

Models of Space in a Mixed-Reality System

Anthony Steed¹, Ian MacColl², Cliff Randell³, Barry Brown⁴, Matthew Chalmers⁴
Chris Greenhalgh⁵

¹ Department of Computer Science, University College London, London, UK, A.Steed@cs.ucl.ac.uk

² School of Information Technology & Electrical Engineering, The University of Queensland, Brisbane, Australia, ianm@itee.uq.edu.au

³ Department of Computer Science, University of Bristol, Bristol, UK, cliff@compsci.bristol.ac.uk

⁴ Dept. of Computing Science, University of Glasgow, Glasgow, UK, {barry, matthew}@dcs.gla.ac.uk

⁵ Department of Computer Science, University of Nottingham, Nottingham, UK, cmg@cs.nott.ac.uk

Abstract

In this paper we discuss the use of models of space in the building of mixed-reality systems. By model of space we mean a geometric or symbolic description associated with a physical space. We outline several types of model that exist, how they are surveyed and authored, how they are represented to the users and how they are supported by middleware and sensors. We show that systems often contain numerous models of space and we discuss the issues in maintaining or reifying assumptions about transformations between models.

We illustrate these ideas by describing the implementation of a collaborative mixed-reality system that allows users to experience a museum in three modalities: physically co-located visitor with personal digital assistant guide, virtual reality visitor and web visitor.

1. Introduction

Many mobile, ubiquitous or mixed-reality systems embody some form of model of physical space e.g. [1][7][9][18][22][23]. The model of space is usually used to describe some sort of application semantics such as “enable X when device Y enters zone Z”.

What is evident however from studying real systems is that they rarely involve just a single model of space. Not only is it common for application programmers to convert between different models of space because of convenience of expression (e.g. from GPS coordinates to map coordinates), but they make different services available using different models. Most commonly, the model in which the application logic lies (e.g. a proximity search in a vector map), is not necessarily the same as the model that is used to present current context to the user (e.g. a raster map). Indeed this type of application is likely to be implemented as set of distributed services.

Creating such applications requires the coordinated use of multiple models of space. This potentially involves transformation between quite different data domains and these transformations are often complex, sometimes ill-defined and may vary over time.

The purpose of this paper is to elicit the problems of working with spatial models by uncovering the configuration work done and the assumptions made when building a co-visiting system that allows three visitors to access a design gallery. In describing this system we uncover issues that are rarely addressed in this field such as long-term maintenance, uncertainty, authorship and verification.

In this paper we first describe the various types of models of space that are encountered in ubiquitous systems. In Section 3 we will then describe a demonstration application and system. The following sections will then analyze the models of space in the application (Section 4), how these models depend on each other (Section 5) and how the models are authored and maintained (Section 6). In the following section discuss the strengths and weaknesses of the approaches we have used (Section 7). Finally we discuss requirements for future work in the area (Section 8).

2. Spatial models and services

Leonhardt [13] gives a detailed account of how an application can describe space in geometric or symbolic terms. A geometric model requires the definition of a coordinate system with an origin and major axes. Once a coordinate system is defined, *location* can be described in terms of regions in 2D coordinate spaces or volumes in 3D coordinate spaces. At any instant a sensing device may report a *position* in the coordinate system, and typically this position will be compared against the 2D or 3D regions in order to determine the user’s location. A symbolic model dispenses with geometric comparisons in a coordinate system and models location solely by symbolic names. A sensing device such as a radio-frequency ID tag may report that a user is within a location or not within a location, but there is no representation as a 2D or 3D position, and thus no distance metrics and no transitive distance relations.

Many real systems contain elements of both geometric and symbolic descriptions of space. Leonhardt calls these hybrid models [13]. Jiang and Steenkiste describe a hybrid system for an indoor location system [10]. Their model uses a symbolic location for gross descriptions of



Figure 1 Early prototype of the system showing the web user view (Dub) on the left-hand machine, a desktop version of the VR user view (Ana) on the right-hand machine and a user carrying the PDA (Vee).

space at building and room level, and then a geometric description for intra-room locations and positions.

Dix, et al. [8], point out many properties that can be expected from a location reporting system. For example, a symbolic location often remains fixed for relatively long periods of time as a corresponding measurable geometric position changes frequently. It makes sense for location to remain constant for a period of seconds to minutes if location is to be a key determinant of a user's context in a context-sensitive application.

Note that the qualities of position and location error are very different. The following properties that might be associated with any particular geometric position report are harder to define when talking about symbolic locations:

- *Accuracy* – either a static, device specific statement of likely variation of report from true position (often given as ranges), or, occasionally a dynamic estimate given actual situation of device (e.g. with GPS).
- *Timeliness* – an estimate of how long ago the report was made. Often it is known how often a device should report position, but occasionally devices only report significant changes.
- *Resolution* – a usually static number that states how small a change in actual position is detectable by the device.
- *Registration* – a measure of the accuracy of transformation between this model and another model or some ground truth.

With a symbolic location, we might prefer to associate a *confidence value*, a probability that the reported location is correct. We could then represent location in a fuzzy manner.

What will be important for later discussion is that real systems often involve several models, where some or all

of the above properties are ignored, or are estimated and not validated. We will see that validating the models through calibration can be extremely difficult.

We thus take a model of space as defining a domain within which explicit interaction or reasoning over the positions and locations of multiple objects can take place. The role of a *spatial service* is to transform between the domains of two models of space. The transformation could be of several types, from affine, as is the situation in simple transformations between two Cartesian spaces of equal dimension, to discretisations of space such as conversions of tracked positions into symbolic locations.

3. City project scenario

The City project has been working in the Mackintosh Interpretation Centre located in the Lighthouse Centre, Glasgow [11]. The Interpretation Centre explores the life and work of the architect and designer Charles Rennie Mackintosh. Our design scenario involves three users, Dub, Ana and Vee sharing a visit to the centre. One of the users is in the physical center but the other two are remote. The City system provides shared audio between the three users, shared awareness through various types of 2D or 3D rendering, and collaborative access to a set of multimedia resources. Access to resources depends foremost of location, but also on user context. Figure 1 shows a prototype of the system.

3.1. Physical visitor (Vee)

The physical visitor is in the centre itself, equipped with wireless headphones and microphone, and a handheld personal digital assistant (PDA). The PDA includes a sensor package that is part of an ultrasonic positioning system [18]. The position is calculated from the flight time of ultrasonic 'chirps' and a geometric model of the gallery (see Section 4.2). The sensor package also includes an electronic compass for orientation information. The position and orientation are displayed on a map of the gallery on the PDA, along with the positions and orientations of the other two visitors.

3.2. VR user (Ana)

The virtual reality visitor uses a first person, 3D display with avatars representing the other visitors. The textured 3D model of the gallery was created from plans and photographs. Exhibits are modeled at a crude level showing form, but not fine detail. For example, text is unreadable within the 3D environment.

3.3. Web visitor (Dub)

Lastly, the *web* visitor uses a standard web browser displaying several Java applets, one of which is a variant of the physical visitor's map. Mouse clicks on the map are interpreted as movements around the gallery.

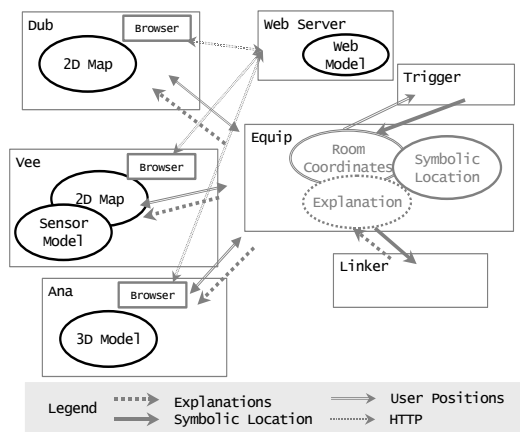


Figure 2 Overview of the city architecture

4. Data models and services

4.1. System architecture

An abstract view of the system architecture is shown in Figure 2. Implementation details can be found in [14].

The core part of the application is a shared dataspace implemented using Equip [12]. Equip provides a shared tuple space that allows applications to publish and receive events when tuples are created or manipulated. For this application the principle data items in the dataspace are positions of the users in a 3D coordinate system, symbolic locations of users, and explanations that are media references to be displayed to the users.

4.2. Identifying the spatial models

Ana, Dub and Vee all see representations of the locations of the others, using either oriented icons or avatars. Two of our models originate in these presentations since they are described differently to the application and are visualized in a different way. A *3D model* is used to describe the space for the purpose of creating a world for the virtual reality visitor. A *2D raster model* is used to form the basis of the map for both web and physical visitors.

The locations of the virtual and web visitors are explicitly defined in the same model that they are visualizing. Thus the web visitor clicks on the map to define their position, and the virtual visitor steers a 3D viewpoint through the 3D model. In contrast the physical visitor's position is measured in a *sensor model*, which is independent of the 3D model or 2D raster map. This model is defined by the positions of sensing devices. This in turn is based on an *ultrasonic model* that models different parts of the space, such as ceiling and main reflecting surfaces for the purpose of resolving ambiguous soundings.

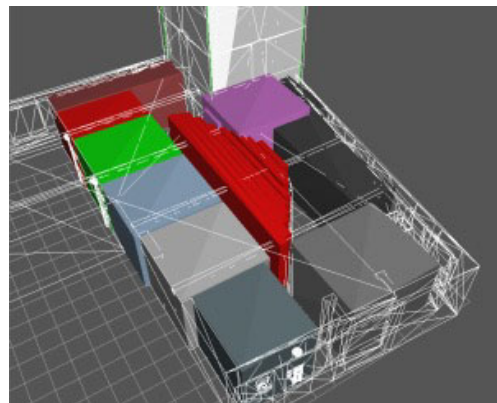


Figure 3 Visualization of room coordinates. A wire frame version of a CAD model of the gallery is included for comparison.

Moving to the system side, the first thing we notice is that the architecture requires all positions to be transformed into one *room coordinate* model. In this system, this happens to be the same as the *3D model*, though it need not be. Finally there is a *symbolic location model*. Location in this model is the primary key that is used to push content to the user.

Room coordinate model

The key geometric model for the application is a definition of room coordinates. This is a Cartesian model of dimension three, with a right-hand convention. Room coordinates are used as the reference frame for visitor positions. They also define a set of geometric zones with symbolic labels that form the key composed mapping from user positions to semantically meaningful or interesting information.

The choice of origin for room coordinates was arbitrary, and for convenience the definition was taken from the origin of a 3D CAD model that was being built. This CAD model followed a standard convention of having the XZ plane as the floor, with Y as “upwards”. X was chosen along the direction of the shortest wall of the room, and Z pointing towards the door. The origin was chosen to be coincident with the floor, and roughly centered in the gallery. The dimensions of the room are meters. The galley and tower fit completely within a bounding box, spanning (-8.7, 0, -12.6) to (11.6, 29.3, 11.3). Horizontal orientation (that is rotation about the Y axis) increases anti-clockwise in plan.

The only data items stored in this model are a set of axis-aligned boxes representing zones and users. A zone is a region of space and it comprises a list of boxes. Zones are non-overlapping. Figure 3 shows a visualization of the zones, where different zones have been given different colors.

A user is represented as a single box. Updates of the user's position in 3D model or 2D raster models updates



Figure 4 Rendering of the 3D CAD model of the Mackintosh Interpretation Centre. This model served as the basis for several other models.

their representation in this model. The position and orientation are not constrained, and thus user position comprises a 3D translation and rotation.

Sensor model

The ultrasonic tracking system defines its own model of space. The model consists of a Cartesian model of dimension three, with a right-handed convention, and a separate single valued orientation. The model is used to represent the position of the ultrasonic receiver. The origin and axes of this model differ from the room coordinates model: the XY plane is the floor, with Z upwards (that is, increasing sensor model Z corresponds to increasing room coordinates model Y, and increasing sensor model Y corresponds to decreasing room coordinates model Z). Unlike the room coordinate and 3D models, the origin and axes of this system are defined by transmitter placement. The transmitter placement was chosen such that the major axes of the sensor model would coincident with axes of the room coordinates. Thus the ultrasonic transmitters, which are placed on the roof of cubicles, are carefully aligned along the direction of the shortest wall and along the axis orthogonal to this.

Orientation is returned by a magnetic sensor and is not converted to a rotation in the Cartesian model. Zero in the orientation component is magnetic north. Note that this is not exactly aligned with any of the major axes. Note also that magnetic orientation increases clockwise in plan unlike orientation in room coordinates.

Dynamic testing of the realization of the sensor model showed: a 50% accuracy of 0.52m; a 95% accuracy of 1.83m; and an overall standard deviation of 1.29m [20].

3D model

The 3D model is a geometric model described in the VRML file format [24]. It contains 3D geometry and surface properties of the room itself, stands and certain

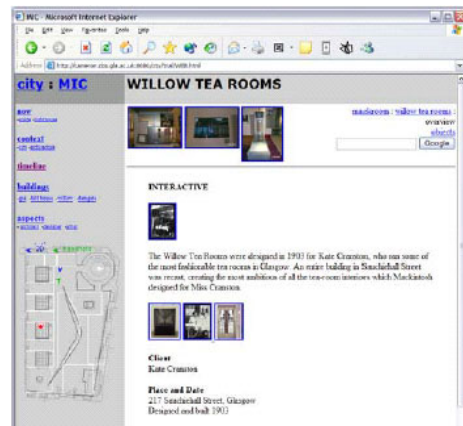


Figure 5 Gallery web pages and map display. Themes are displayed in the hierarchical menu on the top-left.

objects, see Figure 4. The 3D model is loaded by the 3D visualization client, and is internally stored as a scene graph, with geometric objects positioned in 3D space using hierarchical transformation matrices. The model also contains descriptions of the users as avatars. The position of the user's avatar is given as a 3D translation and 3D rotation. For a non-immersive view, the user-control metaphor usually only permits rotation of the user about the Y axis, though for an immersed user, all three rotations need to be specified.

2D raster model

A 2D map overview is provided for the physical and web visitors so that they can see an overview of the space and the users within it. The map is also used for position and orientation input by the web visitor. It is described as a 2D raster and is always presented in a fixed orientation. The origin of the raster model is the top left corner of the map, with X increasing "across", and Y increasing "down".

Orientation is single-valued, increasing anti-clockwise, with zero corresponding to increasing X. The map scale was fixed at approximately 12.4 pixels/meter, based on the PDA screen size (240x320) and web page layout. Users are represented by oriented arrows. Figure 5 shows the map embedded within a web page display.

Symbolic location model

The symbolic location model is a set of strings that are associated with different areas of the gallery. In the current implementation, the volumes are non-overlapping and non-hierarchical.

The symbolic locations were:

entry, guide, lighthouse, stvincent, Glasgow, contemporaries, gsa, architect, hillhouse, designer, willow, artist, Derngate, reputation, timeline

Down-stream processes (see Section 5) that generate dynamic content only use symbolic location and ignore exact positions of the user.

Ultrasonic model

The ultrasonic model measures distance relative to the base transmitter array. It includes a crude model of the gallery for the purpose of identifying reflected signal properties. The ultrasonic chirps are bounced off the ceiling, and thus the receiver does not necessarily have a line of sight to the transmitter. The model contains the relative positions of ultrasonic transducer positions and the ceiling height. The eight ultrasonic transducers are placed on the roofs of cubicles and on the top of a large dividing wall. Converting ultrasonic coordinates to sensor model coordinates involves assuming a receiver height of 1.5m. The two models are kept separate because multiple processes on the PDA know about the sensor model, but only the device driver for the ultrasonic knows about ultrasonic coordinates. This separation is kept distinct so as to enable future work on fusion of tracking data (see Section 8).

4.3. Other data models

Web pages

The gallery has an associated set of web pages containing text and images corresponding to the textual and graphical displays in the physical gallery. The pages are organized into thematic categories, based on documentation produced by the designer of the exhibition.

Explanation model

The explanation model makes the mapping from symbolic location to web pages by applying a contextual filter that includes presentation device and user type. An explanation is thus a URL and it is similar to CoolTown's notion of semantic location [17], though here we don't treat it as a spatial model.

4.4. Spatial services

In the current implementation we can identify the following services that convert between the different models of space:

3D model to room coordinates model

As mentioned, this is an identity transformation since the origin and axes were chosen to be the same.

Sensor model to room coordinates model

A datum needs to be defined in order to take convert sensor coordinates to room coordinates (see [16] for a discussion of datum and practical realizations of datums). In our model, this is simplified somewhat by the origins being the same, and only a switch of axes is required. Orientations differ in direction, offset and units..

2D raster model to room coordinates model

The 2D raster model is converted to room coordinates by first transforming to the sensor model and then transforming as above. The transformation to sensor model is determined by surveying two fixed positions in the two models, and reconciling orientations. Since the 2D raster lacks a third dimension, the user is given a fixed head height of 1.5 meters. The origin is translated and the horizontal rotation is adjusted for direction and offset.

Room coordinates model to symbolic location

Room coordinates describes zone volumes and volumes that represent users. The trigger service interprets collision of a user volume with a zone volume as indicating that the user is inside the symbolic location associated with the zone.

4.5. Other services

Symbolic location model to explanation model (linker)

The linker service generates a mapping of a user's symbolic location to a URL corresponding to an exhibition display. The URLs are passed to clients that load the corresponding web page, corresponding to viewing the physical display.

Note that there is no transformation to and from the web model. The web model is somewhat independent in that it exists within the web browser and is activated not by a user's position changing, but by a user's activity within a web browser.

5. Dependencies

Each of the models described in the previous section is identified separately due to presentation or authoring distinctions. At run-time the interpretation of context in one model requires that its relationship to any other model does not change. Or, if the relationship does change, this change is monitored and reflected in one of the spatial services. For example, if one of the transmitters is moved the ultrasonic model is no longer valid and thus none of the subsequent application behavior will be reliable for the physical visitor. This movement of the transmitter does not affect either the web or virtual visitors other than they may see inconsistent behavior on the part of the physical visitor. Certain parts of the system depend on others, and it is useful to describe two sets of dependencies: authoring dependencies that distinguish how a model is described initially; and data flow dependencies that indicate how models are affected at run-time. In Section 7 we will discuss how choices about application services and spatial services can affect authoring and run-time dependencies.

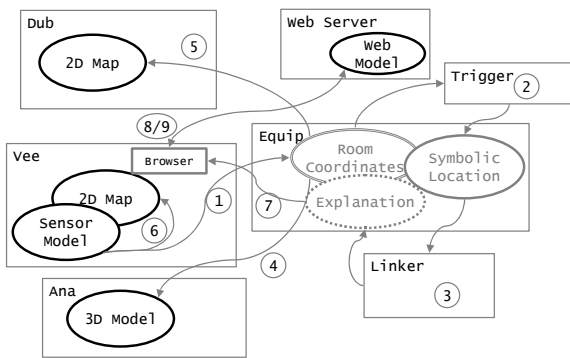


Figure 6 Data-flow resulting from Vee's moving

5.1. Data flow dependencies

Figure 6 shows the pattern of event flow as the physical visitor moves about the gallery. The annotations show how application logic moves between processes. In transforming between processes we call upon one of the spatial services in order to convert from one model to another as follows:

1. Vee's position is written into the shared dataspace (Equip). This involves the device calculating ultrasonic coordinates from time of flight, converting these to the sensor model and then converting these to the room coordinate model.
2. Vee's position is read by the trigger process, which scans through the volumes defined in room coordinates and outputs a symbolic location.
3. Linker compares the sequence of symbolic locations against lists of associations between symbolic locations, user type and explanations. An explanation URL is generated.
4. Positions in room coordinates are transformed to 3D model position—an identity transformation.
5. Positions in room coordinates are transformed to 2D map coordinates.
6. Sensor positions are converted to 2D map coordinates.
7. The explanation is placed back into equip and is picked up by Vee's client.
- 8/9. The URL is fetched.

If we consider each of the other users, we would find that only variations of these services are required. If Dub, the web visitor, updates his position, then the inverse of the transformation in step 5 is required to put his 2D map

position into room coordinates. Similarly if Ana, the virtual reality visitor, updates her position, then the inverse of the transformation in step 4 is required to put her 3D map position into room coordinates. Finally, Vee's map requires the positions of Ana and Dub to be displayed, and this requires a service to convert user positions in room coordinates into 2D map positions. This is a copy of step 5.

We can see that for the whole system to function correctly, each of these spatial services must operate consistently at run-time. To function consistently, we must first be able to monitor any changes in calibration between models. In our case, the only service that might change is the conversion of readings in the ultrasonic model to sensor model. Unfortunately, this is somewhat problematic as the transformation itself is hard to survey, and detecting that a distortion has occurred for whatever reason (such as a new electrical appliance dampening a signal) is difficult since we can't observe the data without resorting to the visualization services. Although you can detect that something is wrong with readings by looking at the map after transformation to 2D raster map coordinates, because of the intrinsic inaccuracy of the tracker it isn't possible to detect anything that is less than a major distortion. We rely on fixing the positions of the ultrasonic transmitters and surveying them precisely.

Although in the current system no other services are dynamic, it is intended in the future that the room coordinate model and subsequent symbolic models will be dynamically extensible (see Section 7.3).

5.2. Authoring dependencies

We have raised the issue of dynamic changes in models and their services, but even without dynamic changes we have potential inaccuracies in our system due to the nature of the models and their interdependencies.

The authoring relationships between the models are shown in Figure 7. The ultrasonic model is derived from a few characteristics of the physical gallery, including roof height, positions of the cubicles and sites for the tracker units. The 3D model is based on the architect's original plans and photos of the gallery as it was eventually built. The 2D raster map is modeled on the architect's plans. The symbolic location map is derived from the 3D model and the explanation URLs are derived from the symbolic locations. The web model was independently modeled on the physical gallery, using catalogue and site information. The dotted lines in Figure 7 indicate the scopes of the various spatial services. Each of these needs configuration as described in Section 4.4. Many of the spatial services are defined implicitly in the authoring step.

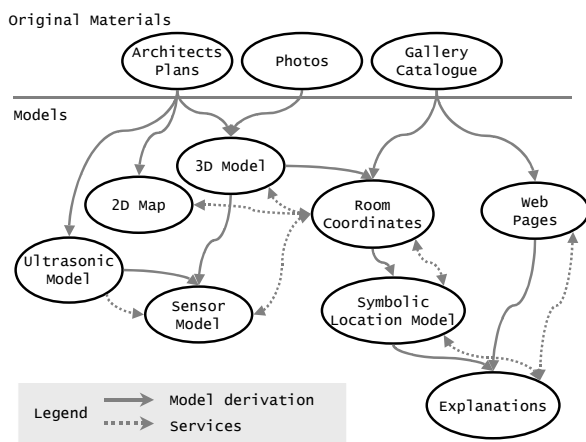


Figure 7 Authoring dependencies

From Figures 6 and 7, we can determine the assumptions that must not be broken, and the configuration that is recorded within the system. We can also determine how accuracy and error will accumulate through the system. We can identify the following sources of uncertainty in the model:

- Positioning errors from sensor model
- Imprecision in position input in the 2D raster model
- Registration between room coordinates and both of 2D raster models and 3D models
- Imprecision in the authoring of the 3D model
- Imprecision in the authoring of the volumes in room coordinates
- Imprecision in representation of user as a box when used by the Trigger service

There are also no consistency checks for the mappings between symbolic location, explanation location and web pages. The only way that errors are found is by experimentation with the run-time system.

Problems arise because data-flow dependencies are not checked against authoring dependencies at run-time. For example, there is no way of automatically checking if the services actually consistently model transformations. Indeed the most likely way that it will be discovered is when ambiguity arises when the reported position is used in another model such as the visual 2D or 3D models. While our user studies [4] confirm that users of mixed reality systems can overcome minor ambiguities or inconsistencies through talk and other shared resources, major inconsistencies might substantially inhibit their engagement and sense of presence.

6. Authoring spatial models and services

In this section we discuss how each of the models and services was described. We start with the 3D model, since the previous section indicated that this was the starting point for many of the model descriptions.

6.1. 3D model

The 3D model was authored using the packages Vectorworks and MicroStation for creating geometry and 3D Studio MAX for adding texture information. It was based on architect's plans, photographs and notes taken from a visit to the gallery. The plans were useful, but did not exactly reflect the gallery as built. For example, a pillar adjacent to the central dividing wall is slightly offset in actuality compared to the architect's plan. Therefore, the detail of 3D model is limited by the precision of the surveying. The implication of this is that the room coordinates and all the symbolic models are slightly inaccurate because they derived from this model.

6.2. Symbolic location and room coordinates

These two models were developed in tandem. There was a tension between larger zones and fine-grained authoring.

The symbolic location model was created by choosing a set of characteristic names for the space. In the Mackintosh Room it was natural to model them on the subject matter of the various displays.

Each labeled volume is a series of axis-aligned bounding boxes. The boxes are modeled in the AC3D package [2]. User position updates are then tested against these boxes in order to generate the symbolic location. See Section 7.3, for a discussion of alternative approaches at this stage.

During the development of the application, this model was one of the ones that changed most frequently. Each time a new symbolic location was required, the boxes had to be re-modeled because we required non-overlapping regions. Due to problems establishing the accuracy of the hand-held tracker (see Section 4), we actually changed from fine-grained boxes to much larger boxes.

The symbolic locations could be used independently as a top-level directory on a web browser, though we have built an independent web model for that purpose. They could also be used with location sensors such as radio frequency ID tags.

6.3. Sensor model and ultrasonic model

For reasons of convenience the sensor model was configured so that the origin of the sensor model would correspond to the origin of room coordinates. Sensor model axes were chosen according to the developer's normal practice and this was different from the 3D model. The ultrasonic model is obviously strongly related to the sensor model. The necessary measurements for conversion of time of flights into distances relative to the sensor model origin were made from plans and by measuring the transmitter placements.

The ultrasonic model contains a simplified model of the room, including room size, ceiling height, transmitter placements and transmitter directions. Time of flight readings are then turned into meters using transmitter

distance. A key part of the ultrasonic model is an approximation of the center dividing wall by two straight lines. All transmitters are either on this wall or on cubicles on one side of this wall, so a user on the opposite side of the dividing wall can only be tracked very imprecisely. For this case the model assumes that they are walking along a path roughly equidistant between the dividing wall and exhibition outer wall. Implicit in the definition of the ultrasonic model is the transformation from ultrasonic to sensor model coordinates. Sensor models coordinates use the floor of the gallery as the origin along Z, whereas measurements are internally made relative to transmitter locations at known heights.

6.4. 2D raster model

The 2D raster model was created from the architect's vector plans. These had to be tidied up by the removal of annotations before being rendered to a raster image, which was then hand modified for clarity. The mapping to sensor coordinates and thus room coordinates was achieved by measuring the raster positions of a small set of features common to the plans and 3D model.

7. Discussion

7.1. Roles of models

In identifying each of the models, we have been able to isolate run-time and authoring dependencies, and thus the errors and inaccuracies that can arise in our system.

The different models were necessary because of the different domains of description, the distributed execution model of the application, the need for heterogeneous user input and the requirements of user displays.

We had decided in early development to treat the users as similarly as possible, and thus most of the application locus is mostly invested in the room coordinate services for matching locations to multimedia explanations. This had the advantage of simplifying the presentation clients, since they now deal with a single representation of all user positions. However we ended up with a model where several disparate authoring processes must be reconciled.

7.2. Alternatives

The choice of detaching presentation models from symbolic location models allows simplicity in description, but it is a compromise. It does allow us to more easily integrate other input devices. For example radio-frequency ID tags could be used to explicitly indicate a user's being in a symbolic locations, thus bypassing the sensor and room coordinates models.

However because our current solution centralizes important facilities this means that disconnection between clients renders inoperative all services aside from local map update. A more robust alternative would be to migrate either instantiations of services on to the clients, or transform those services into local variations exploiting

the models local to the device. Thus the symbolic location mapping service could be done in the 2D raster model, or the sensor model. Such multiple implementations of the services would not remove the need for the transformation services. Re-implementing the services in our case is fairly simple, in that it requires the zone and user boxes to be transformed. However if the trigger were based on a predicate such as visibility this re-implementation would be much harder if not impossible. If the application model was more complex than ours, and involved, for example, explanations that depended on group context, then the results of the symbolic location model would still need to be shared to all sites, potentially introducing a consistency issue.

7.3. Authoring and deployment processes

In user trials [4] we found that the use of bounding volumes for symbolic location authoring was limiting because it was quite a poor model of how people actually look at the exhibits. One alternative way would be to track the PDA and explicitly associate sensor readings with particular exhibits based on actual user browsing activity. Cluster analysis of these readings could provide separable regions in 4D, three for position and one for heading. Transforming these into 2D raster map and 3D model would be difficult because readings in the sensor model are inherently non-linear and discontinuous due to reflection or attenuation affects.

Tracking user activity might also feed into adaptation and correction of the models. While it is possible that model changes may require manual checking by an editor or curator, sources and suggestions for change can be automatically derived from visitor activity. For example, if we find that there is a part of a region where visitors generally read web pages or interact with artifacts associated with another neighboring region, we might shrink the former region and extend the latter region, to better suit user activity. Similarly, if we find that users in a particular region consistently browse pages that are not reachable purely by location, then we might extend the zones to take account of what appears to be useful information.

7.4. Error handling

In Section 5.2, we mentioned the difficulty in detecting when authoring assumptions had been broken and gave the example of the sensor base being moved. In our situation this is the only registration that can dynamically change. We can easily imagine more complex situations, where sensors may or may not be off-line or where tracking systems themselves are mobile. Although we avoid verification of assumptions about registration, it will become necessary in more complex situations. In our situation, verification can be as simple as placing the PDA tracker in a known position and inspecting its subsequent visual update on the 2D raster map. In a situation with multiple sensor systems with overlapping

sensing regions, some form of inter-system confirmation may be possible. Castro et al., use probability estimations to fuse data between different range sensors [7]. Angerman et al. discuss an approach to fusing data from heterogeneous sources using probability density fields [3]. A variation of these processes could be used to detect registration errors.

7.5. Further uses of spatial models

Our rationale for creating multiple models was either to simplify representations for the user, to simplify application descriptions or to simplify deployment issues. However with each model containing only the elements necessary for its immediate function, we have removed detail that might be useful. For example, although we have a detailed 3D model of the environment, we have not used it to its full extent. Brumitt and Shafer note that with a geometric model of the objects in a space, more complex relationships involving visibility between objects can be built [5][6]. A straightforward development would be to prevent the web user's position being placed over or inside objects in the 3D model. A similar development would be to incorporate the geometry described within the 3D model into the ultrasonic model so that positions could be constrained to empty, reachable regions.

An important consideration for evolution of future systems will be the impact of multiple models on latency. At the moment, the user's own position updates almost immediately on their own visualization, but others are delayed by the use of several distributed services. The total end-to-end latency including web re-refresh is around one second. The implication for a system that attempts to correct position reports against solid models is that such a model needs to be as close to the actual positioning interface as possible. As we have noted, conversion of solid models from, say, room coordinates, into 2D raster or sensor coordinates is not so simple. At the very least, we would have the same data in different models, and authoring processes would need to reflect the need to update multiple models.

8. Future requirements

Although successful, the system described in this paper is complex and requires significant configuration. If it were to be re-deployed in another context the authoring process might even be different because different resources would be available at the beginning.

The key step in authoring was defining the common reference coordinate, which in our case was room coordinates. In general this needs to be a coordinate based on some immutable representation of space, such as a plan or map. Sensing systems are inherently ambiguous in that they realize only imprecisely an ideal coordinate system. Indeed, in the UK, the difference between the WGS84 coordinate system that GPS realizes, and the ground truth, can change by 5mm when a high-pressure

weather system moves over the British Isles [16]. Thus no matter how precise GPS devices become there will be an inherent uncertainty in relating readings to the real world.

Some requirements for future work are thus:

- Tools for describing transformation between coordinate systems. In any system with hierarchical or overlapping geometric models, the surveying of two or more (three or more for 3D) points in multiple models establishes a transformation between those models. Of course, more points over-determines the transformations and thus a minimization technique is needed to find the best fitting transformation.
- A host of different tests can be imagined in order to build confidence in the consistency of a system. For example, testing readings of a sensor at a known fixed position against established position, verifying that a known position in a geometric position generates the expected symbolic location or checking a reading against known physical bounds.
- Better tools to rapidly estimate accuracy of a positioning system in order to better customize location dependent information. See [20] for an example.
- Integrated authoring systems that allow 2D vector and raster models and 3D models to be described and visualized in combination.
- Better tools for fusing readings from multiple positioning systems. See [3][7] for examples.
- A fuller ontology of spatial models needs to be established so that tools for describing and realizing coordinate systems can be shared between processes. We also need to evaluate geographical description standards such as GeoVRML [21] for suitability as a basis for describing location models that cover a larger extent.
- Better techniques are required to reason about accumulation of error and uncertainty in values as they propagate through the system.

Future versions of the City project systems will start to embed such facilities in their authoring tools or reflect these concerns in their run-time implementations.

9. Conclusions

Mixed-reality systems demand multiple models of space. We have analyzed a novel mixed-reality system that supports simultaneous co-visiting between physical, web and virtual reality users and we have shown how it requires several geometric and symbolic models simultaneously. This need to support multiple models is most clearly apparent when one combines geometric models with models based on position sensors and models based on symbolic associations between locations. We claim that most similar systems utilize multiple models of space and transform between them. We have shown, with reference to our own system, how run-time use of models of space, each of which might have been built by a different authoring procedure, necessitates reflection on

the consistency of spatial services and treatment of error as it propagates through the system. Despite some errors and ambiguities in position reporting we have demonstrated successful shared visits amongst three users, and we have discussed several avenues for research and development in this area.

Acknowledgements

We would like to acknowledge all of the City project members. This work is supported EQUATOR Interdisciplinary Research Collaboration (EPSRC Grant GR/N15986/01).

References

- [1] Abowd, G, Atkeson, C., Hong, J., Long, S., Kooper. R., Pinkerton, M., Cyberguide: A mobile context-aware tour guide, i, 3, pp. 421-433, 1997.
- [2] AC3D, <http://www.ac3d.org/>
- [3] Angermann, M., Kammann, J., Robertson, P., Steingäß, A., Strang, T., Software representation for heterogeneous location data sources using probability density functions, *International Symposium on Location Based Services for Cellular Users*, Munich, Germany, February 2001
- [4] Brown, B., MacColl, I., Chalmers, M., Randell, C., Steed, A., Lessons from the lighthouse: Collaboration in a shared mixed reality system, *Proceedings of CHI 2003*, Ft. Lauderdale, p577-585, ACM Press
- [5] Brumitt, B., Shafer, S., Better Living Through Geometry, *Personal and Ubiquitous Computing*, 5(1), 2001
- [6] Brumitt, B., Shafer, S., Topological world modeling using semantic spaces, *Workshop on Location Modeling for Ubiquitous Computing 2001*, pp. 55-61.
- [7] Castro, P., Chiu, P., Kremenek, E., Muntz, R., A probabilistic room location service for wireless networked environments, *Ubiquitous Computing*, 2001. pp. 18-34
- [8] Dix, A., Rodden, T., Davies, N., Trevor, J., Friday, A., Palfreyman, K., Exploiting space and location as a design framework for interactive mobile systems, *TOCHI*, 7(3), 2000, pp. 285-321.
- [9] Espinoza, F., Persson, P., Sandin, A., Nyström, H., Cacciatore, E. Bylund, M., GeoNotes: Social and navigational aspects of location-based information systems, *Ubiquitous Computing*, 2001. pp 2-17.
- [10] Jiang, C, Steenkiste, P., A Hybrid Location Model with a Computable Location Identifier for Ubiquitous Computing, *Proc. UbiComp 2002*, pp. 246-263.
- [11] Galani, A., Chalmers, M., Can You see me?: Exploring co-visiting between physical and virtual visitors, *Museums and the Web 2002*, Boston, US. Archives & Museum Informatics.
- [12] Greenhalgh, C., EQUIP: A platform for distributed interactive systems, <http://www.crg.cs.nott.ac.uk/~cmg/Equator/Downloads/docs/equip-tech.pdf> (accessed 9th October 2003).
- [13] Leonhardt, U., Supporting Location-Awareness in Open Distributed Systems, PhD Thesis, Department of Computer Science, Imperial College of Science, Technology and Medicine, University of London, 1998.
- [14] MacColl, I., Millard, D., Randell, C., Steed, A., Shared Visiting in EQUATOR City, *Proceedings of CVE2002*, Bonn, Germany. ACM Press
- [15] Nord, J., Synnes, K., Parnes, P., An Architecture for Location Aware Applications, *HICSS-35*, Big Island, Hawai'i, USA, January 2002.
- [16] Ordnance Survey, A guide to coordinate systems in Great Britain, <http://www.gps.gov.uk/guidecontents.asp> (accessed 9th October 2003).
- [17] Pradhan, S., Semantic Location, *Personal Technologies* 4(4). Springer, 2000, pp 213-216.
- [18] Priyantha, N.B., Miu, A., Balakrishnan, H., Teller, S., The Cricket Compass for Context-Aware Mobile Applications, *Proc. 7th ACM MOBICOM*, Rome, Italy, July 2001.
- [19] Randell, C., Muller, H., Low cost indoor positioning system, *Ubiquitous Computing*. Springer, 2001. pp. 42-48.
- [20] Randell, C., Muller, H., Exploring the dynamic measurement of position, *International Symposium on Wearable Computing 2000*, pp. 117-124.
- [21] Reddy, M., Iverson, L., GeoVRML 1.1 Specification, www.geovrml.org (accessed 9th October 2002).
- [22] Spohrer, J., Worldboard, what comes after the WWW? <http://www.worldboard.org/pub/spohrer/wbconcept/default.html>, 1997, (accessed 10th October 2002)
- [23] Volz, S., Fritsch, D., Klinec, D., Leonhardi, A., Schützner, J., NEXUS - Spatial Model Servers for Location Aware Applications on the basis of ArcView, *Proceedings of the 14th ESRI European User Conference*, 1999.
- [24] VRML97 International Standard (ISO/IEC 14772-1:1997)