Sensing and Visualizing Spatial Relations of Mobile Devices

Gerd Kortuem, Christian Kray, Hans Gellersen
Computing Department, Lancaster University
InfoLab 21, South Drive, Lancaster, LA1 4WA, UK
{kortuem, kray, hwg}@comp.lancs.ac.uk

ABSTRACT
Location information can be used to enhance interaction with mobile devices. While many location systems require instrumentation of the environment, we present a system that allows devices to measure their spatial relations in a true peer-to-peer fashion. The system is based on custom sensor hardware implemented as USB dongle, and computes spatial relations in real-time. In extension of this system we propose a set of spatialized widgets for incorporation of spatial relations in the user interface. The use of these widgets is illustrated in a number of applications, showing how spatial relations can be employed to support and streamline interaction with mobile devices.

ACM Classification H5.2 [Information interfaces and presentation]: User Interfaces. - Graphical user interfaces.

General Terms Design, Human Factors

KEYWORDS: Context-aware computing, mobile computing, spatially-aware interfaces, spatial relations, location systems

INTRODUCTION
It has been widely recognized that interaction with mobile devices can be enhanced by using spatial information such as the location and orientation of the device. Location-aware services ("find the nearest Italian restaurant") have been the subject of extensive research [1, 7] and are now available on devices such as mobile phones and PDAs. A range of research efforts have also shown how indoor location and proximity information can be used to adapt and streamline interaction with mobile devices [2, 12, 26].

Less attention has been paid to using information about the spatial relations of a set of co-located mobile devices for enhancing interaction and collaboration. In this paper we discuss an extension of mobile devices designed to make information on spatial relations available to support interaction with nearby users, devices and resources. We introduce the *Relate* system and toolkit enabling mobile devices to detect co-located peers and to directly sense and model spatial relationships in a very accurate manner. On this basis we show how information concerning spatial relations can be incorporated in spatialized interfaces to support interactive and collaborative tasks of mobile users.

The Relate system is based on *Relate Dongles*, sensor nodes that can be attached to mobile computing devices such as laptops and PDAs via USB, as shown in Figure 1. The dongles are able to measure distance and angular bearing between one another in a true peer-to-peer fashion. This means that spatial relations of a set of devices can be determined wherever these become co-located, indoors or outdoors, without need for any external infrastructure or instrumentation of the environment. The sensor dongles operate over ranges of a few meters (2m for a single hop) and provide centimetre-level accuracy assuming devices are roughly co-planar (e.g. co-located on a table or work surface). Raw sensor data is processed on the mobile device to establish a dynamic peer-to-peer overlay network between co-located mobile hosts, and to compute and maintain a model of spatial relations in real time. The model provides fine-grained relative device position and orientation (quantitative information), as well as qualitative relationships such as *left of, right of, approaching,* and *moving away.*

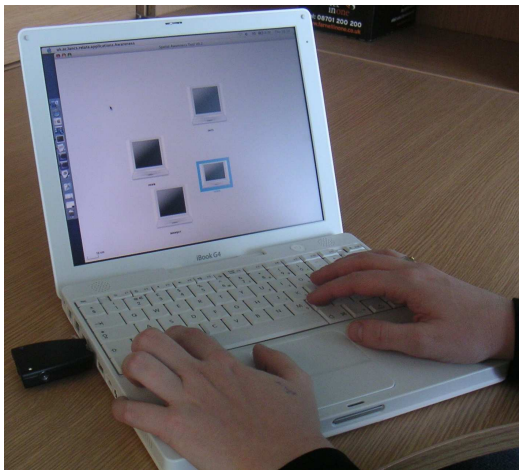

Figure 1: Relate enables mobile devices to sense relative positions of nearby peers and to use this as context in the user interface

In [15] we gave an in-depth description of the Relate sensing system and spatial model including the results of extensive

experimental testing. Using the Relate system, we discuss in this paper a new aspect, namely the construction of user interfaces that make use of information about spatial relations between mobile devices. In order to make spatial context and spatial relations available to interface designers, and ultimately to users, we have designed a set of *spatialized widgets*. These widgets provide a variety of views for spatial relations and support both explicit and implicit interaction using spatial information. The widgets are implemented in the Relate Toolkit, which provides a set of APIs for building user interfaces and applications that make use of spatial relations. To demonstrate the utility of the Relate Toolkit, we present two applications that we have built to support users in face-to-face meetings, and sketch a third one to support service browsing in mobile environments. The experience of building and exploring this first set of applications provides insights and raises open questions that we discuss in the final part of this paper.

RELATED WORK

Currently, mobile computers have a very limited world model and generally lack knowledge about the physical space in which they operate and the presence and exact location of devices. Efforts in context-aware and ubiquitous computing over the last years have focused on making knowledge about the world available to computer systems and spatial knowledge has been of particular concern. As stated by Brumitt et al. [6], the addition of basic geometric knowledge has the potential to greatly increase the shared understanding between user and system.

Two of the earliest projects in this direction were ParcTab and Sentient Computing. The ParcTab system pioneered use of location information to enable a range of context-aware applications and interfaces for small handheld computers [26]. Sentient Computing demonstrated use of fine-grained positioning technology to maintain a spatial model of an environment, which appears to reproduce the perceptions a user has of the world [2, 14]. Both systems involved elaborate infrastructure solutions for locating and identifying devices which limited their wide-spread adoption and deployment.

Location Technologies

Since then a variety of location technologies for mobile and ubiquitous computing have been developed [16]. Many of the available location technologies and systems provide information at metre- or room-level accuracy which has been shown to be useful for a wide range of mobile tasks, including discovery of device and user co-location within a certain space or area [26, 19]. However, few systems reported to date are capable of providing more fine-grained spatial information to devices and users that are already co-located, as targeted by our system. This includes systems using computer vision [6, 10] and ultrasonic ranging [27, 23].

Relate differs in important respects from this previous research. First, Relate focuses on peer-to-peer relative positioning and spatial relations rather than absolute location information; in this specific respect, our system is close to the DOLPHIN [21] and AHLoS [25] systems for localization in ad hoc networks. Second, Relate does not rely on external infrastructure but combines all required sensing in a small add-on for standard mobile computing devices. Finally, Relate has a strong focus on user interfaces and provides a complete solution for building spatialized interfaces. This is achieved by providing a combination of new sensing hardware, a run time system for spatial modeling, and a toolkit for interface and application building.

Face-to-Face Collaboration

With the advance of mobile technology, some researchers have started to recognize its potential to support face-to-face communication and collaboration. Among the early proposals are match-making and awareness technologies to provide roaming groups with a sense of connectedness [18], ad hoc games and gaming platforms that seek to make real-world group mobility part of digital entertainment [3], educational software tools [9], and messaging devices that adopt "word of mouth" metaphors for proximity-based passing of information [4]. Most of these systems and applications could benefit from accurate knowledge about the spatial arrangement of the devices involved, yet their current sense of space is limited to rough estimates of device proximities based on infrared visibility, Bluetooth device discovery, presence within a radio cell with known location, or analysis of network signal strength.

Mobile Interaction Techniques

Knowledge about spatial relations represents a new form of context for mobile devices which can be employed for novel interaction techniques. One of the first projects to consider spatial information beyond mere location was Fitzmaurice's work on spatially-aware palmtop computers [12]. Proximity as criteria for interaction was explored as part of the ParcTab system in the form of proximate selection, a technique for automatically changing interfaces so that the natural defaults reflect the user's current context [26].

Other interaction techniques for mobile devices do not make use of explicit spatial information but use spatial metaphors. Pick-and-Drop is an extension of the drag-and-drop interaction technique for multiple co-located devices [24]. Using a special stylus, it allows users to pick up an object on one computer with a stylus and drop it on another nearby computer. Danesh et al. [9] describe an interaction technique for identifying and selecting devices through a pointing gesture using custom tags and a custom stylus called the gesturePen. It allows users to select a device using a "that device there" gesture instead of navigating through traditional user interface widgets such as lists. Finally, Hinckley [17] explores synchronous gestures for dynamic display tiling, a technique enabling users to tile together the displays of multiple tablet computers just by physically bumping them into each other. Currently none of these examples make use of explicit spatial knowledge.

Toolkits

A major concern of our work in Relate is to bring information about spatial relations to the fingertips of developers and ultimately users. Related work in this respect includes the Context Toolkit that enables interface developers to treat context input like widgets [11]. We also seek to provide widgets, however specifically for interaction using spatial relations. Topiary is a prototyping tool for location-enhanced applica-

tions that like Relate provides specific abstractions for spatial information however in contrast to Relate it is focussed on support in early design stages [20]. Other work concerned with provision of spatial programming abstractions includes "programming with space" in the Sentient Computing project [2] and the Location Stack [16].

RELATE SYSTEM

The Relate system consists of new sensing hardware, a run time system for spatial modeling, and a toolkit for spatialized user interfaces:

1. *Relate Dongle,* a USB ultrasound sensing device for peer-to-peer localization. Relate dongles are designed for use with mobile computers (such as laptops or PDAs) and deliver real time measurements about their relative distance and bearing.
2. *Relate Spatial Engine,* a software system running on the mobile device to which the USB dongle is attached. The engine is responsible for computing and maintaining a high-level model of the spatial relations of co-located mobile devices.
3. *Relate Toolkit,* a set of APIs for building user interfaces that make use of spatial relations.

The dongle and the spatial engine have been described in detail in [15]. In this section we give a brief summary of these components, and also describe how the system is configured for use.

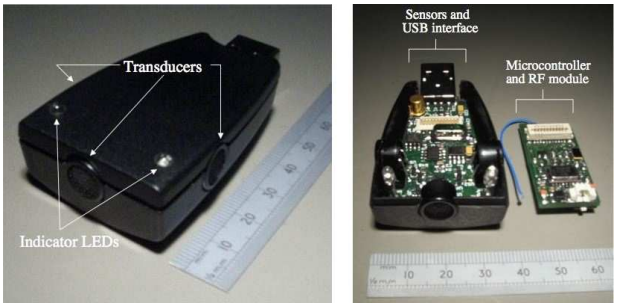

Figure 2: Relate Dongle (outside and inside view)

Relate Dongles

A Relate dongle is a wireless sensing add-on to a mobile host for which it collects data. It is built from custom hardware and uses protocols for ad hoc networking and distributed sensing.

A Relate dongle is shown in Figure 2. It is composed of a circuit board with a microcontroller and RF module, and another separate circuit board with sensors and USB interface. The dongle casing is about 5.5 x 3.5 x 1.5 cm in size, and has a standard A-type USB connector on one side. The other three sides of the dongle each have a 1 cm ultrasonic transducer, arranged to cover the space left, right and in front of the USB port on the host device.

Dongles use a distributed ultrasonic sensing algorithm for peer-to-peer localization. Each dongle performs the following tasks:

1. *Emitting ultrasonic signals:* periodically each dongle emits an ultrasound signal using all transducers simultaneously. At each point in time only one dongle is sending. This is achieved by synchronizing co-located dongles using the built-in short-range wireless network.
2. *Sensing and analyzing ultrasonic signals:* each dongle uses its three transducers to listen for incoming ultrasound signals. The receiving dongles measure the peak signal values and the times-of-flight of the ultrasonic pulses sent by the transmitting device to compute estimates for distance and angle-of-arrival.
3. *Data collection:* Dongles share sensor readings over the RF network channel, and each dongle collects and stores readings (including its own local measurements as well as those provided by other dongles) in a buffer. Periodically, each dongle uploads the sensor data from its buffer over the USB link to its host computer where it is stored and processed further.

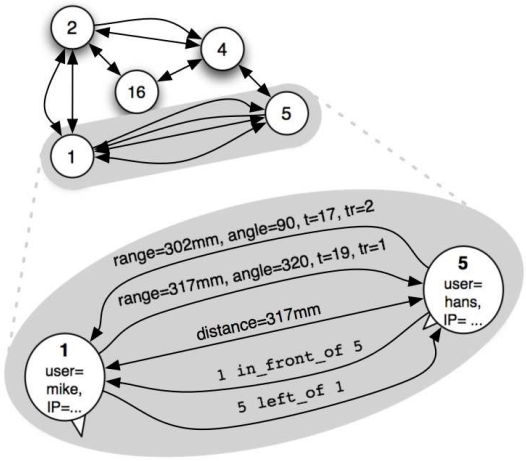

Figure 3: Spatial Relation Graph (Spatial Model)

Relate Spatial Engine and Model

The Relate Engine is a software service running on the mobile hosts. It interfaces with the dongle to receive sensor data for two purposes: (1) to compute and maintain a dynamic *spatial model* as a real time representation of the spatial arrangements of mobile devices; and (2) to establish a dynamic peer-to-peer overlay network between co-located Relate-enabled mobile devices for spatially-aware communication at the application level.

The spatial arrangement of devices is modeled in a graph structure, with nodes representing mobile devices and edges indicating spatial relations between devices (see Figure 3). A graph captures the spatial arrangement at a particular point in time, and the overall model is realized as sequence of time-stamped graphs.

Nodes and edges of spatial relation graphs are labeled with attribute-value pairs. Node attributes include the ID of the

dongle; the location of the USB port on the host device (required to map dongle positions to device positions); IP network address and hostname (enabling Relate to establish an overlay network); and user information (name, affiliation, email). On the local device, most of this information is configured when the Relate engine is first installed, using an end-user configuration tool detailed in Sect. . It then becomes shared between devices via their dongles (i.e. over the dongles' dedicated RF channel). Note that no IP connectivity is required for this.

The model captures binary spatial relations between devices as edges between nodes. Spatial relations are computed from sensor data in a staged processing pipeline. The initial stages are for association of sensor readings with pairs of nodes followed by consolidation of distal and angular measurements using non-linear regression, and yield accurate quantitative information. Further stages are for computation of qualitative spatial relationships that approximate human concepts of space [8], such as *near* and *far*, and for inference of spatio-temporal relations, such as *approaching* and *moving_away*. The computed relations are summarized in Table 1.

Table 1: Spatial Relations modelled in Relate

Relation	**Type**	**Explanation**
distance A B	Numeric $(0 \ldots \infty)$	the exact distance between A and B in mm, e. g. "930"
angle A B	Numeric $(0 \ldots 359)$	the direction from A towards B relative to the orientation of A in degrees, i. e. 0° indicates B is straight ahead of A, 90° indicates B is exactly to the right of A, etc.
left A B	Boolean	indicates that B is inside a 90° wedge to the left of A, i. e. $\pm 45°$ from the cardinal axis with respect to the current orientation of A
right A B behind A B in_front A B	Boolean	analogous to left A B
far A B	Boolean	indicates that the distance between A and B is more than 200 cm
near A B	Boolean	indicates that the distance between A and B is less than 20 cm
approaching A B	Boolean	indicates a decrease of the distance between A and B when comparing two consecutive model graphs
moving_away A B	Boolean	indicates an increase of the distance between A and B

Configuring Relate

The Relate system requires some information on the physical setup of devices and dongles in order to compute coordinates and spatial relations correctly. Figure 4 shows the control panel which provides a number of panes for system configuration:

- an *Identity* pane, where users can enter their name, the name of the device, a URL (e. g. pointing to their home page), the local IP address (optional) and the name of an image file containing a photograph or portrait.
- a *Device* pane, which is shown in Figure 4. In this pane, geometrical properties of the physical configuration are specified. This includes the width and depth of the local host, the side to which the Relate dongle is attached, the distance of the dongle to the left, back corner of the host and the type of the host (e. g. a laptop computer or a PDA). Additionally, the logical port number can be set as well.

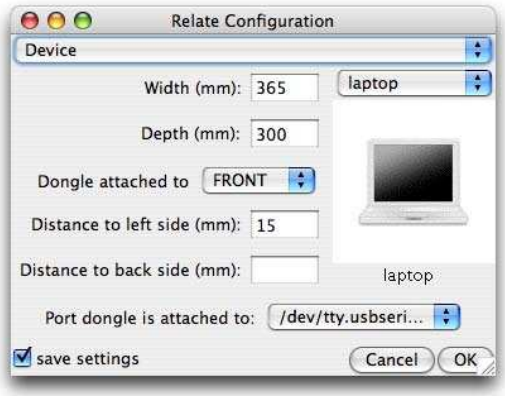

Figure 4: Control panel used to configure the Relate system

USING SPATIAL RELATIONS IN THE USER INTERFACE

Spatial relations as modeled in Relate represent a particular form of context with specific characteristics. Relate context corresponds with the immediate physical space in which it is used. All entities described in the Relate model, i. e. devices as well as users, services and resources associated with these devices, are immediately accessible for the user in "real-world" space. The spatial relations captured in the model place these entities in a context that users have directly "before their eyes." We believe this can be exploited to ease access to and interaction with nearby users, devices and resources in a variety of ways.

Context in general can be used in various ways to dynamically adapt user interfaces of mobile devices. Partially inspired by [26] and [5], we can distinguish three types of *context-aware interface adaptation:*

1. *Context-aware triggering:* automatic execution of actions depending on context
2. *Context-aware actions:* changing the result of a command depending on context
3. *Context-aware presentation:* presentation of information depending on context

In Relate we seek to support all three types of interface adaptation using spatial relations as specific type of context. Context-aware triggering is supported by an event service that allows applications to subscribe to events in the Relate system. These can be model events (updates of the model maintained by the Relate Engine), spatial events (changes in spatial arrangement of devices) and network events (changes in the Relate network such as the discovery of a new device). Alternatively, a service is available to query context information in the Relate model, and to use context query results to trigger certain actions.

Context-aware actions (the second type of interface adaptation) are supported by Relate specifically for communication actions. A spatial communication service provides a number of spatially-aware communication primitives that can be used to disseminate data in a Relate overlay network depending on their spatial relations. The event, query and spatial communication services have been described further in [15]. In this paper we focus now on the third type of context use listed above, i.e. on presentation of information in mobile user interfaces depending on spatial relations as context.

A classic example of spatially-aware presentation is *proximate selection,* a technique for automatically changing interfaces so that the natural defaults reflect the user's current context [26]. Proximate selection can be implemented by highlighting or sorting interface objects (e.g., icons) according to the physical distance of the entity they represent (i.e. device, service, or user). Clearly, knowledge about the spatial relations between devices is necessary for proximate selection. Another example of spatially-aware presentation can be found in the area of awareness support for meetings [13]. Spatial context delivered by Relate can be used to inform a user about the identities of meeting participants and their seating arrangement. This can be implemented by presenting an overhead view of a meeting situation depicting the location and owner names of all laptop computers in a room. In this case spatial context is used in an *explicit* manner (the user is directly presented with spatial information), while proximate selection makes *implicit* use of spatial context.

While it is not too difficult to come up with potential uses for spatial context and spatial relations, the construction of these applications and user interfaces is rather complicated. Spatial information must be acquired (i.e. sensed), processed, abstracted and presented. The first three tasks are handled by the Relate dongle and software engine. In order to make spatial context and spatial relations available to interface designers, we propose a set of *spatialized widgets.*

Spatialized Widgets

A spatialized widget is a visual interaction element whose appearance and behaviour adapt based on spatial context information as represented in the Relate spatial model. It provides a visual representation of spatial relations between devices, services and people and supports spatial interaction techniques. Each widget can be tailored to accommodate the requirements of specific applications.

We distinguish between explicit and implicit spatial widgets. *Explicit spatial widgets* visualize spatial relations; *implicit spatial widgets* make use of relations in a way that is not immediately obvious to a user. In the following we describe one implicit (spatialized list) and three explicit spatialized widgets (spatialized map, compass, and label).

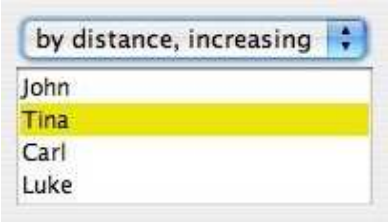

Figure 5: A spatialized list widget

Spatialized List The spatialized list is a listbox whose entries are sorted dynamically with respect to a spatial relation. Each entry in the listbox is associated with a particular relate device. The place of each entry in the list is determined by the current value of the relation. As an example, let's assume we want to display the names of the owners of nearby mobile devices sorted according the current distance to the respective device. Each entry of the spatialized list is a pair consisting of a textual label (the owner's name) and a Relate device ID. As spatial relation we use distance. The widget can than be defined as follows:

$$\begin{aligned} \text{Relation} &= distance \\ \text{Entries} &= \{(Tina, 1), (Luke, 2)(John, 3), (Carl, 4)\} \end{aligned}$$

The spatialized list sorts the names according to the current distance from the local device to any of the four devices. Assuming certain distances, the names might be sorted in the order John, Tina, Carl, Luke as shown in Figure 5 for illustration. However, the list is dynamically updated according to the actual distances stored in the Relate model. As computers are moved, the list order may change.

Spatialized Map Our second widget is a spatialized map that directly visualizes the positions of entities in the Relate model in a two-dimensional view using a relative coordinate system with the local device at the origin. The spatial arrangement of the device icons is a to-scale representation of the actual device arrangement. This allows a user to easily map between reality and map display.

The Map widget not only visualizes entities and their location, but also supports a set of interaction primitives:

- *Selection:* users can select one or more of the depicted entities. This can be used to specify the target of a command, such as pinging a computer or opening a pop-up window with more information on the selected entity.
- *Drag-and-drop:* users can drag-and-drop interface objects such as files onto entities. This can be used to transfer information to a spatially-selected Relate device.

The widget presentation can be influenced using a number of parameters. This includes filters for entities and relationships to be included in the views, various view options (textual vs. iconic etc.), and definition of the origin for the relative coordinate system. Figure 6 shows a map in which each Relate

device is represented by an icon. The local device is depicted in the lower right corner of the widget.

Figure 6: A spatialized map widget

Spatialized Compass In some circumstances it may not be useful or necessary to provide a realistic view of device arrangement, but rather to use a more abstract representation that highlights important relations while ignoring or deemphasizing others. For this purpose, we devised the Compass widget (cf. Figure 7). The compass places the local device in the center of a circle. Nearby devices are depicted along the circle in the direction in which they appear from the local device. Thus a compass uses angular relationships between the local and nearby devices but ignores distances. The compass widget is useful, for example, for indicating the direction in which a service can be found. Similarly to the map widget, the compass widget supports selection and drag-and-drop operations and can be tailored using filters and views.

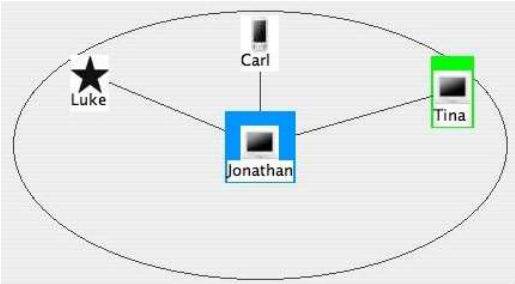

Figure 7: A spatialized compass widget

Spatialized Label All widgets so far have used a graphical representation of spatial relations. However, a textual representation can be useful as well, for instance for use on small screens, or in toolbars. The spatialized label fulfills this role. It is a simple text label whose text value verbalizes the state of one or more relations from the viewpoint of the local device. Figure 8 shows two types of spatialized label: the first one lists the value of selected relations for a specified device, the second one lists all devices that stand in a specific relation to the local device.

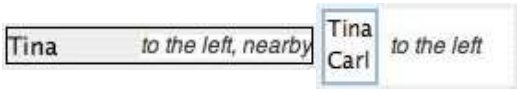

Figure 8: Two types of spatialized label widgets

RELATE TOOLKIT

The Relate Toolkit enables developers to build graphical user interfaces that make use of spatial relations. It is written in Java, currently requires Java 2 Standard Edition version 1.4.2 and runs on Windows, Mac OS X and Linux. It consists of two major components:

1. A set of Java classes implementing spatialized widgets
2. A programming interface for direct access to the spatial model and asynchronous event notification

The Relate toolkit implements a number of widget classes that extend standard Swing components, e. g. `JComponent` to facilitate easy integration of spatialized widgets with conventional user interface elements. In addition, every spatialized widget implements a common interface RelateView, which defines the following methods:

- `update(Model m)` instructs a widget to update its appearance according to the information contained in the model passed as a parameter.
- `autoUpdate(long interval, boolean enable)` configures a widget so that it will automatically update every `interval` milliseconds (by retrieving the current model from the Relate engine). The `enable` parameter specifies whether to turn automatic updates on or off.

While the former method provides developers with a means to take full control of what is being displayed at any time, the latter one can keep the information being displayed up-to-date without requiring any intervention from the main application.

In addition every spatial widget has a small set of individual methods to customize its appearance (see below). Most widgets make use of a `Mapping` object that maps displayed items (such as textual or graphical representations) to Relate devices and vice versa.

For example, the spatialized map widget is implemented in a `RelateMap` class. It extends `JPanel` and provides support for drag-and-drop operations via the standard methods defined in `java.awt.dnd`. Individual items can be highlighted using the `setMarkedDevs(Vector devs)` method. Via the `setRenderingParameters` method, the developer can customize the way in which `RelateMap` renders the items being displayed. Using this method, it is possible to specify whether or not to use a fixed scale for the relative coordinate system and to set a variety of other view options.

Applications

The first set of applications that we have built using Relate target support for co-located users and are motivated by common challenges we observe. The first one addresses meetings in which participants are not very familiar with each other and can benefit from awareness support to match faces to names and affiliations. The second one addresses the problem of transferring a document from one computer to another, which remains cumbersome despite advances in spontaneous networking. In addition to these we sketch a third application targeting support for mobile users to browse nearby services.

Spatial Awareness Support for Meetings Face-to-face meetings are still one of the most effective ways of working together. Social awareness is one of key factors for collaboration, i.e. an understanding of who the collaborators are, what their affiliation is, and what they do. It is a common experience to join a meeting and to not be familiar with the names and affiliations of other participants. With this in mind we have designed an awareness tool that provides an overhead view of the meeting situation depicting the seating arrangement of people, approximated by arrangement of their mobile computers (cf. Figure 9). The awareness tool has the same purpose as ordinary name tags or place tags: it allows meeting participants to identify each other and to address each other by name without having to remember each others names. Contrary to printed tags, however, the awareness tool works without preparation whenever and wherever people meet as long as participants bring along a laptop computer enabled with Relate.

The awareness tool is implemented using one map widget and two spatialized labels. The map displays the location of each computer in a room. Each computer is represented by an icon indicating the computer's owner name and affiliation. The two labels sit under the map and state the names of the persons sitting immediately to the left and right of the local user, assuming awareness for the immediate neighbors is particularly relevant. Furthermore there is a menu item in the view menu, which allows the user to set the origin of the coordinate system to determine where the local user icon should be displayed. The default setting is 'bottom middle' (this choice assumes that all people in the meeting sit around a table so that nobody sits behind another person).

Spatial File Transfer Copying a file from one mobile computer to another is a frequent but often annoying task users are faced with. Despite the fact that most mobile computers are equipped with wireless network technology, people oftentimes resort to using USB memory sticks. The difficulty of transferring a file via a wireless connection is not a technical issue, but a result of the often-cumbersome user interface associated with wireless solutions. In order to copy a file over a wireless link from one computer to the next, a user must perform a number of activities including:

1. identifying the name of the target computer (this may be an IP address or hostname)
2. finding the target computer in the network (this may involve searching for the remote machine using a network browser, selecting a computer from list or typing in the machine name)
3. establishing a connection to the target computer
4. initiating the file transfer

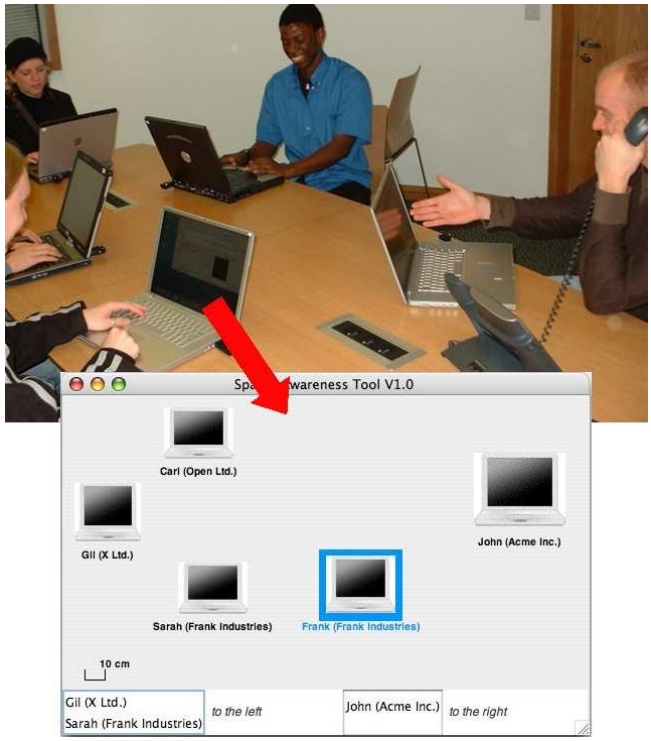

Figure 9: A spatialized awareness tool

Performing these tasks is non-trivial, especially for inexperienced users. This is despite the existence of dedicated FTP tools and the built-in file-sharing capabilities of today's operating systems.

Spatialized user interfaces offer a route to streamline the file transfer task between co-located computers, in particular by simplifying the process of finding and selecting the target computer. We designed a spatial file transfer tool that allows two or more laptop users to copy files among wireless computers using a simple drag-and-drop operation. The current implementation only works for computers connected to a wireless LAN, but the spatially-aware user interface is generic and can be used for other wireless technologies such as Bluetooth and IRDA.

The user interface is shown in Figure 10. It combines a traditional file browser for selection of a source file with a compass view of nearby computers for selection of the target. Files transfer is achieved by simply dragging the associated icon from the file browser onto the icon representing a target computer. This means that users do not have to know computer names and IP addresses. Instead they can identify the desired computer on their screen by mapping what they see on the screen with the reality in front of them.

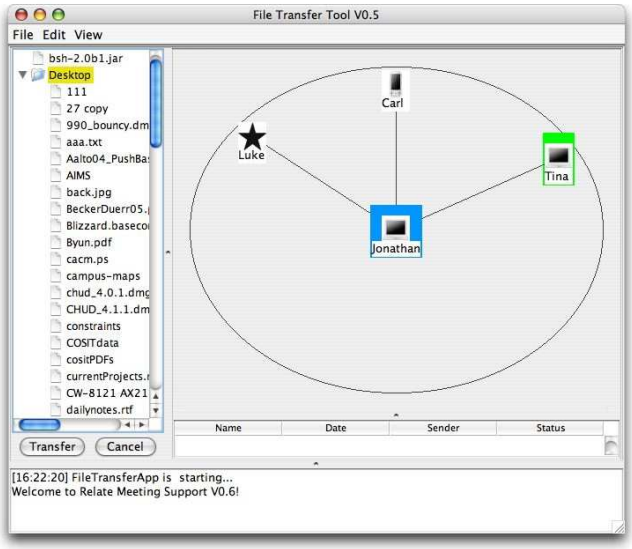

Figure 10: Spatial File Transfer combines a compass view of nearby devices with a standard file browser

Spatially-aware Service Browsing The above applications are targeted at mobile users in meetings. We also envision Relate applications that are targeted and users on the move. One of the challenges that users on the move face is the discovery of services available in their current environment. While there have been proposals to improve service discovery in an ubiquitous computing environment (e. g. [22]), the process is commonly based on network topology rather than real-world space. This can lead to discovery of services well beyond the immediate interaction space of the user. Spatial context as modeled in Relate can be useful to filter services beyond the immediate interaction range, and to list services sorted by spatial criteria. Figure 11 illustrates a possible interface using a spatialized list widget. Note this application has not been explored further at this stage, as it depends on adaptation of the Relate sensor hardware for embedded operation in the environment.

Figure 11: Design of a service browser using a spatialized list

DISCUSSION

The Relate system has been tested and used in configurations involving between 3 and 5 mobile devices augmented with Relate dongles. The devices used were: two Acer tablet PCs running Windows XP, a Dell laptop running Linux, and two Apple PowerBooks running Mac OS X. In a first set of experiments, carried out over a period of several weeks, each laptop was placed at a randomly generated location and orientation on a 2.4×1.6 m surface in an indoor office environment. Many of the randomly generated configurations involved restricted line-of-sight (LOS) between devices. The primary purpose of the experiments was to collect sensor data for analysis of location and orientation accuracy. In addition, a map view was shown on all laptops throughout the experiments, which allowed us to observe over a long period how well the visualization of spatial relations corresponded with reality.

Following the initial experimentation, we have begun to demonstrate the system and the initial set of applications to users to collect informal feedback. This initial use, however limited, has already provided a range of interesting insights. We report here our lessons learned to date and discuss some of the open questions we face.

Lessons Learned

The accuracy of Relate technology is good enough to implement spatialized interfaces—with some limitations. By accuracy we mean the ability of the Relate technology to correctly determine distances, angles and spatial relations. Experimental results have shown that the sensor and model layers provide relative location and orientation estimates at an accuracy and update rate appropriate for the scenarios we envision; the 90% accuracy is about 8 cm and 25°, and up-to-date estimates can be produced several seconds after a device has been moved [15]. The experiments have also shown that the system is able to compensate for partial line-of-sight problems when a sufficient number of devices is present. The limitations of the current implementation are:

- The more devices involved the better the accuracy. Accuracy markedly improves with three or more devices. But the more devices involved, the lower the update rate.
- All devices should be oriented in a 2D plane. The current algorithms are tailored for the 2D case, although we are working on extending Relate to 3D.

Jitter is bad. Computing spatial relations from sensor data is non-trivial. One of the major problems is that quality of measurements can vary tremendously over time. Obstructions, device movement and environmental conditions have a big impact on the accuracy. Although the algorithms used by Relate filter out spurious and faulty measurements to some extent, we found that jitter (temporal fluctuations) poses a problem on the interface level. In particular we discovered that users are able to perceive even small amounts of jitter. For example, users find it very distracting if device icons move when the corresponding real world objects do not.

Relative accuracy is more important than absolute accuracy. Relate is capable of delivering highly accurate spatial information as demonstrated by our experiments. However, we discovered that on the interface level absolute accuracy is less important than relative accuracy. For example, whether a distance between computers that are 1m apart is accurately measured as 1m is less important than the fact that two distances that in reality are equal are actually measured to be equal. Users seem to be very sensitive about proportionality and small relative errors are perceived immediately.

Device representation matters. Originally we assumed that the representation of devices in the map and compass view is of minor importance. Yet contrary to our expectation we found that size and orientation of the icons matters a lot to users. A view becomes much more believable if the icon size is proportional to the distances visualized in a view. In some extreme cases it appeared that wrongly proportioned device icons made it impossible for users to map between computer screen and reality. Icon orientation, although important, seemed of lesser importance than size.

Open Questions

How important is realism for understanding and usability? We do not have any data to decide whether realism in the information presentation improves the usability of a spatialized interface. Out of the four widgets we designed, the map uses a realistic representation, the compass shows some realism while the two others use abstract representations. We need to investigate the particular advantages of different types of representation and to identify classes of applications suited for each.

How can we visualize qualitative relations? With the exception of the spatialized label, the current set of spatialized widgets is tailored toward presenting quantitative relations. Qualitative relations, in contrast, are very useful for spatially-aware triggering of actions and spatially-aware commands. It is an open question how qualitative relations can be visualized by widgets and how these widgets should look.

Are consistent views important? Currently, each Relate-enabled computer computes its own spatial model. Although on the dongle level this involves distributed cooperative sensing, it is not guaranteed and in fact unlikely that the models computed by two hosts are identical. Although the discrepancies are small, the effect is that the view presented to users of different devices is slightly different. As of now we do not know if this causes usability problems or if collaboration between users is impaired. Technically it is possible to ensure that every computer uses the same globally consistent model, but we do not know if consistency is required and how strong the consistency needs to be.

How important is the update rate? In theory Relate is able to compute spatial information in near real time. In practice however, there is a trade-off between update rate and accuracy: the higher the update rate the lower the accuracy. This affects the way a moving object can be tracked and visualized. Currently we use a slow update rate in favor of accuracy but we do not really know what a good trade-off is. We suspect that update requirements are highly dependent on the application.

What are the scalability requirements for Relate? By scalability we mean the number of devices and the area that can be covered. Currently, the Relate dongles have a sensor range of approximately 2 m. This means that two Relate-enabled devices are able to determine their respective relative position if they are closer than 2 m. However, through measurement sharing it is possible to reliably determine spatial relations between devices that are farther apart than 2m as long as there are devices located in between the two. In effect, the Relate dongles and host devices form a multi-hop sensor network in which nodes share and collaboratively interpret measurements. Our current implementation is tuned for scenarios in which people and devices cluster close together. Through multi-hop measurements it is possible to extend the spatial range of the Relate technology, but it is not clear what range and scale of device network to assume.

CONCLUSION

The Relate system extends mobile computing devices with the ability to directly establish their spatial relationships when they become co-located. In this paper we have considered incorporation of such spatial relations in the user interface. We have proposed a set of spatialized widgets that provide different views of spatial relations, and demonstrated their application. Although the use of the Relate system and toolkit has been limited, they allow us to draw conclusions on which we can build further.

First of all, the exploration of applications shows that spatial relations can streamline interaction and collaborative tasks. Although not formally evaluated, it is apparent that spatial reference can serve as a shortcut in interaction, as for example demonstrated for file transfer in a face-to-face setting. Secondly, our initial experience has very much highlighted that what is measured as small and spurious error at the sensor level, can be perceived as a large problem at the interface level. While tolerance for general inaccuracies is relatively high, it is very low for jitter in the presentation of spatial information. Thirdly, we find general advantages of our positioning technology confirmed in our initial use. These are specifically the level of relative positioning accuracy achieved without instrumentation of the environment, and the simple deployment achieved through the use of USB dongles and minimal setup procedures.

ACKNOWLEDGMENTS

The authors would like to thank Mike Hazas who investigated the sensor system underlying the presented work. This research was supported by the Engineering and Physical Sciences Research Council (grant GR/S77097/01), and by the Commission of the European Union (contract 013790, "RELATE").

REFERENCES

1. G. Abowd, C. Atkeson, J. Hong, S. Long, R. Kooper, and M. Pinkerton. Cyberguide: A mobile context-aware tour guide. *ACM Wireless Networks*, 3(5):421–433, 1997.

2. Mike Addlesee, Rupert Curwen, Steve Hodges, Joe Newman, Pete Steggles, Andy Ward, and Andy Hopper. Implementing a sentient computing system. *IEEE Computer*, 34(8):50–56, August 2001.

3. S. Björk, J. Falk, R. Hansson, and P. Ljungstrand. Pirates! - using the physical world as a game board. In *Proceedings Interact 2001, IFIP TC.13 Conference on Human-Computer Interaction*, Tokyo, Japan, July 9-13 2001.

4. R. Borovoy, B. Silverman, T. Gorton, J. Klann, M. Notowidigdo, B. Knep, and M. Resnick. Folk comput-

ing: Revisiting oral tradition as a scaffold for co-present communities. In *Proceedings of the CHI 2001 Conference on Human Factors in Computing Systems*, pages 466–473, New York, USA, 2001. ACM Press.

5. P. Brown, W. Burleston, M. Lamming, O. Rahlff, G. Romano, J. Scholtz, and D. Snowdon. Context-awareness: Some compelling applications. In *Proceedings the CH12000 Workshop on The What, Who, Where, When, Why and How of Context-Awareness*, April 2000.

6. B. Brumitt, J. Krumm, B. Meyers, and S. Shafer. Ubiquitous computing and the role of geometry. *IEEE Personal Communications*, pages 41–43, October 2000.

7. K. Cheverst, N. Davies, A. Friday, and Ch. Efstratiou. Developing a context-aware electronic tourist guide: Some issues and experiences. In *Proceedings of CHI 2000*, pages 17–24. ACM Press, 2000.

8. Eliseo Clementini, Paolino Di Felice, and Daniel Hernández. Qualitative representation of positional information. *Artificial Intelligence*, 95(2):317–356, 1997.

9. A. Danesh, K. Inkpen, F. Lau, K. Shu, and K. Booth. Geney: Designing a collaborative activity for the palm handheld computer. In *Proceedings of CHI, Conference on Human Factors in Computing Systems (CHI 2001)*, pages 388–395, Seattle, USA, April 2001. ACM Press.

10. D. Lopez de Ipina, P. Mendonca, and A. Hopper. Trip: A low-cost vision-based location system for ubiquitous computing. *Personal and Ubiquitous Computing*, 6(3):206–219, May 2002.

11. A. K. Dey and G. D. Abowd. The context toolkit: Aiding the development of context-aware applications. In *Workshop on Software Engineering for Wearable and Pervasive Computing*, Limerick, Ireland, June 2000.

12. George W. Fitzmaurice. Situated information spaces and spatially aware palmtop computers. *Communications of the ACM*, 36(7):39–49, July 1993.

13. C. Gutwin and S. Greenberg. A descriptive framework of workspace awareness for real-time groupware. *Journal of Computer-Supported Cooperative Work*, pages 411–446, 2002.

14. A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster. The anatomy of a context-aware application. In *Mobile Computing and Networking*, pages 59–68. ACM Press, 1999.

15. Mike Hazas, Christian Kray, Hans Gellersen, Henoc Agbota, Gerd Kortuem, and Albert Krohn. A relative positioning system for co-located mobile devices. In *Proceedings of MobiSys 2005: Third International Conference on Mobile Systems, Applications, and Services*, pages 177–190, Seattle, USA, June 2005.

16. Jeffrey Hightower and Gaetano Borriello. Location systems for ubiquitous computing. *IEEE Computer*, 34(8):57–66, August 2001.

17. K. Hinckley. Synchronous gestures for multiple users and computers. In *Proceedings of UIST 2003*, pages 149–158. ACM Press, 2003.

18. L.E. Holmquist, J. Falk, and J. Wigström. Supporting group collaboration with inter-personal awareness devices. *Journal of Personal Technologies*, 3(1–2), 1999.

19. John Krumm and Ken Hinckley. The NearMe wireless proximity server. In *Proceedings of Ubicomp: Ubiquitous Computing*, pages 283–300, Nottingham, UK, September 2004. Springer.

20. Yang Li, Jason I. Hong, and James A. Landay. Topiary: A tool for prototyping location-enhanced applications. In *Symposium on User Interface Software and Technology - UIST 2004*, pages 217–226, Santa Fe, New Mexico, USA, 2004. ACM Press.

21. Masateru Minami, Yasuhiro Fukuju, Kazuki Hirasawa, Shigeaki Yokoyama, Moriyuki Mizumachi, Hiroyuki Morikawa, and Tomonori Aoyama. DOLPHIN: a practical approach for implementing a fully distributed indoor ultrasonic positioning system. In *Proceedings of Ubicomp: Ubiquitous Computing*, pages 347–365, Nottingham, UK, September 2004. Springer.

22. Mark W. Newman, Jana Z. Sedivy, Christine M. Neuwirth, W. Keith Edwards, Jason I. Hong, Shahram Izadi, Karen Marcelo, and Trevor F Smith. Designing for serendipity: Supporting end-user configuration of ubiquitous computing environment. In *Proceedings of DIS '02*, pages 147–155, London, England, 2002. ACM Press.

23. Nissanka B. Priyantha, Allen K. L. Miu, Hari Balakrishnan, and Seth Teller. The Cricket Compass for context-aware mobile applications. In *Proceedings of the Seventh International Conference on Mobile Computing and Networking (MobiCom)*, pages 1–14, Rome, Italy, July 2001.

24. Jun Rekimoto. Pick-and-drop: a direct manipulation technique for multiple computer environments. In *Proceedings of the 10th annual ACM symposium on User interface software and technology*, pages 31–39, Banff, Alberta, Canada, October 1997. ACM Press.

25. A. Savvides, C.C. Han, and M. B. Srivastava. Dynamic fine grained localization in ad-hoc sensor networks. In *Proceedings of the Fifth International Conference on Mobile Computing and Networking (Mobicom)*, pages 166–179, Rome, Italy, July 2001.

26. Bill N. Schilit, Norman I. Adams, and Roy Want. Context-aware computing applications. In *Proceedings of the Workshop on Mobile Computing Systems and Applications (WMCSA)*, pages 85–90, Santa Cruz, CA, USA, December 1994. IEEE Computer Society.

27. Andy Ward, Alan Jones, and Andy Hopper. A new location technique for the active office. *IEEE Personal Communications*, 4(5):42–47, October 1997.