The Use of Models of Space in Mixed-Reality Systems

Abstract

In this paper we discuss the use of models of space in mixed-reality systems. By model of space we mean a distinguishable geometric or symbolic description associated with a physical space. We outline several types of model that might exist, how they are surveyed or otherwise authored, how they are represented to the users and how the underlying middleware and sensors support them. We show that systems often contain numerous models of space and that a great deal of work goes into 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 space [1, 7, 9, 21, 26]. 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, but they might make different services available using different models. Most commonly, the model in which the application logic lies is not the same as the model that is used to present current context to the user.

For example, consider the realization of a restaurant locator service. The user's position might be reported by a global positioning system (GPS) service in the WGS84 coordinate system. Another service might then be used to convert this into a map location, referenced in meters relative to some "fixed" point. Yet another service might determine the street that the user is on using a model of streets based on line segments, and estimating which line the map position is nearest to. The restaurant locator service can then search for a nearby restaurant. Yet another service converts the restaurant's postcode (zip code) into another map position, and this map position is also located on the street network. A route can then be determined, but in order to be presented to the user physical directions have to be given (e.g. turn right, go 200m) or some map has to be rendered to the user.

In this simple hypothetical example we can identify at least five "models" of space:

1. GPS, or rather WGS84 (latitude and longitude)
2. Map coordinates (vector model)
3. Street network (vector model)
4. Postcode (labeled vector regions)
5. Rendered map (raster array)

It can be argued that several of these models could be integrated by, for example, implementing them all within a geographic information system (GIS) system. However in practice several distinct models usually exist, especially systems that combine geometric models (i.e. models that convey measurable distance, such models are themselves spaces in the mathematical sense), with models that use a symbolic model of space (see Section 2). We will argue that they should be considered as different models, because they could be authored or maintained separately, have differing levels of provenance and different coordinate systems.

In the example above, the locus of the application switches from one model to another as the query progresses. Indeed this is likely to be implemented as several distributed services. Each of these switches implies a potential transformation, and these transformations are often complex or sometimes even ill–defined. Thus the process of taking data from one model and converting it into another is potentially fraught with difficulties or assumptions.

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

2. TYPES OF MODEL

Leonhardt [16] 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 this coordinate system is defined (see Section 4 for examples), location can be described in terms of polygonal 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, no transitive distance relations and so on.

Geometric and symbolic locations may be hierarchical. For example, at different levels of granularity, we may have rooms, buildings, cities and countries. Both symbolic and geometric models of space may contain overlapping regions. If the associations between symbols become even less constrained, for example with arbitrary graphs, directed arcs and asymmetry of association, we can leave the domain of metric spaces. Web location models, for example, can have asymmetric connections and distances between nodes. A symbolic model may be used to describe or model a space, but may not necessarily be a mathematical space itself.

Most real systems contain elements of both geometric and symbolic descriptions of space. Leonhardt calls these hybrid models [16]. Jiang and Steenkiste describe a hybrid system for an indoor location system [10]. Their model uses a symbolic location for gross descriptions of space at building and room level, and then a geometric description for intra-room locations and positions. In their terminology, these types of description of space are labeled as hierarchical and coordinate based, though we will use the terminology from Leonhardt since geometric models can imply hierarchy through containment or intersection.

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 as another measurable geometric position changes. Thus symbolic locations can be quite gross. We also expect position sensors to report changes continuously as the bearer moves modulo the sensors' resolution. But 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 as well. 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 (as for example, 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.

Additionally, we might know or be able to detect the *registration* between positions reported in this system and some similar model or between this system and 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.

Classification of Model Use

Not all models of space are explicitly defined, and even those that are explicit might not be of much interest outside of a single process. We thus distinguish between public and private models.

Public — a model that is exposed to services such as rendering or application, or is otherwise shared. Typically a model about which more than one device or service needs to communicate, or in which something must be authored. Shared models come with assumptions that may or may not be explicitly modeled, and these assumptions may or may not be checked at modeling and run-time stages.

Private — a model that is subsidiary and that is not exposed to middleware or users. Typically these will be internal to a single process and include subsidiary coordinate systems such as logical map coordinates (as opposed to screen co–ordinates) or viewing co–ordinates in 3D.

Spatial Services

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 model itself is just a description of the space, it does not describe actions or filters that depend on the configuration of space.

We can distinguish a *spatial service* from a model of space, in that a spatial service transforms between the domains of two models of space. This 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 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 (Vee) is in the physical centre. The other two are visiting using a virtual environment or a web browser. The City system provides shared audio between the three users, shared awareness through various types of rendering, and collaborative access to data resources. Figure 1 shows a prototype of the system.

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).

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 [21]. The position is calculated from the flight time of ultrasonic 'chirps' and a geometric model of the gallery (see Section 4, *Sensor Model* and Section 4, *Ultrasonic Model*). 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.

VR User (Ana)

The *virtual reality* visitor uses a first person, 3D display with avatars representing the other visitors. Figure 10 shows the non–immersive display. The textured 3D

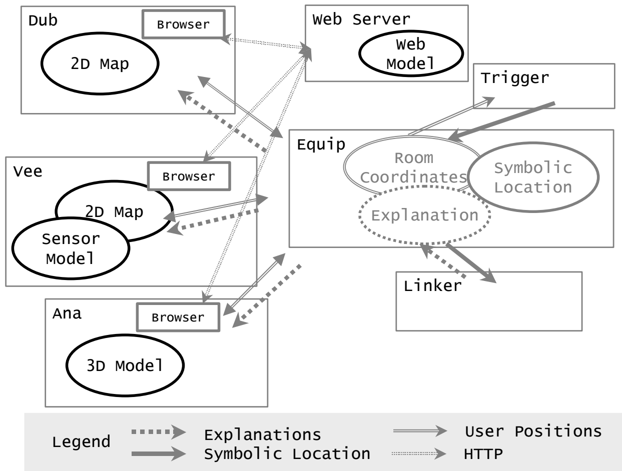

Figure 2 Overview of the City demonstrator architecture

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.

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, with the direction from the old position to the new position treated as the new orientation. As with the physical visitor's map, the other visitors' positions are displayed on the map with differently colored icons.

4. MODELS AND SERVICES IN THE CITY PROJECT

System Architecture

An abstract view of the system architecture is shown in Figure 2. More details about the individual services and implementation can be found in [17].

The core part of the application in this representation 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.

Identifying the Models

The positions of Ana, Dub and Vee are used to form the shared displays presented to each of the three visitors. Each sees representations of the locations of the all three, 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 visualization for the virtual reality visitor. A *2D raster model* is created to form the basis of the visualization 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. 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.

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 are several variants of symbolic models, namely the *symbolic location model*, *explanation model* and *web model*. The reasons for making these models distinct are discussed below.

Public Models

We have identified seven public models.

- *Room Coordinate Model*

  The key geometric model for the application is a definition of room coordinates. This is a Cartesian spatial model of dimension three, with a right-hand convention. Room coordinates are used as the fundamental reference frame for visitor positions, and to 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. See Figure 3. X and Z axes are thus not exactly aligned with any UK mapping convention nor with true or magnetic north, though X happens to be within a few degrees of magnetic north.

  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 with tower fitting 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.

  Within this model are described several boxes that correspond to zones in the center. Each zone can be represented by a list of boxes, but zones are non-overlapping. Figure 4 shows a visualization of the zones. In the model zones are labeled, and these labels form one of the other public models of the space (see *Symbolic Location*, below).

  Users are represented within this model as boxes. Updates of the user's position in 3D model or 2D raster models updates their representation in this model. The position and orientation are not constrained, and thus user position comprises a 3D translation and a 3D rotation.

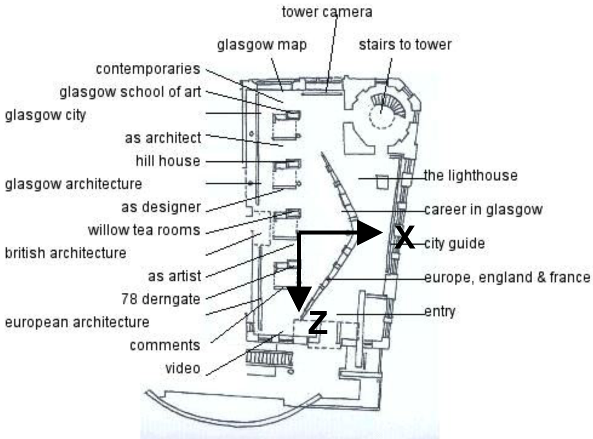

Figure 3 Annotated plan of the gallery, with the origin and axes of the room coordinate model marked. Y is out of the page. The labels formed the basis for the labeling of room coordinates

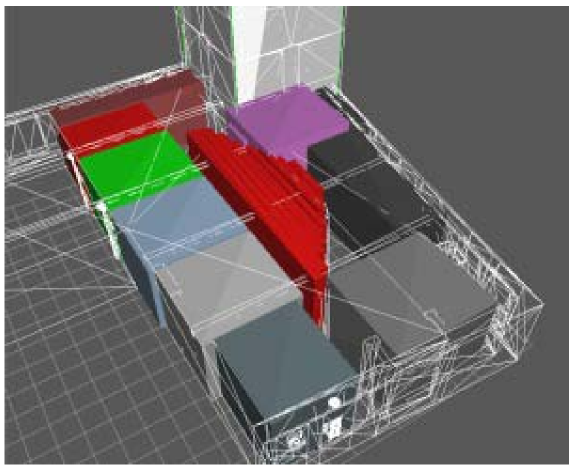

Figure 4 A representation of room coordinates. The model only contains labelled boxes, this visualisation attributes a random color to each box, and includes a wireframe version of a CAD model of the gallery for comparison. Users are represented in this model by axis aligned boxes.

- *3D Model*

  The 3D model is a geometric model described in the VRML file format [27]. It contains 3D geometry and surface properties of the room itself, stands and certain objects, see Figure 5. 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 (see Section 7). 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.

  As described above, the construction of the 3D model served to define the origin and axes for room coordinates. There is no need for the two to be co-incident. Note that the 3D model includes a representation of a room that is not described in the other models. This is the room shown on the extreme left of Figure 4 and Figure 5.

Figure 5 A rendering of the 3D CAD model of the Mackintosh Interpretation Centre. This model served as the basis for several other models.

- *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 calculate the position of the ultrasonic receiver that is modeled as a single 3D position and orientation. 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.

  Dynamic testing of the realization of the sensor model showed: a 50% accuracy of 0.52m, that is, good coverage; a 95% accuracy of 1.83m, that is, poor coverage; and an overall standard deviation of 1.29m [23].

- *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, the users and locations 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 6 shows the map in use on the PDA.

Figure 6 2D raster map displayed on a HP Jornada 568 handheld PC. The three cursors at the top of the figure represent the three users.

- *Symbolic Location Model*

  The symbolic location model is a set of names that are associated with different areas of the gallery as described in the room coordinate model. In the current implementation, the volumes are non-overlapping and non-hierarchical since a choice was made to only allow a single symbolic location per user.

  The symbolic locations were:

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

  We considered this to be a separate model of the space because of alternative ways of locating users through non-positional systems. Currently a symbolic location is generated because a user's position in room coordinates intersects a zone volume. However we could directly sense the user's being in one of these locations, without using a positioning technology. They could, for example, explicitly swipe a radio-frequency ID tag past a scanner, to indicate that they were in this location. The system would not know their position in any of the coordinate systems so far described, but could add this user location to the database.

  A second reason for describing this as a separate model is that the down-stream processes (see Section 5) that generate dynamic content only use symbolic location and ignore exact positions of the user.

- *Web Model*

  The gallery has an associated web model, consisting of links between 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. The web pages correspond roughly to the major displays in the areas of the gallery, so each can be considered a separate symbolic location, where location is defined by the content's reference to a location, rather than explicit spatial continuity. The web model is realized by lists of hyperlinks (shown top left and top right in Figure 7).

  Unlike the symbolic location model identified above, there is a hierarchical structure to the web model. The top level corresponds to the exhibition themes, with each theme spanning several of the symbolic locations identified above (for example, the buildings theme includes the gsa, hillhouse, willow and derngate zones). The second level corresponds to the explanation model (see below), with individual hyperlinks having symbolic location names as their anchors, and web page URLs as their targets. The individual pages comprise the third level since each zone is comprised of multiple displays, with each page corresponding to a display in a zone directly reachable from the other pages in the zone.

  The identification of web page to actual physical space is imprecise, so it is arguable if this is a model of space at all. However it does reflect the tensions in system design for mixed-reality systems and for that reason we include it in our set of models of space. For the real and virtual visitors changes in their location in the web model are dependent on their position in their original coordinate system, so there is an obvious relationship between their movement and the web page. Indeed it would be possible for them to determine what the mapping was over time. For the web user, the web model is active, in that they can "move" around the web model by selecting links. In Section 8 we will discuss whether the system should reflect that as a movement of their avatar.

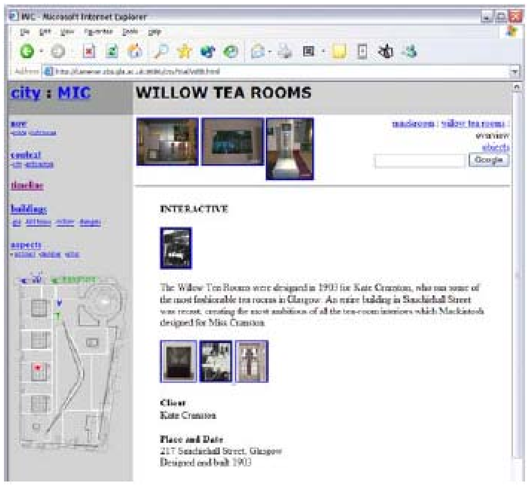

Figure 7 Realization of web page model within a standard browser, showing lists of web page locations top left (first and second levels) and top right (third level)

- *Explanation Model*

  The explanation model is similar to the symbolic location model, but it associates relative URLs to spatial locations. An example explanation location is */trial/gsa/overview.html*. This is similar to the CoolTown semantic location model [20]. The explanation model is based on the symbolic location model above, but also depends on other context information, specifically the device the user has. This means that the same symbolic location is associated one or more explanations. As noted above, the explanation model corresponds to the second level of the web model, but exists separately so it can be used in the linker service described below.

Private Models

These models are not exposed within the system but are used as part of the implementation of particular services. The decision to not represent these models is usually made so as to avoid confusing users of the system, since users usually explicitly operate in only one model, although their experience might incorporate information from other models.

In the implementation of the City system we can identify the following private models:

- *3D Rendering Coordinates*

  Associate with the 3D rendering clients are a number of models that are important only to that client. These include viewing coordinates of the real-time rendering that include field of view calculations. Work in collaborative virtual environments has shown that mis-understandings of the work of others can arise because viewing parameters are not represented in other parts of the system.

- *Ultrasonic Model*

  The ultrasonic tracking system 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 ultrasonic model is essentially the same as the sensor model, with the addition of the ultrasonic transducer positions, the ceiling height and an assumed receiver height of 1.5m. The eight ultrasonic transducers are placed on the roofs of cubicles and on the top of a large dividing wall.

We mentioned in Section 2 that we might consider logical map coordinates as a separate and private model. In the current system, the logical map coordinates associated with the 2D maps presented to physical and web visitors are given in the sensor model.

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 [19] 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. Note that orientation in the sensor model needs to be negated and offset.

- *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 transformed as above. The transformation to sensor model is determined by surveying taking two fixed positions in the two models, and reconciling orientations. Since 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 (Trigger)*

  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 in inside the symbolic location associated with the zone. Collision is performed using an oc-tree algorithm.

- *Symbolic Location Model to Explanation Model (Linker)*

  The linker generates a mapping of a user's symbolic location to a URL corresponding to an exhibition display. As noted above, the explanation model corresponds to the second level of the web model. 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. However as we will discuss later in Sections 7 and 8, the web user's browsing activity is not reflected in map updates, so they get to a situation where their web location and symbolic location are inconsistent.

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: modeling dependencies that distinguish how a model is described initially; and data flow dependencies that indicate how models are affected at run-time. In Section 8 we will discuss how choices about application services and spatial services can affect modeling and run-time dependencies.

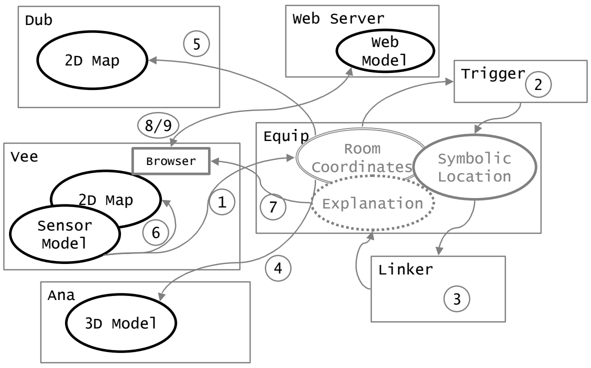

Figure 8 Data-flow resulting from Vee's moving

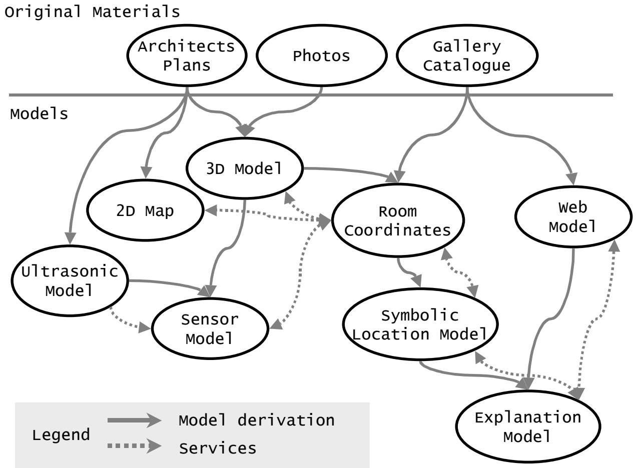

Figure 9 Modeling Dependencies

Data Flow Dependencies

Figure 8 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 transformation 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 a transformation from sensor model to 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 list 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 in this case.
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 which forces the browser to refresh the URL it contains.

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 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 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 service 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 application 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 to the intrinsic inaccuracy of the tracker it isn't possible to detect anything that is less than a major distortion. We rely on surveying the positions of the ultrasonic transmitters precisely and making sure that they do not move.

Although in the current system no other services are dynamic, it is intended in the future that room coordinate model and subsequent symbolic models will be dynamically extensible (see Section 8, *Revising the Authoring and Deployment Processes*).

Modeling Dependencies

We have raised the issue of dynamic changes in models and their services, but even with dynamic changes we have potential inaccuracies in our system due to the nature of the models and their interdependencies.

The modeling relationships between the models are shown in Figure 9. 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 9 indicate the various spatial services. Each of these needs configuration as described in Section 4, *Spatial Services*. Many of the spatial services are defined implicitly in the modeling step.

From Figures 8 and 9, 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 follow 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 modeling of the 3D model
- Imprecision in the modeling of the volumes in room coordinates
- Imprecision in representation of user as a box when used by the Trigger service

These are also concerned with sampling and use of position data, but there are also no consistency checks for the mappings between symbolic location, explanation location and web location. The only way that errors are found is by experimentation.

Essentially the problem is that data-flow dependencies are not checked against modeling dependencies at run-time. Thus there is no way of checking, for example, if the services actually consistently model transformations. If the sensor base is moved after the sensor has been calibrated, there is actually no way of checking consistency at run-time. Indeed the most likely way that it will be discovered is when ambiguity arises with regard to a different model that is not modeled directly relative to the sensor, 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 substantially inhibit their engagement and sense of presence.

6. AUTHORING 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.

3D Model

The 3D was modeled using the packages Vectorworks and MicroStation for modeling 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, although the 3D model is a continuous model, and dimensionally it is accurate, the actual form of presentation is limited by surveying. The implication of this is that the room coordinates and all the symbolic models are slightly inaccurate in as far as they derived from this model.

Modeling packages such as MicroStation and 3D Studio MAX have their own conventions for coordinate axis. 3D Studio Max has a convention that the horizontal plane is the XY plane with Z up. This is in contrast to our finished 3D model where the convention is the VRML convention of X right, Y up, Z towards the user. There was therefore a minor step between export from 3D Studio MAX where the model was wrapped in a VRML transformation node in order to re-orient the whole model.

Symbolic Location and Room Coordinates

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

The symbolic location model is one of the most important since it comprises a semantic model of the space. The space is modeled 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. Figure 4 shows the volumes of the model, which correspond to the semantic locations.

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 8, *Revising the Authoring and Deployment Processes,* 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.

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 axes were chosen according to the developer's normal

```
<linkbase>
 <association id=3D"gsa-info">
  <structure>link</structure>
  <relationtype>explanation</relationtype>
  <description>Glasgow School of Art</description>
  <feature>direction</feature>
  <binding>
   <reference>
    <data id=3D"gsa" state=3D"defined"></data>
   </reference>
   <featurevalue feature=3D"direction">src</featurevalue>
  </binding>
  <binding>
   <reference>
    <association>
     <structure>set</structure>
     <relationtype>concept</relationtype>
     <description>GSA concept</description>
     <binding>
      <context>
       <contextvalue key=3D"device">WEB</contextvalue>
      </context>
      <reference>
       <data>
        <url>/trial/gsa/overview.html</url>
       </data>
      </reference>
     </binding>
    …
    </association>
   </reference>
   <featurevalue feature=3D"direction">dest</featurevalue>
  </binding>
 </association>
 ...
</linkbase>
```

Figure 10 Excerpt from Linker definition file, showing an association between a symbolic location and an explanation for the Web user

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 meter readings using transmitter distance. A key part of the sensor 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.

2D Raster Model

The 2D raster model was created from the architect's plans. These had to be tidied up by the removal of annotations before being rendered to a raster image, which itself 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.

Web Model

The web model describes the room in a traditional web page style, and is based on the exhibition catalogue. The pages themselves do not strictly follow the names of the symbolic location model, but a subset forms the basis of explanations. There is a linking structure between pages based on thematic relationships as well as spatial adjacency.

Explanation Model and Linker Service

The explanation model is defined by the set of target URLs that the linker service can output and that the web server can support. Thus the URLs are recorded from the web server, and then the linker service is built around these. Associations between symbolic locations users and explanation URLs are expressed in an XML file. Currently this is edited by hand, and as symbolic locations or URLs are changed or new associations are made, the file must be edited. Figure 10 shows an excerpt from the Linker service definition.

7. USER PRESENTATION AND USER EXPERIENCE

User Representation

In each model, the user's current position or location is explicitly expressed. This will be retained as state in one or more services, but additionally, some services and models express a more detailed model of the user. This ability to provide a more detailed representation of the user is very important in situations where that representation can be shared with different users and thus used as a means of signifying identity and personal state [13].

The 3D model describes each user by their position and a geometric model (commonly known as an avatar) for representation. See Figure 11.

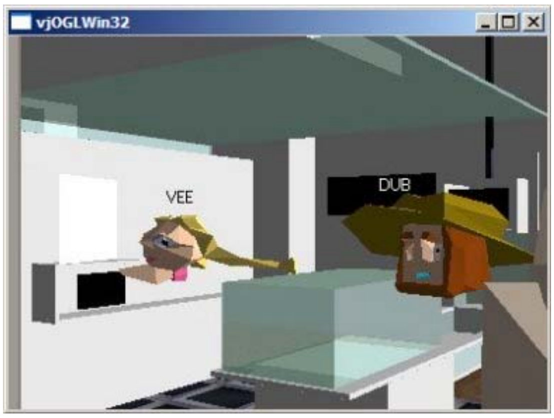

Figure 11 Two avatars within the 3D model

User Experience

In the user trials of our system [4], interaction would sometimes pause when participants found a difference between the physical and digital representations of the Centre. For example, two visitors could be spatially close and facing the same way but be presented with different exhibits because they are in different regions. This was particularly problematic for the VR visitor who could not actually see the region boundaries. They might then mistakenly assume they were both looking at the same exhibit, and begin talking at cross-purposes.

Major differences or gaps in the models could inhibit interaction between visitors. Images and text on certain web pages were only available to the web visitor, and video displays were not available for the web and virtual visitors. When a visitor started to use and talk about a page or an interactive display that was not shared, the others would often refrain from interacting and move on to other exhibits.

One important issue with the web visitor's interface was that although their web model was active, their moving through the web model by selecting hyperlinks on the web pages did not move them on the 2D map. Confusion arose because of the asymmetry here. Moving on the map causes a movement in the web model, but not vice-versa.

8. DISCUSSION

Roles of Models

In identifying each of the models, we have been able to isolate run-time and modeling 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 need to take input from several source and the requirements of user interaction. Thus 3D models and 2D raster models were required for different presentation requirements, and we saw that the 2D raster model in our case is inherently less accurate because of the rendering stage.

We had decided in 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 positions to, ultimately, descriptions of those positions through multi-media explanations. This has the advantage of simplifying the presentation clients, since they now deal with a single representation of all user positions. However we have ended up with a model where several disparate modeling processes must be reconciled.

Alternatives

The choice of detaching presentation models from symbolic location models allows simplicity in description, but it is a compromise. It allows us to more easily integrate our heterogeneous input devices, and arguably with a single user type it should have been done differently. Other systems such as WorldBoard [25], propose a single spatial model for application logic.

Our current solution centralizes important facilities, which means that disconnection between clients renders all services aside from local map update inoperative. A more robust alternative could be to migrate either instantiations of services on the clients, or transform those services into local variations exploiting the local models. Thus the symbolic location mapping service could be done in the 2D raster model, or the sensor model. This does not remove the need for the transformation services, but it does mean that either the spatial volumes need to be transformed, or that they have to be re-authored in each model. 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.

Revising the Authoring and Deployment Processes

In Section 7, we noted a finding from user trials that the use of bounding volumes for symbolic location modeling 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. This is not solely due to the relationship between sensor model and 3D model, but due to this description having embedded within it any non-linearity or discontinuity in the sensor readings 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 in our web model, then we might extend it to take account of what appears to be useful information.

Error Handling

In Section 5, *Modeling Dependencies,* we mentioned the difficulty in detecting when modeling 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.

Further Uses of Spatial Models

Our rational 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]. 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-fresh 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.

9. FUTURE REQUIREMENTS

So far, our system and user trials have been confined to the Mackintosh exhibition room. However, some of the models within the Mackintosh room make reference to buildings and locations across the city. Part of our ongoing work involves extension to this wider geographical area. We will have to handle multiple types of sensors for position, such as GPS and RFID tags as well as ultrasound. Similarly, we face issues of combination and consistency between different communication systems as we leave our 802.11 'hotspot'.

In the current system we have focused on room-scale systems, but as we move to city scale and beyond, precision and resolution will become a problem. We have used VRML to define a room coordinate system, but VRML itself only deals with Euclidean spaces, and it is not appropriate for very large-scale models. Fortunately this has been realized and the GeoVRML specification provides a way of transferring between multiple coordinate systems on a planetary scale [24]. GeoVRML allows the convenience to the author of working locally in standard, non-geographically aware packages, but then registering these to global models that use geodetic coordinates based on latitude and longitude.

We also are beginning to work with bodies of information that are larger and less under our control that at present, e.g. the web sites of other cultural institutions and tourist services, and 2D and 3D data describing the entire city center. At present, our models are relatively closed and static, but soon they will have to be more open and dynamic with consequent issues for transformation and consistency.

For this new system we will need to revise the authoring processes to reflect the modeling and run-time implications discussed. We aim to move towards system that can verify the consistency of models and transformations between them. This is especially important with our various symbolic models because the number of variety of different locations will massively increase, and it will no longer be possible to keep hand-modeling changes as new zones or web resources are added.

We will also need to reflect the errors and uncertainties in the system explicitly. As discussed in Section 6, at the moment the zones are quite large and separated by small distances so that collision of the box defining the user position with the boxes defining zones in room coordinates achieves the expected result and so that symbolic location does not change frequently when the ultrasonic device is on the boundary of a zone. We have thus tried to make the system more robust and resilient by reflecting error in the zone modeling. A more rigorous approach would be to reflect error in the collision process, and leave the zones as simple descriptions. This would enable us to estimate confidence in the updates to a user's symbolic location.

Our user trials as well as our observational studies of more traditional visits to museums and visits [4] showed some of the ways that subtle details of position, orientation, gesture and body language are used to coordinate movement and as comments on the exhibition. For example, a visitor looking at an artifact with another visitor may lean slightly move back to show that he or she is ready to move on. Body language would often be used to express a person's interest or opinion on an artifact. Users are represented simply and roughly, through position and orientation, and they largely rely on talk to overcome this. While we do not feel that detailed tracking of photo realistic rendering of gesture, stance etc. are necessary, we feel that future versions of our system could benefit from means to represent more of the degree and nature of a user's engagement with an artifact.

10. CONCLUSION

Mixed-reality systems demand mixed models of space. We have analyzed a mixed-reality system that supports physical visitors, web visitors and virtual environment visitors and we have shown how it supports 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 positions sensors and models based on symbolic associations between locations. We have claimed, that although not explicitly discussed, many 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 modeling 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 how we plan to tackle these errors and ambiguities when building a successor system that will support a much larger physical space.

ACKNOWLEDGEMENTS

<Anonymised>

REFERENCES

1. Abowd, G, Atkeson, C., Hong, J., Long, S., Kooper. R. & Pinkerton, M. Cyberguide: A mobile context-aware tour guide, Wireless Networks, 3 (1997), pp. 421-433.
2. AC3D, http://www.ac3d.org/
3. Angermann, M., Kammann, J., Robertson, P., Steingaß, 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. <anonymised>, Lessons from the lighthouse: Collaboration in a shared mixed reality system, submitted for publication.
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. In 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. In Ubiquitous Computing, Springer, 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. In Ubiquitous Computing,. Springer-Verlag, 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, published as Springer LNCS 2498.
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 2002).
13. Hindmarsh, J., Fraser, M., Heath, C., Benford, S., Greenhalgh, C. Fragmented Interaction: Establishing mutual orientation in virtual environments. Proceedings of CSCW'98. New York: ACM Press. pp. 217- 226.
14. Hindmarsh, J., Fraser, M., Heath, C., Benford, S. and Greenhalgh, C., Object-Focused Interaction in Collaborative Virtual Environments, in ACM Transactions on Computer-Human Interaction (ToCHI), 7 (4), December 2000, pp. 477-509, ACM Press
15. Hohl, F., U. Kubach, A. Leonhardi, Rothermel, K., and Schwehm, M. Nexus - an open global infrastructure for spatial-aware applications. In Proceedings of MobiCom, Seattle, USA, 1999.
16. 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.

17. <anonymised>, Shared Visiting in EQUATOR City, Proceedings of CVE2002, Bonn, Germany. ACM Press
18. Nord, J., Synnes, K., Parnes, P. An Architecture for Location Aware Applications, HICSS-35, Big Island, Hawai´i, USA, January 2002.
19. Ordnance Survey, A guide to coordinate systems in Great Britain, http://www.gps.gov.uk/guidecontents.asp (accessed 9th October 2002).
20. Pradhan, S. Semantic Location, Personal Technologies 4(4). Springer, 2000, pp 213-216.
21. 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.
22. Randell, C., Muller, H. Low cost indoor positioning system. In Ubiquitous Computing. Springer, 2001. pp. 42-48.
23. Randell, C., Muller., H. Exploring the dynamic measurement of position. In International Symposium on Wearable Computing 2000. IEEE.
24. Reddy, M., Iverson, L., GeoVRML 1.1 Specification, www.geovrml.org (accessed 9th October 2002).
25. Spohrer, J., Worldboard, what comes after the WWW? http://www.worldboard.org/pub/spohrer/wbconcept/default.html, 1997, (accessed 10th October 2002)
26. 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.
27. VRML97 International Standard (ISO/IEC 14772-1:1997)