Scalable Techniques for Collaborative Outdoor Augmented Reality

Gerhard Reitmayr and Dieter Schmalstieg
Vienna University of Technology
{reitmayr|schmalstieg}@ims.tuwien.ac.at

Abstract

Research on mobile augmented reality (AR) so far concentrated on building and deploying prototypes and investigating technical and user interface aspects. To create plausible application scenarios other concerns also need to be addressed. In particular, the size of the physical world made accessible by mobile AR raises new challenges in terms of scalability: the system will among others need a very large 3D world model, display this model efficiently without visual clutter or graphics overload, should operate in networked mode with arbitrary collaborators, provide tracking anywhere etc. In this paper we address some of these scalability issues, namely data management for very large geographic 3D models, and efficient techniques for collaborative outdoor user interfaces. As an application scenario, we use a tourist group guide application operating in the city of Vienna.

Keywords: Augmented Reality, Mobile Computing, CSCW, Data management, XML database, alligators.

1. Introduction

Mobile augmented reality (AR) applications have become a mainstream research topic in AR. Basic research revolved around building and deploying such systems and developing appropriate interaction techniques for mobile users. More complex issues such as collaboration were only explored in very task-specific settings such as one-to-one remote collaboration with a mobile user.

A central aspect of mobile AR systems is scalability. Such systems are expected to be used in a wide area. Consequently, a number of scalability issues related to the large physical space need to be investigated. These include collaboration between mobile users, data management for large scale applications, network architectures and communication protocols, ubiquitous tracking services allowing for transitions between tracking devices and user interface management. In this paper we investigate two of these issues, namely the requirement to support spontaneous collaboration among mobile users and the requirement for a scalable data management architecture serving a potentially large user base having diverse requirements.

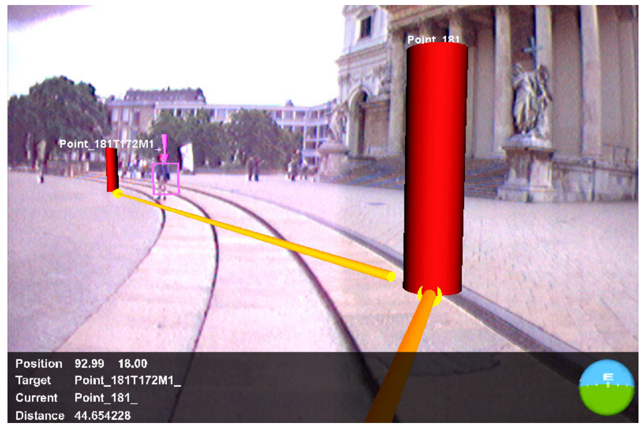

Figure 1. Collaborative navigation enables following another user who is indicated by the purple arrow and frame.

Collaboration is motivated by the fact that mobile users have a large set of potential persons to collaborate with. Interaction between mobile AR users will be the norm rather than the exception. We investigate the collaborative aspects of a typical AR application scenario, the tourist guide, and describe our extensions of the navigation and information browsing features to enable collaboration.

Working areas for mobile AR application will extend over large spaces such as buildings or throughout a city. If we assume a uniform density of interesting information over our working area, we can expect an increase in the total volume of data the application has to handle as the area increases. Moreover, some data items will change frequently, such as location information for users and mobile objects. Creating, updating and administrating such data volumes requires a distributed effort similar to the World Wide Web. Applications running on mobile AR systems will retrieve relevant information from a networked environment and play the role of subscribers to information providers. To this end we describe our three-tier architecture for managing data for mobile AR applications.

2. Related work

Navigation and information browsing are two common themes used in demonstrating wearable computing and mobile augmented reality. The Touring Machine [11], the work described by Thomas et al. [33], and the Context Compass [32] show how pedestrian navigation can benefit from heads-up displays and AR interfaces. Information browsing was first demonstrated in the Navicam project [24] and has since become a popular topic of AR applications. A notable example of a wearable tourist guide application is the GUIDE project [8] that presents tourists with location-based information as they roam through the city of Lancaster. Some work on information filtering [16] and label placement [4] addresses the issue of managing user interfaces for large numbers of data items, but from a user's point of view rather than that of the application.

Augmented reality has also been identified as an important means of computer supported collaborative work (CSCW) as demonstrated in the Shared Space project [5]. Collaboration between stationary and mobile users has been investigated by different groups in the MARS project [14] and Tinmith project [21]. Mobile users could also create world models in the BARS project [3] but only in well controlled indoor environments. The Human Pacman [6] is a collaborative AR game. The Augmented Stroll [26] provides mobile collaborative AR services to archeologists in their field work and comes closest to our vision of simple structured collaboration between several mobile users in an outdoor setting.

Typical AR demonstrations work with small data sets that have been entered manually and do not require data management. Consequently, there has been little work done on data management techniques for large AR models. Höllerer et. al [15] describe the use of a database and description logic based meta-data to store a model of a building floor which is annotated with meta-data for navigation target selection. The sentient computing project [2] uses a CORBA run-time infrastructure to model a live environment as distributed software objects where locations and attributes of objects are updated permanently. Newman et al. [20] describe a set of AR applications based on this infrastructure.

The Nexus project [27] is unique in that it specifically deals with the software architecture required for ubiquitous location-based applications and provides abstract interfaces for location data to such applications. Although the project does describe some preliminary augmented reality applications [12], it does not focus on AR applications interacting with complex information structures. Glonass [9] also describes a software architecture to distribute context information but does not provide the extensive models that advanced AR applications require.

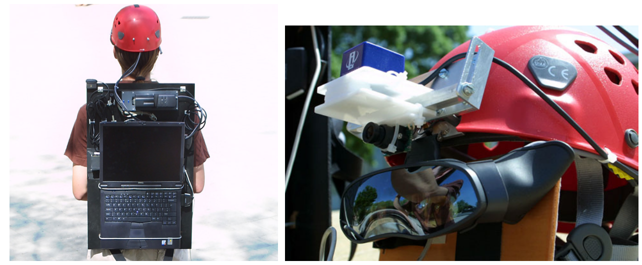

Figure 2. A laptop computer, GPS receiver and battery packs are mounted on a backpack. The HMD, inertial sensor and a camera are mounted on a helmet worn by the user.

The geographic information systems (GIS) community has a lot of experience with storing and manipulating large scale geometric data [29]. However, the current data sets are still mostly dealing with 2D features without complex interrelations. A current trend towards 3D models for communities such as the City of Vienna [10] will hopefully provide a better basis for mobile AR applications.

Earlier results on data management for AR have been described in two workshop contributions [22, 23].

3. Tourist guide application

The needs and requirements of a tourist are a suitable starting point for testing location-based applications. A tourist is typically a person with little or no knowledge of the environment. However, tourists have a strong interest in their environment and also want to navigate through their surroundings to visit different locations. Tourists tend to travel in groups, therefore collaboration is a natural requirement. Guided tours are also common practice and provide for more structured interaction between participants. In such a situation a single person directs a group of people and presents information.

We have chosen a tourist guide application for the City of Vienna as an example scenario for an augmented reality application that integrates a large amount of data from different sources. It provides a navigation aid that directs the user to a target location and an information browser that displays location referenced information icons that can be selected to present more detailed information in a variety of formats. Users can also author simple 3D-referenced annotations. All functions support collaboration between multiple mobile users.

3.1. Augmented reality system

The mobile AR setup uses a notebook computer with a 2GHz processor and an NVidia Quadro4Go graphics accelerator operating under Windows XP. It includes a wireless

LAN network adapter to enable communication with a second mobile unit. A Trimble Pathfinder Pocket differential GPS receiver is used to determine the position of the system in outdoor applications. All the equipment is mounted on a backpack worn by the user. We use a Sony Glasstron optical-see-through stereoscopic color HMD fixed to a helmet as an output device. An InterSense InertiaCube2 orientation sensor provides information on the direction the user is looking in, and a PointGrey Research Firefly camera mounted above the HMD is used for fiducial tracking and video see-through configurations. Both devices are mounted on a helmet worn by the user (see Figure 2).

We use Studierstube [28] which is based on Open Inventor (OIV) [31] as a software platform for developing AR applications. It provides a multi-user, multi-application environment, and supports a variety of display devices including stereoscopic HMDs. It also allows 6DOF interaction, either with virtual objects or with user interface elements displayed in the user's field of view. Applications are developed as scene graphs that can be described with the declarative OIV file format. Studierstube is a very capable rapid prototyping system, but by itself does not incorporate any database functions beyond a scene graph based runtime data structure.

Collaboration between different users requires distribution of application state among different setups. To simplify development of such distributed applications, we implemented Distributed Inventor [13] that provides shared memory semantics on the scene graph data structure. Changes to a distributed part of the scene graph are transparently communicated to other instances of the application without exposing this process to the application programmer.

3.2. User interface

The system presents information to the user on the head mounted display. This information is either presented as graphical objects rendered to fit into the natural environment or as text, images and 3D objects providing a heads-up display. The graphical objects are drawn to enhance and complement the user's perception of the natural environment. They can represent abstract information, alternative representations of real objects or highlighted real structures. The heads-up display is used to provide a graphical user interface consisting of typical 2D widgets such as buttons, lists, and text fields and to provide other status information. Figure 3 shows a typical view through the users display.

The user can control a cursor within the 2D user interface part in the upper right corner with a touch pad that is either worn on the belt or hand-held. She can switch between different modes of the application such as navigation, information browsing and annotation. Each mode presents a number of individual panes to provide control of parame-

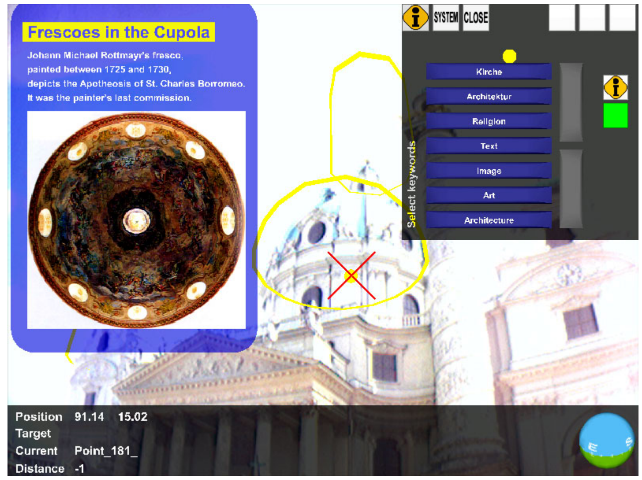

Figure 3. Video-overlay of the user's view. Real objects are highlighted in yellow with associated information displayed to the left. A panel on the right holds 2D user interface components and a heads-up display at the bottom shows additional information.

ters and other options related to the current task. A general heads-up display at the bottom of the view presents generic information such as the current location, selected target location, distance to the target and an orientation widget modeled after a compass.

Interaction with the environment either occurs implicitly using the user's location and orientation or explicitly using the viewing direction. Some functions use a ray-picking interaction that is directed by the viewing direction to select objects or determine placement of objects. In this case a cross-hair displayed in the heads-up display identifies the exact location of the intersection in screen space.

4. Collaborative applications

Three tasks are supported by the tourist guide application: Navigation, information browsing and annotation in the form of simple virtual graffiti. All functions are usable in a stand-alone mode by a single user. Additionally, they also support collaborative touring where two scenario situations are interesting. Either the members of a group are equal or they have a more structured setting with a single person directing the other group members. In the latter a tour guide of a group of tourists acts as an expert and is able to control the functions of the tourists' devices.

The functions allow for a rather passive, observation style of use. Users are presented with information hints and can opt to see more, if they are interested. Collaboration should not add more cognitive load or require more knowledge of how to use the system. Therefore, we took care to

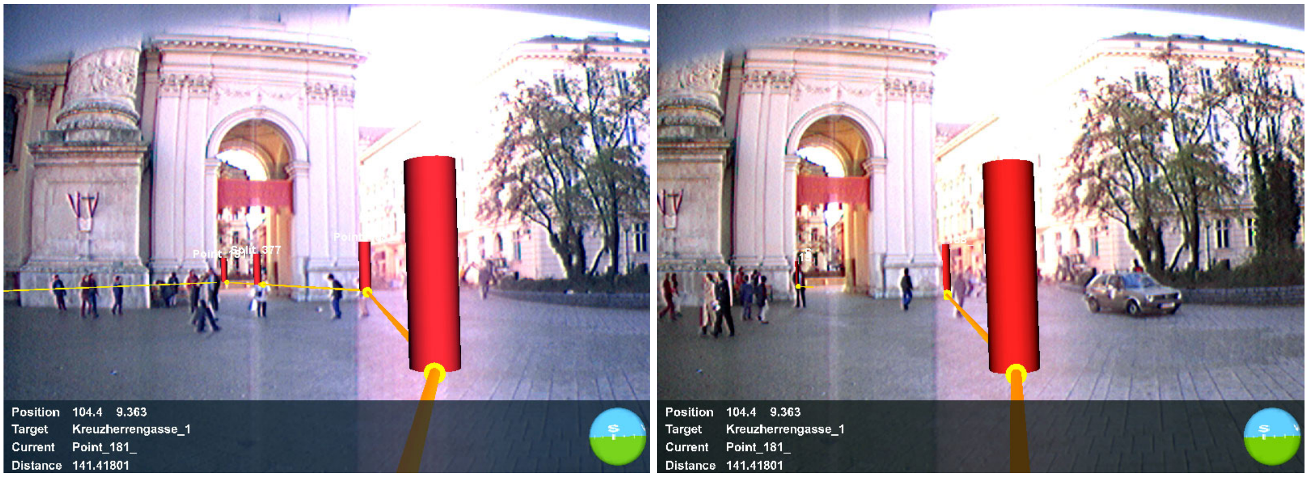

Figure 4. (Left) A visualization of the path to the selected target without clipping on known objects. (Right) The same path clipped at an object.

provide features that require only minimal adaption on the part of the user to be usable.

The general approach to supporting collaboration builds on sharing of user interface state. The displayed information for a single user depends on parameters set by the user and the current interaction (such as looking at an icon). Such information is encapsulated in a part of the scene graph which is distributed between the users' systems and therefore implicitly shared. Thus a user can set parameters of other participants simply by manipulating her own user interface. The "I set your parameters" style provides the required functionality and is simple to understand. Mechanisms to join or leave the session provide the required control of the individual user over her system. As results of interactions such as selections are also shared within a session, users will perceive the results of actions performed by other participants.

More complex interactions such as specific navigation modes require more support by the system. In this case it is important to identify the high-level goals of the users and provide explicit sharing of low-level data. For example, establishing a meeting point would require detailed knowledge of the waypoint network and the users location. Thus, the system should use its knowledge to compute the meeting point and only require input of the person the user wants to meet. A high-level interface tailored to a specific task is presented to the user, while still relying on the underlying graph search algorithm used for automated wayfinding.

4.1. Navigation

In the navigation mode the user selects a specific target address or a desired target location of a certain type such as a supermarket or a pharmacy. The system then computes the shortest path in a known network of possible routes. Moreover, it is interactive and reacts to the user's movements by continuously re-computing the shortest path to the target, if the user goes astray or decides to take another route.

The information is displayed as a series of waypoints that are visualized as cylinders standing in the environment. These cylinders are connected by arrows to show the direction the user should move between the waypoints. Together they become a visible line through the environment that is easy to follow (see Figure 4 left). The user can enable an additional arrow that points directly from her current position to the next waypoint. Buildings can clip the displayed geometry to enable additional depth perception cues between the virtual information and the real world (see Figure 4 right). Finally, simple directional information is displayed, if the user is not able to perceive the next waypoint because she is looking in another direction.

If two or more users are present, a number of collaborative interactions are possible. The interface presents a list of all users that have joined the collaborative session. Every user can select a partner and specify an interaction mode:

Follow The user can decide to follow the selected user. The navigation display will update the target location to always coincide with the waypoint closest to the selected user. Interactivity is now extended to include both the positions of the user herself as well as of the target user.

Guide The selected user can be guided by setting her destination point. The navigation system of the selected user will then behave as if that user had selected the target herself.

Meet The system also supports rendezvous with the selected user. The navigation system calculates the meeting point to be halfway between the two waypoints the users are closest to. The destinations of both users are then set

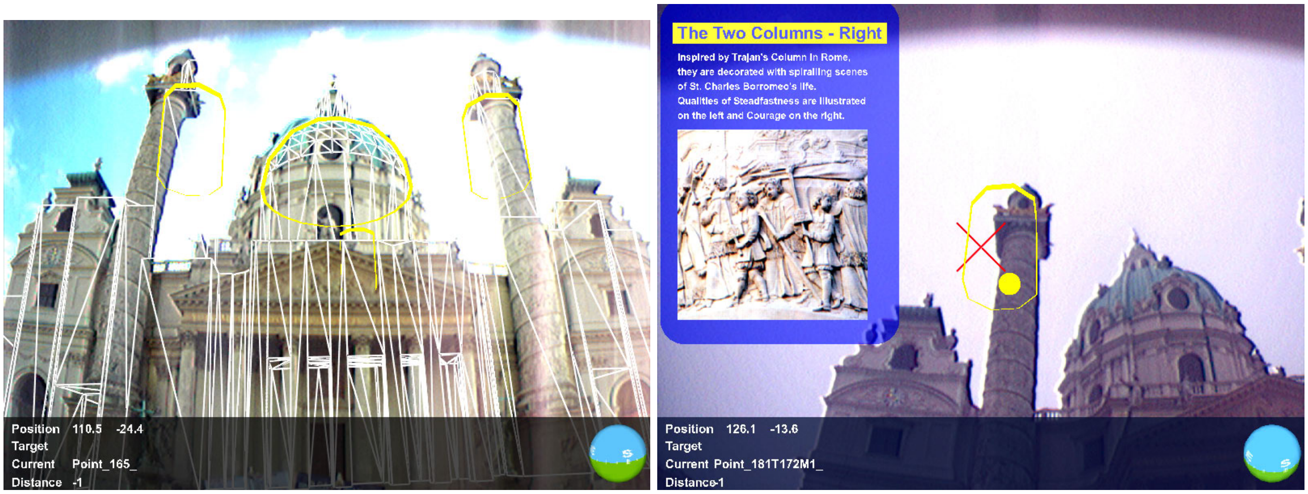

Figure 5. (Left) Different parts of the building are highlighted to show possible additional information. (Right) The user selects the column by looking at it and the content is displayed.

to this new target. Each user can still change the common target to a more suitable location if desired.

In meeting mode the meeting point stays fixed, even if both users deviate from the planned route and another point could provide a closer meeting place. Both users can still trigger recomputation of the meeting point by re-selecting their partner. Another possibility would be to interactively recompute the meeting place. However, we think that a fixed meeting point is closer to the user's mental model of the behavior of the application which aims at simplicity and should not surprise the user.

4.2. Information browsing

The information browsing mode presents the user with location-based information. Location-referenced information icons appear in her view and are selected by looking at them. Associated information is presented in a pop-up. The application conveys historical and cultural information about sights in the City of Vienna. For example, frescos on a building facade carry information on the sculptor who created them and the topic of the picture. Statues give information on the background of the person they are representing.

The information icons can have any shape for display as well as for ray intersection. In the current application we use geometric representations of building parts to annotate these with cultural information. The icons appear to the user as outlines of parts of the building. A virtual ray is cast through the center of the display and intersected with the information icons. The first icon that is hit triggers the display of information associated with it (see Figure 5).

A user might not be interested in all types of available information or may get overwhelmed by a large number of locations that trigger information displays. Therefore, objects and content are described with a set of keywords and the user can select interesting information by selecting a subset of these keywords. Only information matching the selected keywords will be displayed.

The information browsing mode supports multiple users. Users can choose to share their selection of topics, or alternatively, tour guides can control the selection for a group of guided users. A user can also trigger the highlighted information on another user's display.

Two toggle buttons *Listen* and *Guide* set a single user's interaction modes with the group. If the Guide mode is active the user's keyword selection and active icon will be shared with the group of participating users. If the Listen mode is active the shared keyword and active icon information of another user will be used to set the state of the local user's display. Therefore, the local user will see the same information content that the guiding user has selected and will also use the same keyword selection, if she decides to still browse the icons on her own.

4.3. Annotation

Users are able to contribute, in a simple way, to the available information. Annotation allows users to place graphical icons with different colors and shapes in the environment. To simplify the interaction for determining the position, icons are always placed on the surface of existing buildings, which reduces the necessary manipulation to specifying a direction from the user's current position. The icon is oriented parallel to the tangent plane at the intersection point (see Figure 6). The user can select predefined shapes and colors and can also choose which kinds of icons to display. More interesting data could include text, audio or video annotation to realize at an Augmentable Reality

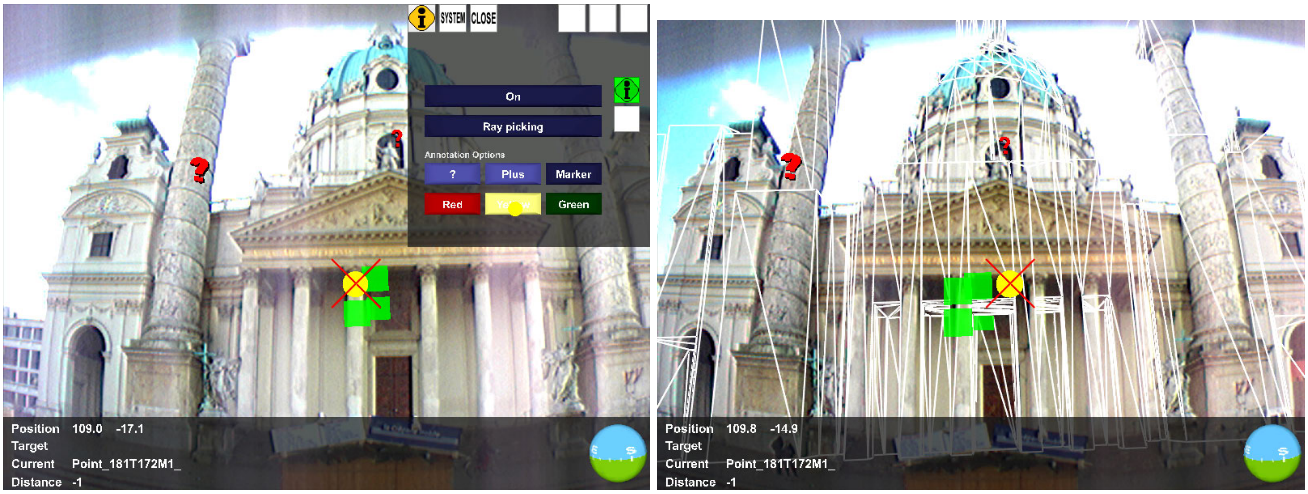

Figure 6. (Left) Annotation options and icons. (Right) The building geometry used for ray intersection is overlaid.

[25]. While such features could have been to included in our system, they were not the focus of our work.

A group of users can share the different icons that each participant creates, so that each user can see them. Again, to reduce possible clutter, a filtering option allows users to restrict which icons are visible to them. Possible criteria are the icon's creator, color and shape.

The virtual markers can help to point out features on distant structures such as building facades. Users can attach and discern different meanings associated with markers by assigning different styles. They support collaborative work styles because they are shared information and can help users to communicate information concerning individual locations in the surrounding.

The annotation mode only supports symmetric roles in the collaboration because it is intended to be a simple feature for unstructured work styles and groups.

5. Data management for AR

The implementation of the tour-guide functions requires extensive environment data and information to be presented to the user. Accurate and complete models of the buildings and other obstacles in the environment are required for rendering occlusions, highlights of buildings and 3D interaction with the buildings surfaces. Additional models mark the active regions of objects which trigger information displays. The navigation mode requires a network of paths and waypoints and address information related to the buildings. Meta-data concerning address points and historical information is also required and should be linked to the appropriate models.

Different functions may be based on the same data but require this data in different forms which are not necessarily only different subsets of the complete data set. However, authoring and managing of such data sets benefits from a centralized location where such activities can be aggregated. In the envisioned scenario most data and content will be provided by dedicated providers and not created by the users themselves. Therefore, general updates to the model by the mobile systems themselves were not required. We propose a dedicated architecture to resolve the trade-off between supporting multiple applications and data management.

Our architectural concept is based on a 3-tier model [1]. The first tier is a central database storing the overall model. The second tier mediates between database and application by converting the general model from the database into data structures native to the respective application. The client application is the third tier and only deals with data structures in its own native format.

A common data model and data store reduces the amount of redundancy in the stored data required for different applications and allows centralized and efficient management of this data. The middle layer separates presentation from data storage and applications. Thus the applications can be developed using efficient data structures independent of the actual storage format. Moreover, the transformation can be adapted to changing data formats without affecting either the application or the storage back-end, because it is a distinct and separated entity.

Our architecture differs from the traditional 3-tier model for client-server applications in some aspects. The second tier is somewhat passive and focuses only on data manipulation and will not provide extensive application functionality. The argument for a thin client offering only user interface functions does not hold in our architecture, because the client needs to provide interactive realtime feedback. Calculations by the application need to be available to the user interface instantaneously, which is another driving force in the unification of application data and graphical data in the scene graph data structure.

We propose to use this architecture building upon XML technology, thus leveraging recent developments in the web application community. The proposed architecture is very common in this area and directly supported in a number of products, either open-source or proprietary. The use of XML has a number of advantages:

- A hierarchical data model fits well to our general spatial model. Rather than using a flat enumeration of building representations, a hierarchical model can represent several levels of a spatial hierarchy, from districts and streets down to rooms within buildings and other detailed geometrical data.

- While a document and file-oriented approach is generally sufficient for research prototyping, it obviously lacks scalability. More powerful storage solutions are required for real applications. Some of these exist in the form of XML databases [19, 30]. As XML technology is generally aimed at compatibility, the tools and APIs used for prototyping are directly supported by commercial XML products, and the transfer to a production system is greatly simplified.

- XML tools such as XSLT [7] allow rapid prototyping and development of import, transformation and export tools to and from the data model. Such tools focus on the functional aspect of the transformations and reduce the overhead in implementing parsers and generating data structures.

- Parsers and generators exist for a wide range of programming languages, and allow applications and tools to use the most appropriate language for the task.

- Standards for meaningful descriptions of data exist, on a syntactic level such as RDF [17] as well as on a semantic level for meta-data such as the Web Ontology Language [18] or the Dublin Core [34]. This allows the definition and use of ontologies to support semantically rich queries and interactions. While we have not used such techniques yet, we plan to investigate the possibility of building complex models based on real ontological descriptions.

5.1. Modeling

At the heart of our architecture lies a data model that encompasses all application requirements. Care was taken in keeping the model extensible so that new data could be integrated during the development. This data model is described by an XML schema.

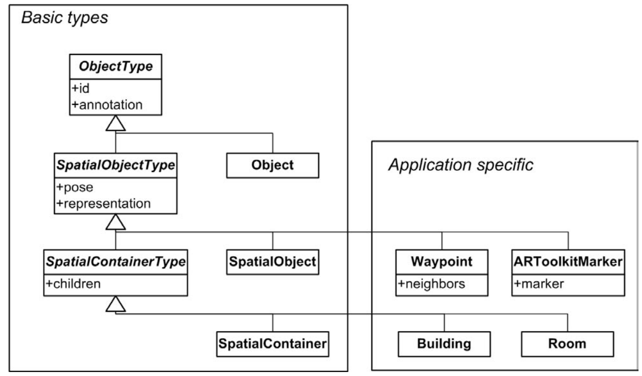

Figure 7. Overview of the type hierarchy in the model schema. A set of basic types can be used for general modelling. Applications may derive additional types for specific requirements.

The model should fulfill a number of key requirements:

- Geometric representations and hierarchies need to be stored in the model.

- Interaction with other data schemas should be possible to maximize reuse of already established knowledge presented in the form of these schemas.

- Extensibility for new applications and data types with fall-back options for generic processing is important.

The model is based on an object-oriented approach using a type hierarchy to define a taxonomy of objects. The root type is called *ObjectType* and contains an id and a generic annotation subelement that can be used to store any XML tree. All data types defined in the model are derived from this type. The *SpatialObjectType* adds pose information and a geometrical representation to the super class. We further derive the *SpatialContainerType* that adds a child subelement to aggregate entities of type *ObjectType* for hierarchical composition.

From the three base types, we derive a number of application specific types that are used in the actual data files. The *Object*, *SpatialObject* and *SpatialContainer* elements are used for general purpose data and correspond directly to the base types (see Figure 7). Applications can define additional types derived from the base types to provide more specific information. For example, we define a special *Waypoint* element used by an outdoor navigation application which has a specific subelement to define neighboring waypoints connected by a path. Because elements refer back to their base type, an application can always provide a reasonable fall back behavior if it encounters an unknown derived application element. The Nexus project [27] uses a similar structure to model their data types.

The XML tree is interpreted in the standard geometrical way, by defining a child's pose relative to its parent. We chose this mapping to support conventional modeling of visual data as trees. However, the open XML based format is not bound to any particular visualization tool or platform, and affords the definition of other than spatial relations by using relational techniques such as referring to object ids.

The annotation subelement of the abstract root type can be used to model free form data or to augment pre-existing types with extra information. This allows more flexible technologies to annotate the objects in our model.

5.2. Data manipulation and retrieval

Having defined a model and data format, there are a number of tasks and tools necessary to fill the database, transform and manipulate the data and finally make it available to the user by developing appropriate applications. The typical tasks include the following:

Import Extract information from source data formats based on XML or other formats and map it to the data model. Non-XML source formats also require the combination of an appropriate parser with an XML generator to map the foreign format to XML.

Maintenance Maintaining a model requires the application of filters and transformations on stored data.

Export Transformations are applied to retrieve relevant application data from storage and generate data structures for the applications. Each application uses a custom XML stylesheet to directly generate the required scene graph and additional data structures described in the Open Inventor file format from the general model.

The use of a separate step to transform data into the application format has a number of advantages. It separates the general data format from the application specific data structures and provides an interface between the two. It also provides a point for customizing the presentation independently of the application, similar to the way traditional cascading stylesheets work for HTML. As the stylesheet generates the actual graphical content, it can adapt to different output requirements or profiles.

5.3. Implementation

The implementation of the architecture is based on a commercial XML database [30]. We configured a manipulation layer as part of an HTTP server within a servlet engine which would execute queries to the database and perform XSLT transformations on the result sets. The transformed data is transmitted back in response to the query.

Figure 8. A subset of the 3D model of the City of Vienna. It includes a digital elevation model, building blocks and roofs. 3D model curtesy of Vienna City Administration.

Furthermore the result set can be transformed on the client side by executing additional transformations. Either the transformations already produces an Open Inventor scene graph used to configure an application, or one is created from the resulting XML document via a standard mapping.

5.4. Data acquisition

A general model was built to serve as the basis for all tourist guide functions. A 3D model of part of Vienna was kindly provided by the city administration of Vienna (see Figure 8), and is part of the official general map of the city [10]. The model itself was delivered in VRML97 format which we converted to the Open Inventor ASCII file format. Then we developed an import tool that read the Open Inventor file and constructed a representation in the model format, which was then used as the base for the overall model.

The department of Geoinformatics at Vienna University of Technology supplied a network of accessible routes for pedestrians, delivered in GML2 format (an XML based file format used in the GIS community). This model was derived from the general map of Vienna and is represented as an undirected graph. Each node in this graph is georeferenced and used as a way point in the navigation model. For each building, an "address point" was defined and incorporated into the path network to construct a path to this address. As navigation graph data was available in an XML-based format, a simple XSLT transformation script was sufficient to incorporate this data into the model.

Furthermore, annotation information such as businesses located at certain addresses was derived from the general map of Vienna. This information is connected to address points in the spatial database.

It was necessary to compute the intersection of the 3D model data and the navigation graph, as the relevant input data was derived from two overlapping sections of the city map. This was achieved by computing a subset of the model within a given bounding box and then repairing the internal structures of the navigation graph to ensure data coherence after clipping. The maintenance tool directly reads from and writes to the common data model.

Finally, we placed the icons as spatial representations of interesting information into our model. Cultural information taken from a guide book was included at various places to provide the detailed data for the information browsing component.

6. Discussion

While our work deals with collaboration and data management, a number of other scalability issues were not addressed. Information displays are prone to visual clutter and a number of solution described in the related work [16, 4] can offer improvement.

The proposed architecture and set of tools are based on established techniques in contemporary computer science. However, we have found little evidence of the use of scalable data management techniques in large scale AR. There is a small set of related areas where applicable techniques are used. GIS databases are primarily aimed at 2D information and static, stationary use cases. Location-based services are aimed at abstract, text-oriented information and low-end devices. Visual simulation focuses on high-fidelity graphical representation and stationary displays. In this work, we have attempted to explore the combination of aspects of these areas into a scalable mobile AR database system.

Today's most common technology for storing data are relational database management systems (RDBMS). However, we did not base our approach on a relational data model for several reasons. The hierarchical structure inherent in 3D data models is more directly mapped to an XML structure. One of the great advantages of XML technology is its self-description and self-organization by using schemas and namespaces. AR applications benefit from this flexibility, because data definitions can be decoupled by domain and developed independently without breaking application compatibility.

We did not address on-the-fly database updating by mobile applications. The application presented was tailored to a browsing style of interaction and we believe that this will be the most common mode for data presented in such a system. There will be dedicated authoring tools that have more knowledge about the data model and will be used by professionals to maintain the model. However, individual applications will benefit from specific update functions. For example, the annotation icons should be persistent to create a more interesting application. We will continue to investigate the inclusion of such update functionality starting with use cases such as annotation.

7. Conclusions and future work

The presented work provides an outlook into possible future user interfaces for mobile collaborative augmented reality applications. Although many of the implemented user interface features have been demonstrated before, our work exceeds former work in two areas. Firstly, the collaboration features add another dimension to the possibilities of such systems by supporting groups of users in their tasks. While the features of the system are implemented, we need more tests of the collaborative aspects to determine useful and interesting extensions. Secondly, an integrated approach to handling the data required by mobile AR applications allows the system to scale to environments of realistic size. The use of a flexible data model also simplifies extension of the system by future applications that will have new requirements for data to be stored in the repository.

Future work will revisit the design of the model schema to investigate the more expressive possibilities of semantic webs for modeling the required data. Such a model could also form the basis of simulation environments for location-based and ubiquitous computing. Update functionality from live applications to the database are also required for persistent changes by the users.

Acknowledgments

This work was sponsored by the Austrian Science Foundation *FWF* under contracts no. P14470-N04 and START Y193, and Vienna University of Technology by an infrastructure lab grant ("MARDIS"). We would like to thank the Vienna City Administration for providing the 3D model of Vienna and Markus Probst for working on it, Prof. Stephan Winter for the navigation model and Joe Newman for setting up the DGPS system.

References

[1] A. Aarsten, D. Brugali, and G. Menga. Patterns for three-tier client/server applications. In *Proc. PLoP'96*, Allerton Park, Illinois, USA, September 4–6 1996. Addison-Wesley.

[2] M. Addlesee, R. Curwen, S. Hodges, A. Hopper, J. Newman, P. Steggles, and A. Ward. A sentient computing system. *IEEE Computer: Location-Aware Computing*, August 2001.

[3] Y. Baillot, D. Brown, and S. Julier. Authoring of physical models using mobile computers. In *Proc. ISWC 2001*, pages 39–46, 2001.

[4] B. Bell, S. Feiner, and T. Höllerer. View management for virtual and augmented reality. In *Proc. UIST'01*, pages 101–110, Orlando, Florida, USA, November 11–14 2001. ACM.

[5] M. Billinghurst, S. Weghorst, and T. Furness. Shared space: An augmented reality interface for computer supported collaborative work. In *Proc. of Collaborative Virtual Environments Workshop '96*, Nottingham, Great Britain, September 19–20 1996.

[6] A. D. Cheok, S. W. Fong, K. H. Goh, X. Yang, W. Liu, and F. Farbiz. Human pacman: A mobile entertainment system with ubiquitous computing and tangible interaction over a wide outdoor area. In *Proc. Mobile HCI 2003*, volume 2795 of *LNCS*, pages 209–223, Udine, Italy, September 8–11 2003. ACM, Springer-Verlag.

[7] J. Clark. XSL transformations (XSLT) version 1.0. http://www.w3.org/TR/xslt, 1999.

[8] N. Davies, K. Cheverest, K. Mitchell, and A. Friday. 'caches in the air': Disseminating tourist information in the guide system. In *Proc. WMCSA'99*, New Orleans, LA, USA, February 25–26 1999. IEEE.

[9] A. Dearle, G. N. Kirby, R. Morrison, A. McCarthy, K. Mullen, Y. Yang, R. C. Connor, P. Welen, and A. Wilson. Architectural support for global smart spaces. In *LNCS 2574*, pages 153–164. Springer, 2003.

[10] L. Dorffner and A. Zöchling. Das 3D modell von wien - erzeugung und Fortführung auf basis der wiener mehrzweckkarte. In *Proc. CORP 2003*, pages 161 – 166, Vienna, Austria, February 25 – March 1 2003.

[11] S. Feiner, B. MacIntyre, T. Höllerer, and A. Webster. A touring machine: Prototyping 3D mobile augmented reality systems for exploring the urban enviroment. In *Proc. ISWC'97*, pages 74–81, Cambridge, MA, USA, October 13–14 1997.

[12] N. Haala, J. Böhm, and M. Kada. Processing of 3d building models for location aware applications. In *International Archives on Photogrammetry and Remote Sensing IAPRS*, volume XXXIV, pages 157–162, Xian, August 2002. ISPRS Commision II Symposium.

[13] G. Hesina, D. Schmalstieg, and W. Purgathofer. Distributed open inventor : A practical approach to distributed 3D graphics. In *Proc. ACM VRST'99*, pages 74–81, London, UK, December 1999.

[14] T. Höllerer, S. Feiner, T. Terauchi, G. Rashid, and D. Hallaway. Exploring MARS: developing indoor and outdoor user interfaces to a mobile augmented reality system. *Computer & Graphics*, 23(6):779–785, 1999.

[15] T. Höllerer, D. Hallaway, N. Tinna, and S. Feiner. Steps toward accommodating variable position tracking accuracy in a mobile augmented reality system. In *Proc. AIMS'01*, pages 31–37, Seattle, WA, USA, Aug 4. 2001.

[16] S. Julier, M. Lanzagorta, Y. Baillot, L. Rosenblum, S. Feiner, and T. Höllerer. Information filtering for mobile augmented reality. In *Proc. ISAR 2000*, pages 3–11, Munich, Germany, October 5–6 2000. IEEE and ACM.

[17] F. Manola and E. Miller. Rdf primer. http://www.w3c.org/TR/rdf-primer/, May 3 2003.

[18] D. L. McGuinness and F. van Harmelen. OWL web ontology language overview. http://www.w3.org/TR/2004/REC-owl-features-20040210/, February 11th 2004.

[19] W. Meier. eXist: An open source native XML database. In A. B. Chaudri, M. Jeckle, E. Rahm, and R. Unland, editors, *Web, Web-Services, and Database Systems. NODe 2002 Web- and Database-Related Workshops*, Springer LNCS Series, page 2593, Erfurt, Germany, October 2002. Springer.

[20] J. Newman, D. Ingram, and A. Hopper. Augmented reality in a wide area sentient environment. In *Proc. ISAR 2001*, pages 77–86, New York, New York, USA, October 29–30 2001. IEEE and ACM.

[21] W. Piekarski, B. Gunther, and B. H. Thomas. Integrating virtual and augmented realities in an outdoor application. In *Proc. IWAR'99*, pages 45–54, San Francisco, CA, USA, October 21–22 1999. IEEE CS.

[22] G. Reitmayr and D. Schmalstieg. Data management strategies for mobile augmented reality. In *Proc. STARS 2003*, pages 47–52, Tokyo, Japan, October 7 2003.

[23] G. Reitmayr and D. Schmalstieg. Collaborative augmented reality for outdoor navigation and information browsing. In *Proc. Symposium Location Based Services and TeleCartography - Geowissenschaftliche Mitteilungen*, volume 66, Vienna, Austria, January 28–29 2004.

[24] J. Rekimoto. Navicam: A magnifying glass approach to augmented reality. *PRESENCE - Teleoperators and Virtual Environments*, 6(4):399–412, August 1997.

[25] J. Rekimoto, Y. Ayatsuka, and K. Hayashi. Augment-able reality: Situated communication through physical and digital spaces. In *Proc. ISWC'98*, Pittsburgh, Penn., USA, October 19–20 1998. IEEE and ACM.

[26] P. Renevier and L. Nigay. Mobile collaborative augmented reality: the augmented stroll. In *Proc. EHCI'01*, volume 2254 of *LNCS*, pages 299 – 316, Toronoto, Canada, May 2001. Springer-Verlag.

[27] K. Rothermel and A. Leonhardi. Maintaining world models for context-aware applications. In W. R. Wells, editor, *Proceedings of the 8th International Conference on Advances in Communication and Control (COMCON-8)*, 2001.

[28] D. Schmalstieg, A. Fuhrmann, G. Hesina, Z. Szalavari, L. M. Encarnao, M. Gervautz, and W. Purgathofer. The Studierstube augmented reality project. *PRESENCE - Teleoperators and Virtual Environments*, 11(1), 2002.

[29] S. Shekhar, M. Coyle, B. Goyal, D.-R. Liu, and S. Sarkar. Data models in geographic information systems. *Communications of the ACM*, 40(4):103–111, 1997.

[30] Software AG. Tamino XML server. http://www.softwareag.com/tamino/, January 5th 2004.

[31] P. Strauss and R. Carey. An object oriented 3D graphics toolkit. In *Proc, ACM SIGGRAPH'92*. ACM, 1992.

[32] R. Suomela and J. Lehikoinen. Context compass. In *Proc. ISWC 2000*, pages 147–154, Atlanta, Georgia, USA, October 16–17 2000. IEEE.

[33] B. H. Thomas, V. Demczuk, W. Piekarski, D. H. epworth, and B. Gunther. A wearable computer system with augmented reality to support terrestrial navigation. In *Proc. ISWC'98*, pages 168–171, Pittsburgh, USA, October 19–20 1998. IEEE and ACM.

[34] S. L. Weibel and T. Koch. The dublin core metadata initiative: Mission, current activities, and future directions. *D-Lib Magazine*, 6(12), December 2000.