A location model for communicating and processing of context

MICHAEL BEIGL, TOBIAS ZIMMER AND CHRISTIAN DECKER

TecO, University of Karlsruhe
Vincenz-Priessnitz.Str.1, 76131 Karlsruhe, Germany

{michael, zimmer, cdecker}@teco.edu

Abstract

Location is one of the most important elements of context in ubiquitous computing. In this paper we describe a **location model**, a spatial-aware communication model and an **implementation** of the models that exploit location for processing and communicating context. The presented location model describes a location tree, which contains human-readable semantic and geometric information about an organisation and a structure to describe the current location of an object or a context. The proposed system is dedicated to work not only on more powerful devices like handhelds, but also on small computer systems that are embedded into everyday artefact (making them a **digital artefact**). Model and design decisions were made on the basis of experiences from 3 prototype setups with several applications, which we built from 1998 to 2002. While running these prototypes we collected **experiences** from designers, implementers and users and formulated them as **guidelines** in this paper. All the prototype applications heavily use location information for providing their functionality. We found that location is not only of use as information for the application but also important for communicating context. In this paper we introduce the concept of **spatial-aware communication,** where data is communicated based on the relative location of digital artefacts rather than on their identity.

Keywords*: location model, spatial-aware communication, RAUM, guidelines, experiences, digital artefacts*

1 Introduction

More and more Ubiquitous computing scenarios arise where computers are embedded into everyday objects like chairs, doors or heating as secondary artefacts. In such scenarios everyday objects perceive their environment with the help of sensors, interpret this information and hence derive context based on their perception. This information is then in many cases communicated to provide additional information to other objects. We define such enhanced everyday objects as "digital artefacts", which are generally, and throughout this paper, referred to as "artefacts".

For digital artefacts, one of the major contexts in such environments is location. Moreover, most other findings of the environment as sensor or context information are only valid or make sense in conjunction with the location of their creation. We believe that there are three reasons for this: Firstly, the user utilises location as a major order criteria for objects and information and to carry out tasks. Kirsh writes in one of his essays [1]: "But, in having a body, we are spatially located creatures: we must always be facing some direction, have only certain objects in view, be within reach of certain others. How we *manage* the spatial arrangement of items around us is not an afterthought: it is an integral part of the way we think, plan, and behave." Models and applications that are designed to present information or to react in a human understandable way therefore have to take the location of things into account. Secondly, the designer and application programmer (also humans) transfers their association of the structure of the environment into program logic at design and development time. Although it is possible to design applications from a viewpoint where human experience plays no role we found that in practice designers and programmers tend to ignore or shift such viewpoints. Thirdly, technical systems e.g. communication or sensor systems are intrinsically bound to location (e.g. communication or sensor range) and therefore deliver a view of the world that is implicitly bound to location.

Today, many Ubicomp applications use location implicitly. E.g. the communication range of a technical network is often adjusted to influence the propagation scope of sent or received transmissions [2]. In this paper we want to present examples that are explicitly aware of the location information and take advantage of having this knowledge. This awareness is made available by the RAUM location model that is presented in this paper. The RAUM location model describes locations of artefacts relative to the environment – e.g. to the rooms in an organisation – and in relation to each other.

In the remainder of the paper we will first present three examples where location, mainly from the perspective of within an organisation (e.g. a company or a university) is used. These examples are discussed from three viewpoints. Experiences from these examples led to guidelines that have directed our design of the RAUM model. In the third section we present the design and use of the RAUM model in more detail. The fourth section shows the implementation of the model and parts of the examples presented before. The concluding sections discuss related work and provide an outlook on future developments.

2 Experiments and experience

The experiences presented in this section are based on three major prototypes and about 200 digital artefacts that we have set up in our lab between 1998 and today. These prototypes and most of the applications presented here are still in operation today. They provide a valuable resource for us to test and experiment with different aspects of ubiquitous computing. The first of these projects is the MediaCup, a collection of applications centred on information derived from a coffee cup that has embedded computing, sensing and communication facilities. Second is the MemoClip, a system to remind people based on location information. This application is able to store such location and other context information for later reuse. Thirdly, there is the Smart-Its project, an ongoing project in which we develop a standard platform for integrating computing, sensing and networking into everyday objects and the associated applications. All these projects use and model location information as a central context. In the course of the projects the RAUM location model, introduced with the first prototype, was developed and changed to adapt to findings we discovered throughout modelling, designing, implementing, running and using these prototypes.

This section will present experiences using the RAUM location model from three different views: The users view is the view of the human living or working in the ubiquitous computing environments we build in our example assembly. Users are one client of the location model, e.g. when requesting or monitoring location related information. Nevertheless, it is not necessarily the case that all users explicitly consume or use the information and the applications. Many users are only implicitly involved in an interaction e.g. by using an object with embedded electronics inside. The second viewpoint is the applications view. Location information is needed at development time as well as at run time of the application. E.g. the developer needs information about the location model and the operators for implementing location aware applications. Subsequently, Ubicomp applications use location as a context for making decisions, triggering and reacting to events, to present information and to store location as data. Therefore location information has to be expressed, transformed, presented and communicated to humans and to computing objects. Third is the technologies view. Technology is seen here as a support for the location system. Therein, how to transfer the specification from the location model into something technologically realized is one issue here. Other issues are the simplicity of the compilation, administration and maintenance of the resultant technical implementation.

In this section we will briefly explain the design and setup of the three projects and the implementation of the location model therein. For each of the projects we will have a discussion based on the three views introduced above. This discussion is drawn from experiences during assembly of the prototypes, based on reports from over 30 persons involved in the design, implementation, maintenance and usage. In addition to reports on experiences, the results also allowed guidelines to emerge that have partially influenced the design of subsequent prototypes built.

2.1 MediaCup

This prototype consists of several applications that rely on context information obtained from the MediaCup [3]. The MediaCup (figure 1) itself is an everyday cup equipped with sensing, communication and processing capabilities. The cup is able to deliver introspective status information such as "someone is drinking", "I am hot" etc., to other objects around. Such communication is defined as **spatial-aware communication,** in contrast to the identification (e.g. IP number) dependent communication used in networks today. The concept of spatial-aware communication introduced with the MediaCup prototype allows two artefacts to communicate to each other because they are nearby and not because they know each other.

The MediaCup is our oldest functional prototype, and has been running for about 4 years in our offices. Objects used in this setting are cups, electronic doorplates, coffee machines and coffee pots with embedded computers, computer watches and Internet servers. About 10 cups, several doorplates and other objects are included in the operational configuration. Each of these objects host particular applications that are intended to complement everyday activities through functional support and simplification.

Figure 1: MediaCup: An implementation of an digital artefact

In the *electronic doorplate application* a doorplate observes and collects context information in the room it is responsible for and reacts accordingly. In this application, location information is used to separate useful information from unnecessary information. In the "*coffee machine and coffee pot"* application a coffee machine brews coffee automatically based on the status of the coffee pot and those of the coffee cups. The resultant reaction is based on accumulated context information from objects within dedicated contiguous areas. The *computer watch* (figure 3) application displays information from one dedicated cup that is held in very close proximity. This location criterion alone determines the information displayed. *Internet Server applications* are used to generate and render visualization of all cup states on a Web Server, or to log information for later reuse. This type of application may therefore be employed to collect organization-wide information. RAUM location information is required to determine organizational borders, controlling access to and correctness of information.

Human View. In the MediaCup project, location information is both used to attach additional details to sensed and communicated information (e.g. from the MediaCup or from the coffee pot) and to guide its delivery to the right place. We found that the human usage of the system is quite intuitive. Concepts like the ones utilized in the doorplate application are easily integrated into the organizational culture; that is, doorplates and signs attached to a door are used to express information about the room to which the door belongs. On the contrary, the use of the computer watch has to be learned, tested and tried. Nevertheless, as humans innately recognise things that are located nearby as belonging together (Proximity in Gestalt theory), the use of the watch was soon found to be natural behaviour. The use of the coffee machine was however even more difficult; although the location correlation of area to coffee machine was simple to understand, the logic of when coffee was brewed was not. The need to utilize context beyond that which is simply perceived, especially user intention, may be attributed to the logic complexity. The Internet server application did not present such a challenge; users have adopted the understanding of the Web server's role as a provider of organization-wide information, including ongoing projects, people employed etc. Therefore, additional information about the (organization-wide) status of cups is seen as a natural extension of useful content.

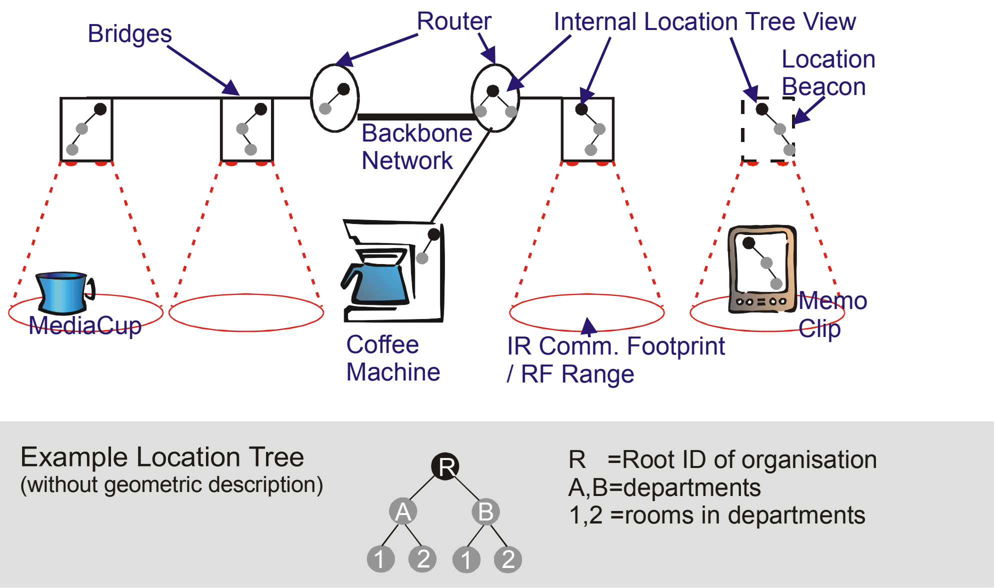

Figure 2: Example environment, location tree and artefacts communicating through spatial-aware communication

Applications View. The design and development of the MediaCup project was done in several stages. Consequently, access and presentation of location information has changed throughout the lifetime of the project. The first location model uses a rather flat hierarchy of places (areas of the technical system) that are identified by a number. However, using such a system provides almost no support to the application designer and developer. As a result, programs contained fixed knowledge of the environment that diminished the flexibility, scalability and portability of applications, with respect to alteration in the application environment (e.g. new offices or technical modification). Addressing this problem was the initial motivation behind developing the RAUM location model, which provides a framework for describing and expressing location information. Initially, this model was intended to only support the description of the location information, as none of the designed or planned applications at that time used geometrical information (note that the computer watch was one of the later applications). A purely semantic model was therefore designed. This model is capable of supporting the representation of one organization (as shown in the Internet server application), its divisions and its rooms (as in the doorplate application). As depicted in figure 2, the information was ordered in a tree structure, with the organization identifier (usually a worldwide unique 4 byte IP address, described as "R" in figure 2) at the root and integer room identifiers ("1", "2"…) as leaves. Divisions of the organisation, "A" and "B" in figure 2, are placed between the root and the leaves.

Subsequently, this tree was extended by additional geometric information in the leaves for expressing the position of objects inside a room. Nevertheless, the first implementation of the model was not entirely accepted by application programmers because location descriptions, according to the model, introduced a certain level of complexity when expressing and processing location information. It was still simpler to use lower-level technical identifiers of the location (e.g. the identifier from a location beacon) than using the location model description. To solve this problem, we introduced operators for the location description so that application programmers were able to query the location description through an application-programming interface. As one of the major advantages of the system we found the possibility to send and receive messages based on their location (the spatial-aware communication). Such communication behaviour allows us to write very small programs with quite complex behaviour. In comparison, older versions of the same program using identity-based communication were up to several times as large and complex.

Figure 3. MediaCup communicating with Computer Watch

Technologies View. The MediaCup uses a highly distributed approach for storing location information in the environment. Figure 2 shows an example setting where objects of the setup contain only partial location information sketched as a small location tree part in each object. For example, the coffee machine knows only about that part of the location tree where it is responsible. The use of infrared (IR) communication and location infrastructure allowed for the detection of object positions in about one square meter grid (the footprint). The entire system was seemingly very reliable from its inception. Nevertheless, there are still some problems to be resolved, mainly arising from the distributed approach. For instance, inconsistencies in location descriptors may arise in parts of the system but there are currently no consistency checks implemented. Additionally, the fine-grained location system required us to install a lot of network hardware due to the restricted footprint of IrDA compliant infrared transmitters. From a communications perspective, the delivery of required context information to applications is not possible by simply broadcasting; as such an approach would overload some of the low-bandwidth networks (e.g. IrDA based networks). Therefore, a more efficient routing mechanism, based on application need, is required.

Guidelines. From the MediaCup prototype we established the following guidelines for the further development of a location model and system as important:*[Spatial-aware communication]:* A location model should support context delivery/communication: Use location information to improve your communication performance. Most information is communicated between new types of embedded computers based on their location. *[Provide operators]*: a location model in itself is not of use; programmers do not have the tolerance to be bothered with such details – they will find ways to get around the location model instead. *[Application understandable location information]:* Location information must be provided in an application understandable way. Important location zone descriptions are Room and Workspace that should be identifiable from the application.

2.2 MemoClip

The MemoClip [4] is an application that was designed and implemented shortly after the MediaCup prototype. The MemoClip is a wearable PDA-like artefact that reminds a user of things he should do depending on *where* he is. A user can associate information to be remembered with a description of a location, download it onto the MemoClip, and then receives notification accordingly when entering the selected location.

Users View. Practically, the user initially enters location and associated information into the device and is then issued an automatic reminder from the device, upon re-entering the selected location(s). In contrast to the MediaCup prototype, location information is directly used (read and entered) by a human. The dialog for entering the information in the first prototype uses the RAUM tree information for presenting the location to the user in a form that is simple to understand. RAUM location descriptions refer to places inside an organisation or a building, e.g. your company or your private home. Due to the semantic description in the location tree, users in the given cultural background are able to anticipate location information very easily. The MemoClip application requires us to describe locations more fine-grained than room level to identify particular locations of interest. We addressed this issue by allowing additional semantic descriptions beyond room level in the RAUM location tree. Such details are important for describing contiguous areas inside rooms.

Applications View. A memorising system is expected to be very dependable. As a result, application programmers are especially in the need of particularly reliable location information. We also found that application programmers are not willing to pay attention to the characteristics of different technologies used to derive information. In the last implementation of the MemoClip application, both fine-grained infrared beacon based services and also cell-of-origin based radio frequency (RF) location services can be used. We address this issue by adding an uncertainty value to all location descriptions. As the MemoClip is often used while “offline”, location information has to be storable inside the MemoClip itself.

Technology View. For practical reasons there was a need to integrate different location detection technologies. In our setting we used infrared-based location beacons for fine-grained detection and RF based location detection for room-level detection (see figure 2, left). During the project, energy consumption was found to be a critical factor, both for our (wireless) beacon location system and for the mobile MemoClip. Constant request and sending of location information was found to be a major source of energy consumption in our beacons and MemoClip devices. Special request/reply protocols are delivered within this project to ensure minimal energy consumption while still being responsive to changes in the environment.

Guidelines. Due to the direct presentation of location information to the user in the MemoClip application many guidelines are found that have influenced the further development of the RAUM system. The following guidelines are found:*[Human readable location description]:* Provide a human readable and understandable form of the location description and the location model. Use semantic "common sense" description in your location model. Such descriptions are simpler to understand and allow direct presentation to the user without additional technology (e.g. middleware or services). The result is a system that can be operated even on small devices when off-line. *[Off-line access]:* Provide off-line access to information. Description of location information of interest has to be small enough to be carried around in mobile devices. *[Reliability and maintenance]*: Use simple-to-setup and maintain and reliable technology. There is often no chance to deal with all errors or a high error rate. *[Openness to new technology]*: Keep your system open to new technology; especially do not bind your location model to one technology. This allows the system to integrate upcoming developments. *[Energy consumption]*: Monitor energy consumption. Although this seems to be a practical issue there is a considerable effect to the overall functionality of the system and therefore affects the usability.

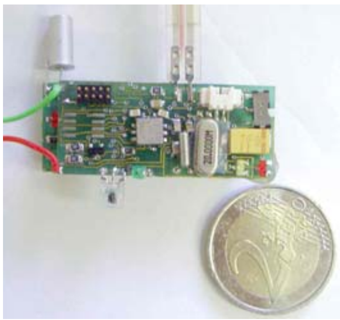

Figure 4. Smart-Its: A general sensing, computing and communication platform to be embedded into everyday artefacts

2.3 Smart-Its

Smart-Its [5] is an ongoing project providing a hardware and software platform for integrating computing, sensing and communication capabilities into everyday objects. Smart-Its consist of a hardware platform (the Smart-Its, see figure 4) with various types of sensors, a processor, memory and radio frequency (RF) communication. Smart-Its are user programmable devices bundled with a toolkit and operating system for adapting them to the needs of special applications. Smart-Its are especially intended to be attached to everyday objects (e.g. chairs, desks, pens) to enhance their functionality and to communicate contexts that are derived from the sensors. The communication stack integrated in the operating system allows to communicate either with other Smart-Its in communication range or with other objects and services through an infrastructure. Today, the Smart-Its platform itself does not hold a location description due to two reasons: Firstly, Smart-Its are intended to also work in a peer-to-peer network without any infrastructure, including for the detection of location. Secondly, Smart-Its are low-end computing devices that are not powerful enough to hold or process location information such as provided by the RAUM. Instead the Smart-Its infrastructure uses the RAUM location model to add location information to context data received by Smart-Its. This extended information data is then forwarded to other devices and applications, and is referred to as “location-stuffing”.

For experimental setups we produced about 180 Smart-It devices up to now. With the help of the Smart-Its platform we have built and are currently building several application settings. Of interest in this context are the Aware Office and the History Server application. In the Aware Office application objects like chairs, pens and doorplates of meeting rooms are enhanced with Smart-Its devices. These devices communicate with each other, to PC based computer systems and to backend services. As an example, the application enables doorplates to interpret usage patterns inside a room to change the display of the doorplate from “free” to “occupied”. Furthermore, the backend meeting-room scheduling service – a Web-based service normally used for pre-booking of rooms – uses this information for spontaneously occupying or freeing meeting rooms. We are also working on an application where meeting notes are automatically annotated with activity level (derived from the sensors embedded into objects of the room), pen usage patterns (from the pen sensors), still pictures and audio recording (from the controller PC of the meeting room). The History Server application is a service that allows logging and storing all information coming from different devices inside an organization. It also allows for adaptive filtering of information, compatible with PDA or PC based user interface.

Users and applications View. From a user’s perspective the presentation of location information is transparent to the user and needed only by the application itself in the Aware Office context. As many information types are handled by the application, location information must be interpreted at application level. It is especially important to retain access to all room-based information. An example is the determination of activity level in the room by an annotation service. To provide this information, only contexts *relative* to the current *room* are of interest. A location system has to support queries in a manner that is relative to a room.

Technologies View. The Smart-Its setup includes a large-scale integration of many of the former prototype systems. There is a need for a variety of tools to support setup and maintenance of location descriptions and equipment used for position determination. One of the goals of the History Server was to deliver basic information for such tools.

Guidelines. In the Smart-Its assembly only two new guidelines are found, but these two are of major importance for large-scale settings. *[Relative operators]:* Provide operators that operate relatively on the location system. We found that is of importance for many Smart-Its applications. *[Provide tools]:* Provide tools for management, information and assembly. Tools also accelerate testing of applications.

2.4 Guidelines

This section summarizes the guidelines found in the example settings. They influenced the design of the location model, the communication model and the implementation of the system. The guidelines were:

- Support location dependent and spatial-aware communication
- Human and application understandable location description:
- Allow Off-line access of location data
- Be open to new technology
- Provide both absolute and relative operators
- Be reliable and provide maintenance tools.
- Be aware of the energy consumption

In the next two chapters, guidelines given above are revisited by explaining the location and communication model that fundaments the RAUM concept, and also by the implementation.

3 The RAUM model

In this section we will present the RAUM model for spatial-aware communication in ubiquitous computing environments. The RAUM model consists of two main parts, the location representation model (LRM) and the communication model (CM). The LRM defines how location information is represented, stored and communicated in the RAUM-system. The CM defines how this location information is used in the communication of artefacts belonging to the RAUM environment.

3.1 Representation of location information

The RAUM system is designed to provide a communication framework for use in the field of ubiquitous computing. As stated in the guidelines, integrating computer-augmented objects seamlessly into everyday life requires a communication model that fits into our lives as seamlessly as the used technology. Therefore, we tried to capture significant features of the human perception of space and communication in the RAUM model.

In the RAUM-LRM a tree representation for location information was selected (see figure 6). This location-tree consists of three general layers: A tree-root, semantic sub-layers and a position stated in three-dimensional Cartesian coordinates. Up to three semantic sub-layers as well as the position in Cartesian coordinates are optional for providing more flexibility.

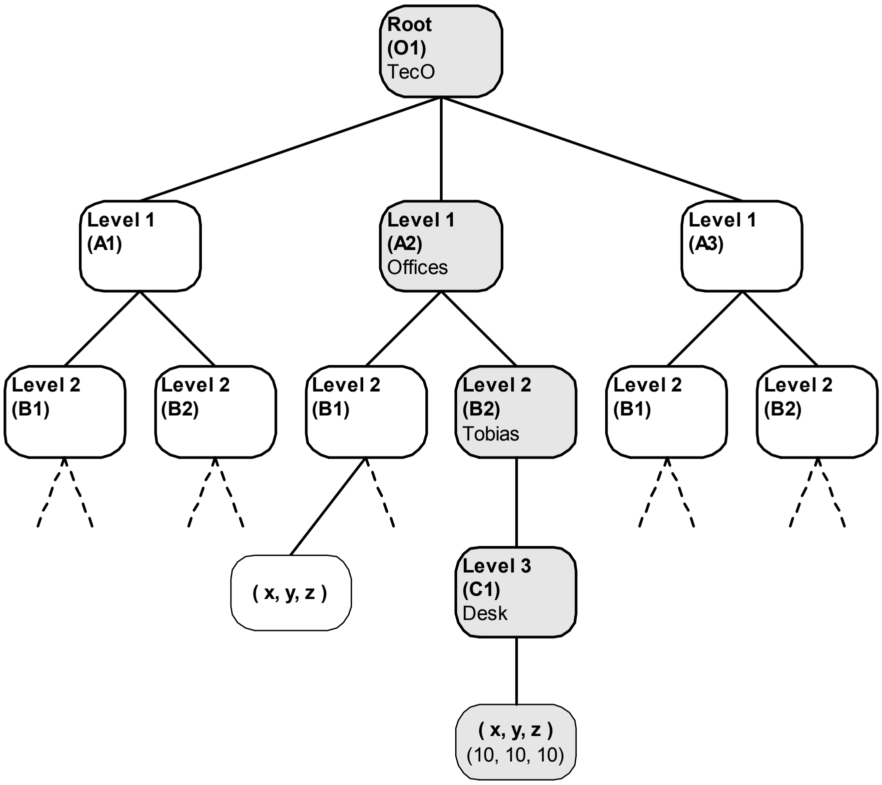

Figure 5: Example of a location tree

The tree root consists of a semantic representation of the communicational sphere of interest for the RAUM-system and a globally unique identifier. Thus the location tree is understandable for a human, yet still interpretable for direct use in an application. It contains a textual description of the communication sphere, such as the name of the company or institute running the system on its premises. This textual description is extended by a unique root-ID for providing communication links among different RAUM installations. The root-ID can consist of the IP or IPv6 address of the organisation running the system for compatibility with existing IP networks.

The location of a specific artefact is represented by a path in the location tree, starting at its root and ending in one of its leaves. An example of a location is marked in figure 5. This representation is used to store location information in artefacts.

The RAUM system makes use of distributed storage of the location information as far as all artefacts are capable of handling this information. Thus no central entity for storing and providing the complete location tree is necessary.

3.2 Structure of the locations represented by the tree

This section presents the theoretical background for the location representation in a location tree. The computer systems using the location data provided by the RAUM system have to be aware of the structure that is represented by this data. The advantage of a formal description is that operators on the system can also be expressed formally.

In the RAUM-LRM we used a tree for the representation of locations and their relation. This relation of locations is responsible for forming the tree structure of the representation system. It can be expressed as a mathematical relation as follows:

Let a, b, c be locations and A, B, C sets of locations ($A = \{a, b, ...\}$), where a set of locations is called *area* and let P be the power set of those areas. Then I is defined as a relation on P with:

(1) $\forall A \in \mathrm{P} :< A, A > \notin I$ (Irreflexive)

(2) $\forall A, B, C \in \mathrm{P} :< B, C > \in I \wedge < A, B > \in I \Rightarrow < A, C > \in I$ (Transitive)

(3) $\forall A, B \in \mathrm{P} :< B, A > \in I \Rightarrow < A, B > \notin I$ (Asymmetric)

(4) $\forall A, B, C \in \mathrm{P} :< B, A > \in I \wedge < C, A > \in I \Rightarrow B \cap C = \emptyset$ (Disjoin)

(5) $\forall A, B, C \in \mathrm{P} :< B, A > \in I \wedge < C, A > \in I \wedge \exists X \in I :< X, B > \in I \Rightarrow < X, C > \notin I$

This definition forms the hierarchical set relation I on P where (1)-(3) imply a strict partial ordering on the power set of areas. (4) Restricts the lattice of areas formed by (1)-(3) by only allowing the derivation of disjoined areas. (5) Prohibits the derivation from more than one parent. The resulting relation I can be visualized as a tree, where the right hand side of a statement $< B, A >$ is annotated one layer higher than the left hand side.

For introducing application-readable semantic information as a new feature of the LRM every node in the resulting location tree can be extended by a globally unique id that is called incarnation of the node. The nodes incarnation can be used to match human readable descriptive name of nodes to their semantic representation. For example we found that most of the applications we used for evaluating the RAUM system are interested in the rooms as a basis for communication. (We will go more into detail on the communicational spheres of interest of artefacts in the next section.) The RAUM system makes use of human readable names for nodes contained in the location tree. These names obviously depend on many factors like the used language. To be able to develop applications that can be used independent of the linguistic representation of a semantic location as a room, it became necessary to introduce a machine-readable representation of the underlying semantic of the naming of a node. This enables applications to search for semantic areas in the location tree without the need to standardise the naming of the nodes.

3.3 Structure of the RAUM communication model

The RAUM system uses spatial co-location of artefacts to determine valid communication partners by defining spatial areas called RAUMs. A RAUM is a contiguous set of geometrical positions in the real world, grouped together into a sphere of communicational interest. This real world area then is projected to the logical representation introduced in the last section.

Each artefact communicating in the RAUM system tags each transmitted data packets with a description of its actual location. This description is the equivalent to a path through the location tree from its root to a leaf. The receiver of such a packed computes, based on this location information, whether itself is contained in the same RAUM as the sender. If the two artefacts are contained in the same RAUM a data packet is passed on to higher layers of the RAUM-stack for further processing, if not the packet is discarded.

To allow a flexible handling of communication issues three types of RAUMs were introduced. In these RAUMs we differentiate between the artefact defining the RAUM and artefacts contained in RAUMs defined by other objects. Types of RAUMs are:

Listener-RAUM: In a Listener-RAUM the defining artefact can receive data form all other objects in this RAUM. Artefacts contained in a Listener-RAUM cannot communicate with each other. For example this kind of communication RAUM can be used by doorplates that are interested in events from the room they are associated with. A doorplate defining a Listener-RAUM covering the physical area of the room can receive all events produced by artefacts contained in this physical room.

Speaker-RAUM: A Speaker-RAUM can be defined for sending data packets to all contained artefacts. Communication in the reverse direction, from contained artefacts to the definer of a Speaker-RAUM or among the contained artefacts is not supported. Location beacons providing localisation information for the artefacts in an environment can use this RAUM-type. The beacon defines a Speaker-RAUM covering the physical area in which the localisation information it provides is valid.

Discussion-RAUM: The Discussion-RAUM defines the most general communication scope. Every artefact contained in a Discussion-RAUM receives all events from all other contained artefacts. This RAUM-type is used whenever peer-to-peer communication is required.

Apart from different communication types the RAUM system draws a distinction between different RAUM shapes. Valid RAUM shapes can be classified into semantic RAUMs that contain complete nodes of the location tree with all their children and geometrical shapes that are restricted to the node they are defined in.

3.4 Operators on the RAUM system

The RAUM-CM defines a set of operators used to interact with the system. In this section we will introduce these operators by a functional description. The described operators change the state of the system or provide information on the current state of the system.

Apart from the location tree the RAUM system uses two additional sets per artefact for management purposes. These sets are called its R^C and R^D. An artefact's R^D set contains all RAUMs defined by this artefact. The set called R^C consists of all RAUMs in which the specific artefact is contained.

The operators defined in the RAUM system can be assigned to one of two general groups: operators working on the location tree that are used to gain knowledge on the spatial and semantic configuration of the environment and communication operators used to implement the spatial communication features of the RAUM system. These operators partly make use of the location operators of the first group.

Location operators

Location operators work exclusively on the set- and relation-structures forming the basis for the RAUM system: the location tree, respectively its sub-trees, the R^C and R^D sets and the relations defined on these structures. **CONTAIN** operates on an artefact's R^C set. Whenever the CONTAIN operator is invoked with a RAUM as the argument, it returns whether the invoking artefact is contained in this RAUM or not. CONTAIN can also be invoked with no arguments, returning all RAUMs an artefact is contained in. Exemplary we will give the formal definition of this operator:

Let $l_{artefact}$ be the actual location of a given artefact and R the RAUM that the artefact invokes the CONTAIN operator on. The RAUM R be defined as a set of locations $LOC_R = \{a, b, c, ...\}$ where $LOC_R \subseteq I$ and $R = \{Type_R, Shape_R, LOC_R\}$. When CONTAIN is invoked all sub-sets $A, B, C, ...$ representing semantic locations in the path of $l_{artefact}$ are progressional matched against LOC_R, where this matching operation returns true if:

$$\forall a \in A, b \in B, c \in C, ... : \{a, b, c, ...\} \subseteq LOC_R$$

The **SEARCH** operator can by applied to the location tree or respectively to the path through the location tree stored in each artefact specifying its position. This special operator searches the tree or parts of it for incarnations associated with nodes. It returns this semantic information allowing semantic dependent interpretation of location information.

Communication operators

The communication operators mainly work on the R^C and R^D sets. They make use of the systematic and semantic information provided by the location operators. A fundamental attribute of all communication operators is that they initiate announcements that can influence the semantic and geometric structure of the RAUM system. The **OPEN** operator is used to initially define a RAUM. The RAUM is added to the R^D set of the defining artefact and a RAUM-OPEN announcement is sent. Artefacts receiving RAUM-OPEN announcement check whether their actual position is contained the newly opened RAUM. If an artefact is contained in a RAUM, it adds this RAUM to its R^C set, indicating the containment relation. **CLOSE** is the decomposition operator of the RAUM system. It is used to delete defined RAUMs. Only the definer of a RAUM can close it. Whenever CLOSE is invoked on a RAUM it is deleted for the R^D set of the artefact and a RAUM-CLOSE announcement is sent to the system. Artefacts receiving a RAUM-CLOSE announcement check whether this RAUM is part of their R^C set. If this check is positive the RAUM is deleted form the R^C set. The **MOVE** operator is used whenever an artefact has to indicate a change of its position. This is necessary because geometrical RAUMs are defined relatively to the position of the definer. Therefore the members of the containment relation of a RAUM may change when the position of the defining artefact is changed. To handle this situation a RAUM-MOVE announcement is sent, making all artefacts contained in the RAUM update their R^C sets. The **SEND** operator is used to transmit massages in the RAUM system. The reception of messages sent via the SEND operator is controlled by the application of the CONTAIN operator.

Example

In a simplified operational example of spatial-aware communication, a coffee machine opens a listener RAUM for all data packets inside department "A" of the organisation (figure 6, step 1). The opening announcement is communicated to all routers that are responsible for this branch of the location tree and this information is added to the R^C list of all objects in this particular branch (step 2). As soon as one object, e.g. the MediaCup, sends a message (step 3) the packet is picked up by the bridge and forwarded to the router (step 4). Here the packet is analysed based on the location information contained in the packet and – based on the R^C set – routed to the coffee machine (step 5).

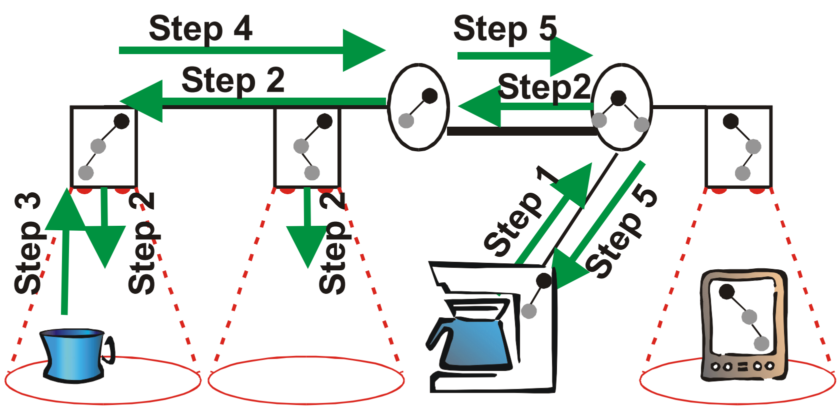

Figure 6: Simplified operational example of spatial-aware communication

4 Implementation

Providing an implementation as a proof of concept and to study further application scenarios is a widely accepted strategy in the Ubicomp research area. We developed various devices to explore the spatial-aware RAUM communication model. Most of these devices augment everyday objects, implying additional constraints for the available resources like memory, computing power and communication bandwidth. Our focus lies with a fast, small and thin implementation of the location and communication aspects of our model.

This section starts with a consideration of pre-implementation decisions, continues with implementation details of location and communication related issues and we conclude with a description of the implemented infrastructure and mobile artefacts. In the latter, a review of the guidelines stated at the beginning of this paper is done.

4.1 Pre-Implementation Decisions

To follow the vision of Ubicomp and let mundane everyday objects communicate with each other, it requires the use of very small electronics for our artefacts. The artefacts we developed are the MediaCup, the MemoClip and Smart-Its, which contain all micro controller based computing devices, implying that they have at their disposal a small amount of memory, computing power, and energy. The aspect of mobility characterized by the aforementioned devices, incurs limitations and crucial considerations with respect to energy consumption and wireless communication.

Reflecting about the limited capabilities of the devices we introduced a network infrastructure. An object-based network infrastructure, which also implements our model, was set up to provide additional resources if necessary. Furthermore, the infrastructure extends the communication range for artefacts, which are outfitted with a short-range communication device. To build up a network on top of all artefacts gives us the possibility to introduce new networking technologies without changing already existing application scenarios. It needs just an interface to connect to the network and artefacts at various locations can communicate transparently with each other via the network segments at their location.

The location tree represents the superset of all locations within a certain area or organisation and the network infrastructure in that area should represent the location tree. The floor plan design of our lab motivates the setup of the location tree used for all application scenarios we have implemented so far. The tree in figure 6 is used as a template. The root node is associated with the name of our lab “TecO”. This is the most general location description, which contains all artefacts we have implemented. Step-by-step we decompose this organisational structure into smaller units. The next level (“room level”) describes all rooms within TecO, as there are offices, kitchen, meeting room, floor, and student computer pool. Each of them is represented as a child node of the root on the first level. Describing further sub-sections of the rooms enables a finer granulation of locations. Offices are divided into workplaces of employees, the kitchen is divided into cooking and preparation areas, the meeting room is considered as presentation and auditorium area, the floor is split up into left and right hand side etc. The finest semantic level breaks up the declared sub-sections to the level of inventory objects like desks, chairs, shelves, and lockers. On this level whiteboards and overhead projectors in case of the meeting room or objects like a fridge and an oven in case of the kitchen, can be found. Beyond this semantic layers of the “TecO” location tree the geometric description resides.

The complete location tree is held only in infrastructure objects. All other artefacts in the system only hold a part of the tree that is relevant to them. E.g. most artefacts only have to store the path through the tree representing their own location. This enables us to implement most of the location tree in a distributed manner. Furthermore this way of handling the location information supports easy maintenance of parts of the tree and it is quite robust against network failures in the infrastructure implying a high reliability in view of long-term usage. Apart from that new nodes can be easily inserted into the tree without the need to propagate the changes of the environment through the whole system. This approach keeps us close to the development guidelines suggested in section 2.

In the infrastructure network we use different communication media depending on their characteristics. Data transmission via infrared links is easy to implement, the facing can be adjusted very accurately and the power consumption is quite low, but a line of sight is needed. This makes communication via infrared always a short-range directed communication. The higher energy consumption of RF based infrastructure artefacts implies their use within bridges running on a continuous power supply. They however fail in determining the accurate location of a client due to the lower resolution of RF-spots compared to the IrDA beacons. A mobile artefact can be detected within the range of the RF transmitter, which is mostly within an ordinary office room. As a result of this characteristic RF bridges are used to detect the presence of an artefact, which is useful location information. Furthermore, RF based infrastructure artefacts can communicate with other artefacts even when they are hidden in cases or integrated into objects allowing an unobtrusive integration in our lab environment.

4.2 The Location Description

As identified in the guidelines, the RAUM location model was designed to be independent from the location system and the location representation was designed to be human readable. This enables the system to rely on various location systems like GPS for outdoor use, ActiveBadge, and IrDA compliant infrared location beacons for indoor use. As semantic location information in a human readable form, in general, consumes lots of resources like memory and communication bandwidth, we limited the length of these location descriptions to 9 bytes per node. We found that this restriction applies to almost all existing naming schemes of organisations – at least with the abbreviations normally used. With the maximum tree depth of 4 levels including the root node, - this restriction may be relaxed in further versions - this approach results in up to 36 bytes for the semantic part of a location description. In a next step, a special encoding, based on a reduced alphabet was applied. This alphabet is case insensitive and reduced to characters, allowing to form a URI which enables us to store 3 characters in just 2 bytes. With respect to this compression the encoding schema is referred to as 3-to-2 encoding. The application of this encoding reduces the maximum space needed for one location description by one third. For efficient storage of the geometric coordinates we use our ab-encoding. This encoding scheme makes use of the fact that positions, lengths and time values are not perceived lineally. To humans it makes sense to describe positions on a desk in centimetre scales while on the playground distances are more likely judged by the meter. The ab-encoding scheme takes this into account by using the two most significant bits of a data byte to set a multiplier. The 6 least significant bits hold the value the multiplier is applied to. Table 1 shows the different multiplier for values that are interpreted as lengths.

ab bits	Value range
00	* 1 cm
01	* 10 cm
10	* 100 cm
11	* 1000 cm

Table 1: ab encoded ranges

The location description (figure 7) also contains a unique 4-byte identifier for the tree's root and an additional deviation byte to express the accuracy of the position itself. So all together 32 bytes are needed to form the resulting location description that is communicated with every message:

Figure 7: Location description

4.3 Communication

In our model, every artefact has to have a location description to participate in the spatial-aware communication. Every message sent or received by an artefact contains this location description. According to this description the receiving artefact can decide whether it accepts the message or ignores it. If the artefact is unable to determine its own location it can query an infrastructure object. In this case the infrastructure object provides the RAUM location information, which is then inserted into an internal data structure of the artefact. Each time the artefact changes its position it has to resend the query. To avoid permanent communication, which is crucial for a low energy consumption of the artefacts, we implemented transparent location stuffing. Location stuffing is a technique that uses infrastructure objects that receive certain data packets to fill in the necessary location information. In this case the artefact starts to send a RAUM PDU (Protocol Data Unit) and the receiving infrastructure object inserts automatically its own RAUM location description. But this technique introduces the problem of duplicated artefact PDUs containing different location descriptions. Especially when using RF-links for communication it may happen that artefact PDUs are received by various infrastructure objects and so get duplicated. In this case the infrastructure network has to be able to identify these duplicates by an additional sequence number in the artefact's PDU.

In our spatial-aware communication model an artefact has to decide whether to accept or ignore a message according to the location description included in a received PDU. This description is compared to the dimensions of the RAUMs the artefact is contained in. This management of the communication in the RAUM system is done on basis of information that is held in two lists, called R^d-List and R^c-List. Every artefact in the RAUM-system holds a list containing all RAUMs it defines, the R^d-List. This list consists of a unique identifier, the type and shape of the RAUM as well as a time to live (TTL) field. Artefacts in the RAUM-system may also implement the R^c-List, containing all RAUMs it is contained in. Table 2 summarises the entries that the R^d-Lists and R^c-Lists have to hold for managing the different RAUM-types in case of defining respectively contained artefacts. The communication operators using these lists are implemented as API calls in our operating RAUM system. Due to the straightforward implementation of these operators we will not go into detail here.

	Defining Artifact		**Participating Artifact**	
	R^c	R^d	R^c-	R^d
Speaker RAUM	X	(X)	(X)	-
Discussion RAUM	(X)	(X)	(X)	-
Listener RAUM	(X)	(X)	(X)	-

X: needed for all implementations

(X): not needed for a minimal implementation

Table 2: RAUM-lists for Artefacts

4.4 Implementation of Infrastructure and Mobile Artefacts

The devices we developed may be classified as either infrastructure artefacts or mobile artefacts. Examples of infrastructure artefacts include location beacons, bridges and RAUM routers.

Location beacons periodically send spatially limited RAUM location information consisting of the location of the beacon itself via an infrared or a RF connection. Our hardware implementation uses standard batteries to power the location beacons. Due to the minimalist design of the beacons, these batteries last for months. The range of the minimalist IR beacon implementation (figure 8) ranges from 3 to 10 meters, allowing a footprint from 1 m² to about 20 m². The IR beacon signals are IrDA standard compliant and can be used directly - without hardware changes - by PDAs, laptop computers, computer watches and some of our digital artefacts. Configuration is done only once, when a new location beacon is set up and never changed during the lifetime of the beacon, but can be changed afterwards by software. In general every location beacon defines its own RAUM in its R^d-List, resulting in only a single entry. A mobile artefact can determine its position by parsing the location description within the location beacon's PDU. The **RAUM bridges** provide access to the services of the RAUM network via infrared or RF connection. RAUM bridges act as the interconnection points for the different communication medias used in the RAUM system. They can perform location stuffing by inserting their location description into an artefact's PDU, or act as a location beacon. Bridges form the central infrastructure object in the RAUM system due to their crucial role in providing backend access for the mobile devices in the environment. **RAUM routers** build the networking backbone of the RAUM system. They have knowledge of the entire location tree and hold complete lists of all RAUMs that are defined in the parts of the system they are responsible for. To allow the routing of RAUM packets we implemented the location name service (LNS). Amongst other things, this service is responsible for mapping location descriptions to the router that is responsible for a given area. To reduce the complexity of the infrastructure one might wish to avoid the use of RAUM routers and just broadcast all information. However, this approach is only feasible if the size of the environment is fixed and does not exceed a certain threshold.

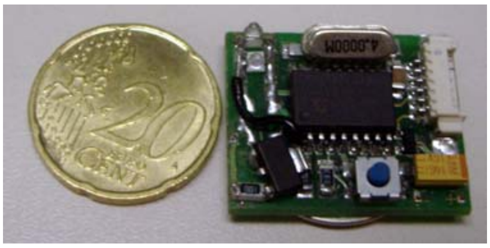

Figure 8: A very low power infrared (IrDA) based location beacon

So far, these infrastructure artefacts were built into different rooms like meeting room, kitchen, floor, two office rooms and a student computer room. They represent the structure of a location tree as described in the previous section. Apart from the infrastructure turning our lab into a "living lab"[6], we developed various mobile artefacts implementing the RAUM model to communicate in a spatial aware way.

MediaCup

The MediaCup is a common coffee cup that was augmented with sensing, computing and communication capabilities. It is able to recognise its current context, like someone is drinking or playing or the temperature of the coffee inside. The MediaCup communicates these contexts via a directed infrared link to a network of IrDA access points installed at the ceiling. Location information for the MediaCups is provided through location stuffing. The access points are connected to the RAUM routers via a CAN bus. As a consequence these access points act as bridges from the wireless IR part of the RAUM network to its wire based part. The MediaCup was the first implementation of a digital artefact equipping an everyday object like a cup with a computing device, without changing the functionality of the object itself. The MediaCup is designed for seamless integration into our everyday office lives by making the augmentation as unobtrusive as possible. For example the needed electronic components including processor, memory, sensors, rechargeable battery, infrared communication and other electronics were integrated into the bottom of the cup and recharged wireless using “MediaSaucers” [3]. The most important sensors used in the MediaCup are movement sensors that are used to detect the status of the cup -e.g. “drinking”, “playing” but also used to detect potential location changes. Other sensors detect the cups fill status and the temperature of the liquid in the cup. The processor’s task in the cup is to understand the sensor values according to the application domain "Cup" and to generate context like "drinking" out of it. It is also the processor that cares for energy e.g. by switching off parts of the system when they are not used.

Further applications in the MediaCup setting were the electronic doorplate, coffee machine and computer watch. E.g. the electronic doorplate is capable of listening for RAUM PDUs in an associated room. The doorplate receives the current context of the cups in this room and can display for instance “Meeting” if there are some cups containing hot coffee moving in the room of interest.

MemoClip

The MemoClip [4] is a portable active reminder appliance. While first implemented as proprietary embedded computing device with sound and LCD output it is now implemented as an application and add-on for a small Visor Edge PDA computer. Here the PDA calendar application is extended with a location aware component. The MemoClip is designed to remind the user of events referring not to a temporal context like the calendar but to a spatial context. The MemoClip can be used to trigger a reminder according to the location of the user. For example a forgetful user could associate the door of his apartment with a reminder not to leave his keys. Technically the MemoClip understands RAUM location data sent by an infrared location beacon or a RF location beacon. If the current location of the MemoClip, determined by the beacon’s signal, matches a predefined location the alert signal of the reminder is triggered. We implemented the infrared as well as the RF based version of the MemoClip. On one hand the resolution of our IR location system is still higher at the moment than the resolution of our RF based system, but on the other hand the IR location system is in need of a free line of sight between the location beacons and the receiver of the location information. Therefore we aim to improve the performance of the RF beacons in order to develop a RF based MemoClip that can be carried in a pocket and yet make use of the same fine-grained location resolution as the IR based version.

Smart-Its

Smart-Its [5] are a completely new kind of devices. Smart-Its are not artefacts by themselves but are very small computing, sensing and communication devices that are specially designed for post hoc augmentation of everyday objects. The idea behind this is to have a platform for developing new ubiquitous computing applications involving various artefacts without the need to build each and every augmented object needed in a scenario from scratch. Smart-Its devices are therefore designed as very small devices (figure 4) containing a set of standard sensors as two-axis acceleration, light level and frequency, audio and sound, temperature and pressure. Actuators on the Smart-Its are LEDs and sound output. Smart-Its also come with RF communication at 125 kbit/s, their own processor (5 MIPS at 20 MHz) and up to 8 kbyte of memory. Smart-Its are easily extendible through connectors with other sensors and actuators. They are programmable and configurable to match the needs for use as an attachment to an everyday object. User programs access Smart-Its functionality through a small operation system that schedules the access to the sensors and actuators, cares for saving energy and allows communicating context to other computers in an abstract way.

Smart-Its are interconnected to the remaining RAUM environment via bridges translating the RF based SPOT-net packets used in the Smart-Its communication to full RAUM packets – Smart-Its make use of the reduced location descriptions mentioned before. Due to the ad hoc character of the SPOT-net, Smart-Its make use of a specially designed implementation of the RAUM stack that is able to communicate in a RAUM compliant manner in the presence of infrastructure but can switch to a stand-alone operation mode if no infrastructure is available. In this case only relative positions of Smart-Its are used. Due to the complexity of this sub-system of the RAUM system we will omit further details in this paper.

5 Related work

ActiveBadge [7] can be seen as the first use of location in a general Ubicomp application setting. Location is derived by adding cell information of the infrared network to the data. This location information is then used to either provide automatic actions like telephone call re-routing or to visualize people's location on a computer. A similar technology and model was then used for the first Ubicomp prototype, the ParcTab [8] environment built at XeroxParc that applies the idea of an Active Map [9]. The technology here provides only a coarse grained access to position. Newer systems like the CRLs Active Bat system [10] and following developments [11] allow very precise recognition of location down to only a few centimetres. Other approaches like Microsoft's Radar [12] or University of Washington's SpotOn [13] are used to detect positions requiring only a minimal infrastructure. In the RAUM location system and the example setups presented, various types of location technologies are integrated, among them RF based and infrared based location systems. RAUM is able to express the different characteristics and can deliver a unified description, hiding the underlying technologies.

A similar approach is also implemented in other systems like CMU's Aura [14]. In contrast to RAUM, location information here is not designed for distribution among the nodes. Furthermore, operators need a considerable amount of processing power and therefore energy, and are hence not intended for execution on small nodes like those presented in our examples.
Ulf Leonhard introduces [15] another extensive model for providing a location service that can integrate location descriptions delivered by a variety of sources. It is capable of integrating semantic (here: symbolic) descriptions and geometric descriptions and allows queries on the location tree. In

contrast to this, the RAUM system follows a practical approach where the location model is developed from long-term experiences in the design, use and setup of Ubicomp environments.

An example for systems using location information for routing and delivery of data is GeoCast (RFC 2009) [16]. GeoCast uses GPS-based coordinates to broadcast information into certain areas that are described geometrically. This is similar to Location Aided Routing (LAR [17]), where the network routers are not given a pre-set position but are also aware of their own position via GPS. In contrast to RAUM only broadcast characteristics are supported with these systems. Additionally both systems do not support semantic description of places.

6 Outlook

This paper presents experiences from Ubicomp prototype setups that use location information for communicating, presenting and processing (context) information. Future applications will be in need of even more and more fine-grained information from the location system. It is planned to develop such technology and also integrate relative position detection without infrastructure in the RELATE project that starts at the end of the year [18].

Although new technology will be implemented providing finer and more ubiquitous access to location information it remains unclear how users can take advantage of all this information. In the ongoing MemoClip prototype we plan to research how users annotate location with associative information. The outcome here should be a more natural and individual view on the location.
Another aspect in this context is the use of a meta-model for our RAUM location model. The RAUM location tree describes the location structure of one organisation. From a global viewpoint this leads to a set of organisational views that are independent of each other. This fact implies the future research of structures to represent these organisational relations.

We recently started the development of tools to support maintenance of location information between different devices in the context of the Smart-Its project. Such tools will contain a central repository for storing location information of an organisation, a common view to the model with a graphical interface and an automatic consistency check service.

Another topic in the MemoClip project will be the use of location information for finding data in a large, automatically stored database. Here continuously stored audio and video data should be tagged with various context information. As location is one of the most important contexts for a human, retrieval of information from such a device will be done by associating the recorded audio or video streams with the location.

7. Acknowledgements

This work is partially funded by the Commission of the European Union as part of the research initiative “The Disappearing Computer” (contract IST-2000-25428) and by the Gottlieb Daimler- und Karl Benz-Stiftung.

8. References

[1] D. Kirsh. The Intelligent Use of Space. *Artficial Intelligence* 73, p. 31-68, 1995

[2] L.E. Holmquist, J. Wigström, J. Falk.. The Hummingbird: Mobile Support for Group Awareness. Demonstration. *ACM CSCW 98*, Seattle, WA., USA, 1998

[3] M. Beigl, H.W. Gellersen, A. Schmidt. MediaCups: Experience with Design and Use of Computer-Augmented Everyday Objects. *Computer Networks, Special Issue on Pervasive Computing*. Elsevier, Vol. 35, No. 4, p. 401-409, 2001

[4] M. Beigl: Memoclip: A location based Remembrance Applicance, *2th International Symposium on Handheld and Ubiquitous Computing (HUC2000)*, p230-234. Bristol, UK, 2000

[5] Smart-Its: http://smart-its.teco.edu. EU Initiative “The Disappearing Computer”, IST-2000-25428

[6] G.D. Abowd, D.G. Atkeson, A.F. Bobick, I.A. Essa, B. MacIntyre, E.D. Mynatt, T.E. Starner. Living laboratories: The future computing environments group at the Georgia Institute of Technology. *CHI 2000 Extended Abstracts, Conference on Human Factors in Computing Systems,* The Hague, The Netherlands, p215-216, 2000

[7] R. Want, A. Hopper, V. Falcao, J. Gibbons. The Active Badge location system. *ACM Transactions on Information Systems*. 10(1):91-102, January 1992

[8] R. Want, B. N. Schilit, N.I. Adams, R. Gold, K. Petersen, D. Goldberg, J. R Ellis, M. Weiser. The PARCTAB Ubiquitous Computing Experiment in Mobile Computing. H. F. Korth and T. Imielinski, (eds.), *Kluwer Academic Press*, 1996

[9] B. N. Schilit and M..M. Theimer. Disseminating Active Map Information to Mobile Hosts, *IEEE Network,* 8(5), p. 22-32. IEEE Computer Society, 1994

[10] A. Ward, A. Jones, A. Hoper. A new location technique for the active office, *IEEE Personal Communications*, 4(5): 42.47, 1997

[11] M. Hazas, A. Ward, A novel broadband ultrasonic location system, *Ubicomp 2002*, Gothenburg, Sweden, p 264-280, 2002

[12] P. Bahl and V. N. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. *IEEE Conference on Computer Communications (INFOCOM)*, vol. 2, p. 775-784, Tel-Aviv, Israel, 2000

[13] J. Hightower, R. Want, and G. Borriello, SpotON: An Indoor 3D Location Sensing Technology Based on RF Signal Strength, *UW CSE 00-02-02*, University of Washington, Department of Computer Science and Engineering, Seattle, WA, 2000

[14] C Jiang and P. Steenkiste. A Hybrid Location Model with a Computable Location Identifier for Ubiquitous Computing, *Ubicomp 2002*, Gothenburg, Sweden, p 246-263, 2002

[15] U. Leonhardt. *Supporting Location-Awareness in Open Distributed Systems*. Ph.D. Thesis, Dept. of Computing, Imperial College London, May 1998

[16] J.C. Navas, T. Imielinski. *GeoCast - Geographic Addressing and Routing*. Computer Science Department Rutgers University, 1997.

[17] Y. B. Ko, N. H. Vaidya. Location-Aided Routing (LAR) in Mobile Ad Hoc Networks. *ACM/IEEE International Conference on Mobile Computing and Networking MobiCom'98*, p. 66-75, 1998

[18] RELATE: http://www.teco.edu/relate. EU Initiative "The Disappearing Computer"