ambient telepresence is focussed on mediating background activity. Ambient telepresence is based on generation of remote presence from handling everyday objects in work environments, based on Ubiquitous Computing and context-awareness technologies. For experimentation, we have developed *MediaCups*, coffee cups in our office environment equipped with sensor, computing and communication facilities. Such a cup is an item with a corresponding program (running on a Microchip PIC microprocessor) that takes part in a RAUM. MediaCups located in dedicated office rooms (each of the office is defined as a RAUM) are interconnected with loudspeakers in the other offices. Ambient Telepresence is realized by associating cup movements to sounds: Detected cup movements of the cup are transferred as events to the remote location and made audible as associated noises. The Ambient Telepresence system interconnects two or more RAUMs. While events in a local RAUM are detected by the RAUM-system, normal IP based communication is used for interconnecting the RAUMs. This example setup also shows the integration of new computerized objects of the everyday life, e.g. the MediaCup.

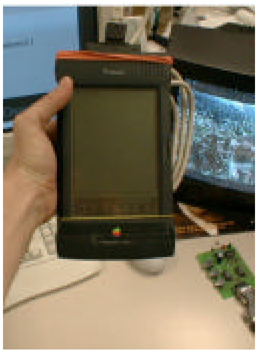

Fig. 6. Two example set-ups of the RAUM system. The left picture shows the PDA with the UbicompBrowser, a PC based network access node and a TV set as controlled device. Right the MediaCup with an IrDA based access point as it is used in the Ambient Telepresence example.

6 Related work

Several suggestions have been made how Ubiquitous Computing environments, especially with regard to Human-Computer Interaction (HCI) problems, should be constructed or how problems that arise in this field have to be solved (e.g. [7, 21, 23, 25]). Some of these suggestions were implemented as experimental set-ups. Test-scenarios were constructed and the usefulness of these suggestions was verified. Such investigations help specifying conditions for Ubiquitous Computing systems and give guidelines for the construction of such systems.

The first Ubicomp systems ParcTab [28] and Responsive Environments [9] from XeroxParc used X-Windows based technology. ParcTab uses human computer interfaces similar to those used at desktop computers. More recent systems like Domisilica [22] from Georgia Tech also provide desktop computer bound interfaces, but additionally integrate objects of the everyday environment into the system. To interconnect both interface types, a Multi-User Dungeon (MUD) is used here as a joint communication space.

The ReactiveEnvironment [11] and the ActiveOffice [8] project at the University of Toronto are examples aiming especially at non-desktop computer interfaces. This projects deal mainly with the question of complexity of the human-machine interaction in environments enhanced with computer technology. In Active Office Buxton mentioned five Design Principles for minimising the complexity of the Ubicomp application for the user. These design principles refer to a close correlation of the *physical world* of physical items and artifacts and the *virtual (electronic) world* of programs and data. An approach to collect experience with new kind of tangible interfaces for connecting both worlds is the Tangible Bits project [16] at the MIT MediaLab.

The question of how to match both worlds is also stated as one of the major problems in other research. Some proposed Ubiquitous Computing systems match physical artifacts [7, 8, 14, 22, 29] to virtual objects. Some also have concepts for matching social interaction [8] to their virtual counterparts. All existent Ubicomp systems differ in the way, how the virtual object reflects its physical behavior. None of the above mentioned systems reflect the spatial-dependent communication concept found in the physical world. Spatial order has a role in [8] as part of the design principles. For manipulative user interfaces (e.g. [6, 13, 29]) or augmented reality systems (e.g. [10]) the meaning of spatial arrangement provides a basis for the applications functionality. Therefore, it has been implicitly implemented in these applications. Some systems [1, 10, 15, 27] use a location services to provide explicit information about the spatial layout of objects in the environment.

7 Conclusion

RAUM was introduced as a concept and a system to support co-operation and communication between objects in Ubicomp environments. RAUM communication takes the spatial order of physical objects into account. This paper indicates that in Ubicomp environments such spatial-dependent location-aware communication is superior to other communication strategies. The location-aware RAUM-model, the system, architecture and implementation are introduced in this paper. The RAUM-system supports construction and operation of Ubicomp applications in many ways. Like humans, the RAUM-system orders objects in the environment according to their spatial location making this order available to applications. Such location dependence allows users to have a transparent view of ongoing program activities. Three example set-ups show how to integrate physical devices through simply writing the service for taking part in the RAUM-system. This is even possible for very restricted devices with embedded microprocessors and small amount of storage space such as MediaCups. As a result, allowing physical set-ups to be built very fast brings the flexibility known from component-based software design to Ubiquitous Computing environments. Ubicomp environments with a large amount of computerized objects benefit also in other ways from the proposed RAUM concept. Due to the spatial restriction in RAUM privacy can be assured without crippling functionality or security, communications is carried out more efficient and management of Ubicomp environments is facilitated.

The proposed system uses exact defined boundaries for RAUM scopes (ignoring practical conditions as the uncertainty when determining a location caused by the imperfection of current technology). In contrast humans have a more blunt concept of the area they are interested in. Future experiments will show if the presented concept of exact defined scopes provides enough support for applications in Ubicomp environments or if a concept using a more blunt scope definition is significantly superior. More practical questions rise when implementing a RAUM-system. Upcoming network technologies with a very limited transmission scope (e.g. Bluetooth) facilitate the construction of spatial-depended communication, but existing network technologies must also be integrated. Criteria have to be found to optimize the setup of such mixed networks for RAUM communication.

8 References

1. Beadle, H. W. Peter, Harper, B, Maguire Jr, G. Q, Judge, J.: Location-aware Mobile Computing: Proc. IEEE/IEE International Conference on Telecommunications (ICT'97). Melbourne. April (1997)
2. Beigl, M., Schmidt, A., Lauff, M., Gellersen, H.: UbicompBrowser: 4th ERCIM Workshop on User Interfaces for All. Stockholm, Sweden. October (1998)
3. Beigl, Michael, Gellersen, Hans-Werner: Ambient Telepresence: Proceedings of the Workshop on Changing Places. London, UK (1999)
4. Beigl, Michael: ElectronicManual: 8th International Conference on Human-Computer Interaction. München, Germany (1999) to be published
5. Bellotti, V.: Design for Privacy in Multimedia Computing and Communication Environments. Technology and Privacy: In Agre, Rotenberg, : The New Landscape, MIT Press (1997)
6. Brave, S., Ishii, H. and Dahley, A.: Tangible Interfaces for Remote Collaboration and Communication: in Proceedings of CSCW '98, Seattle, Washington USA, November, ACM Press, (1998), 169-178.
7. Buxton, William A. S.: Living in Augmented Reality: Ubiquitous Media and Reactive Environments: In K. Finn, A. Sellen & S. Wilber (Eds.). Video Mediated Communication. Hillsdale, N.J.: Erlbaum, (1997) 363-384
8. Buxton, William A. S: Ubiquitous Media and the Active Office: Ubiquitous Video: Nikkei Electronics, 3.27 (no. 632), (1995) 187-195
9. Demers, Alan J.: Research Issues in Ubiquitous Computing: In Proceedings of the Thirteenth Annual ACM Symposium on Principles of Distributed Computing. Los Angeles, California, USA. August 14-17, (1994) 2-8
10. Feiner, S., MacIntyre, B., Seligmann, D.: Knowledge-Based Augmented Reality. Communications of the ACM, Vol 36, No.7, July (1993)
11. Fitzmaurice, George W. :Graspable User Interfaces: PhD Thesis. Toronto (1996)
12. Gelernter, David : Generative communication in Linda: ACM TOPLAS, Vol. 7, No. 1. January (1985)

13. Gorbet, M., Orth, M. and Ishii, H., Triangles: Tangible Interface for Manipulation and Exploration of Digital Information Topography: In Proceedings of Conference on Human Factors in Computing Systems CHI '98, ACM Press, Los Angeles: April (1998), 49-56
14. Harrison, Beverly L., P. Fishkin, Kenneth, Gujar, Anuj, Mochon, Carlos Want, Roy: Squeeze Me, Hold Me, Tilt Me! An Exploration of Manipulative User Interfaces; CHI 98, Los Angeles CA, USA, 18-23. April (1998)
15. Harter, Andy, Hopper, Andy**:** A Distributed Location System for the Active Office: IEEE Network, Vol. 8, No. 1, January (1994)
16. Ishii, Hiroshi, Ullmer, Brygg: Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms: CHI97 (1997)
17. Kirsh, David: The intelligent use of space: Artificial Intelligence 73(1-2).(1995) 31-68
18. Lehman, Tobin Mclaughry, Steve, Wyckoff, Peter: T Spaces: The Next Wave : Hawaii International Conference on Systems Sciences (1999)
19. Leonhardt, Ulf, Supporting Location-Awareness in Open Distributed Systems: PhD Thesis, Imperial College, University of London. May (1998)
20. Leonhardt, Ulf, Magee, Jeff: Security Considerations for a Distributed Location Service: Journal of Network and Systems Management, 6(1). March (1998) 51-70
21. MacIntyre, Blair, Feiner, Steven: Future Multimedia User Interfaces: In Multimedia Systems, Springer Verlag, (1996) 250-268
22. Mankoff, J., Somers, J., Abowd, G. D.: Bringing People and Places Together with Dual Augmentation: Collaborative Virtual Environments. Manchester. June (1998). 81-86
23. Myers, Brad, Hollan, Jim, Cruz et al Isabel: Strategic Directions in Human-Computer Interaction: ACM Computing Surveys, Vol.28, No4. December (1996)
24. Norman, Donald A: The Invisible Computer: MIT Press. (1998)
25. Salber, D., Dey, A.K., Abowd, G.D.: Ubiquitous Computing: Defining an HCI Research Agenda for an Emerging Interaction Paradigm: Tech. Report GIT-GVU-98-01. Feb. (1998)
26. Schmidt, Albrecht, Beigl, Michael, Gellersen, Hans-Werner: There is more to context than location: Proceedings of the International Workshop on Interactive Applications of Mobile Computing (IMC), Rostock, Germany November (1998)
27. Shafer, S., Krumm, J., Brumitt, B., Meyers, B., Czerwinski, M., Robbins, D.: The New EasyLiving Project at Microsoft Research: Joint DARPA/NIST Smart Spaces Workshop. Gaithersburg, Maryland. July 30-31, (1998)
28. Want, R. Schilit, B.N., Adams, N.I., Gold, R, Petersen, K., Goldberg, D., Ellis, J.R, Weiser, M.: An overview of the PARCTAB Ubiquitous Computing experiment: IEEE Personal Communications 2(6), (1995) 28-43
29. Wisneski, C., Ishii, H., Bahley, A., Gorbet, M., Braver, S., Ullmer, B., Yarin, P.: Ambient Displays: Turning Architectural Space into an Interface between People and Digital Information: Cooperative Buildings. Springer Verlag, February 25-26 (1998)
30. www.jini.org