Central European Multimedia and Virtual Reality Conference (2005)
J. Zara, J.Sloup (Editors)

The World as a User Interface: Augmented Reality for Ubiquitous Computing

Dieter Schmalstieg
Graz University of Technology
schmalstieg@icg.tu-graz.ac.at

Gerhard Reitmayr
University of Cambridge
gr281@cam.ac.uk

Abstract

We discuss the possibilities of augmented reality (AR) as a ubiquitous user interface to the real world. A mobile AR system can constantly provide guidance to its user through visual annotation of the physical environment. The first part of the paper discusses the necessary ingredients for ubiquitous AR, on which we have worked in the recent past, namely mobile AR hardware, wide area tracking, unobtrusive user interfaces, application prototypes, and geographic data models suitable for AR. The second part of the paper examines future requirements of such data models in greater detail. Based on the lessons learned in our previous work, we identify shortcoming of existing standards for geographic information systems and visualization models. Ubiquitous AR requires independence of the data model from specific applications and their implicit assumptions. A semantic network model of geo-referenced data provides such a data model. We examine how such a model fits the requirements of AR applications, and how it can be implemented in practice.

Categories and Subject Descriptors (according to ACM CCS) I.3.6 [Computer Graphics]: Graphics data structures and data types H.5.1 [Information Systems]: Artificial, augmented, and virtual realities

1. Introduction

Augmented reality (AR) is an excellent user interface for mobile computing applications, because it allows intuitive information browsing of location-referenced information. In an AR environment, the user's perception of the real world is enhanced by computer-generated entities such as 3D objects and spatialized audio [Azu97]. Interaction with these entities occurs in real-time providing convincing feedback to the user and giving the impression of natural interaction. Augmented reality as a user interface becomes particularly powerful when the computer has access to location-based information so that it can seemingly merge virtual entities with real world objects in a convincing manner.

Over the last years, we have created a mobile augmented reality system and a set of applications to gather experience of ubiquitous augmented reality applications. We focused on navigation and created solutions for both indoor and outdoor navigation. Both applications require extensive 3D models and information which is presented to the user. Accurate and complete models of buildings and their interiors are required for rendering occlusions and highlights of buildings. Different navigation models for indoor or outdoor use were developed to fit specific requirements.

The first part of this paper summarizes our work on mobile indoor and outdoor AR: presenting various hardware platforms for mobile computing and graphics; hybrid indoor/outdoor tracking solutions; user interface considerations and test applications. The second part introduces the concept of a semantic world model for AR, which directly derives from an analysis of the requirements of applying AR techniques within ubiquitous computing applications. We review recent developments in the geographic information systems (GIS) community, and how they can be used by mobile AR systems. As part of this approach, we present a conceptual design for a semantic network for AR, which can serve as a computational back end providing a new level of contextual information for AR and other types of ubiquitous computing services.

Fig. 1: The backpack system is based on a conventional notebook computer displaying stereoscopic graphics overlays in an optical see-through head mounted display.

Fig. 2: This handheld AR platform is based on a mini tablet PC that is operated as a video-see through "magic lens" device. It currently supports five different tracking technologies (3 shown) and weighs less than 1.5kg.

Fig. 3: The smallest fully featured AR platform to date is based on commercial PocketPC 2003 handheld computers. Using an embedded camera, these devices can perform optical marker tracking and real-time 3D graphics at ~20fps. Shown is the "Invisible Train" multi-user game successfully demonstrated at SIGGRAPH 2004.

2. Mobile augmented reality platform

2.1. Hardware and software

The successful delivery of mobile Augmented Reality (AR) is an ongoing challenge, as interactive 3D applications must be implemented on constrained hardware platforms, requiring tracking over a large area of operation at high accuracy. While earlier work on mobile AR uses backpack prototypes, e.g. [FMH*97] or [TDP*98], more recently there has been a trend towards smaller, discreet, lightweight handheld setups based on PDAs and cell phones. We have assembled a series of experimental platforms of varying form factor and capability for real-world testing of the trade-offs in wearable AR.

Our implementation of a classic backpack solution (Fig. 1) involved a frame equipped with a notebook computer running a standard operating system (specifically a Dell Precision 8100, 2GHz P4 CPU, NVidia Quadro4Go, Windows XP). This standard platform provides standard interfaces for communication (802.11 wireless LAN, GPRS) and peripherals. A differential GPS receiver (Trimble Pathfinder Pocket or Garmin GPSMouse) is used to determine the position of the system in outdoor applications. We use an optical-see-through stereoscopic color head mounted display (Sony Glasstron D100-BE) fixed to a helmet as an output device. The helmet also supports an inertial orientation tracker (InterSense InertiaCube2) and a camera (PointGrey Firefly) for indoor fiducial tracking and video see-through configurations [RS04].

While the backpack system uses very powerful hardware and allows development directly on the target platform, its weight and ergonomic properties are clearly unsatisfactory. We are therefore also experimenting with smaller handheld computer platforms, which allow a "magic lens" style of video see-through augmentation. Such a handheld AR platform is inexpensive and ergonomically superior to the backpack solution. Most potential users are already familiar with camcorders and consequently understand the handling (hand-eye coordination) of a handheld video-see through device. Informally we have observed that users prefer handheld AR over head mounted displays despite the lack of stereoscopic graphics and hands-free operation. The lower computational power of handhelds is partially compensated by the reduction in graphical complexity: monoscopic rather than stereoscopic; smaller screens, increased tolerance of lower resolutions and frame rates.

We have developed two handheld setups. The first setup (Fig. 2) is based on a mini tablet PC (Sony Vaio U70, 1GHz Pentium-M, Windows XP). This platform combines a regular PC compatible computer with several peripherals into a very compact form factor (footprint 15x20cm, 1400g including peripherals). The second setup (Fig. 3) is based on the Pocket PC standard for personal digital assistant (PDA) computers (ARM9 CPU currently attaining maximum speeds of 624MHz, Windows CE). While these devices weigh only around the 180g mark, they still feature a touch screen and a built-in camera. We have managed to implement real-time optical tracking and 3D graphics on the Pocket PC platform [WS03]. In terms of weight, size, and price these devices are almost ideally suited for our purposes, but software development for PDA operating systems is still not a straightforward task.

The software framework that allows rapid prototyping of AR applications with a high degree of 3D interaction is *Studierstube*, a versatile environment for developing virtual reality and AR applications [SFH*02]. It is based on an object-oriented scene graph (Coin), which allows the description of 3D scenes and 3D interactions through convenient declarative scripting.

2.2. User Interface

The main use of a mobile AR device is as an information appliance operated in browsing mode. The system should provide context-sensitive cues while the user is busy performing a task or navigating through the environment. Consequently, most input to the system should be automatically derived from situational context, without requiring explicit user attention. The main method for achieving this is by tracking the user's position in the environment, and the user's current viewing direction. Consequently, wide area tracking is of major importance for a mobile AR system (see next section).

However, applications will still require a certain amount of direct control. For example, a user may want to select a navigation target from a list of addresses or a map, or if the destination is within visible range more directly by using gaze direction. For such explicit control, we have investigated a number of interface alternatives.

Touch pad and touch screen: handheld AR devices are already equipped with a touch screen, which can be conveniently used to display on-screen menus operated by stylus or a finger (Fig. 1). The same touch screen can be used for selecting objects in the video-see through display by tapping on their position on the screen, effectively a form of raypicking interaction. For configurations using a head-mounted display, we have relied on either a handheld touchpad peripheral that is used to control a 2D cursor in the heads-up display, or by an additional PDA which can display menus directly on its screen. Both the touchpad and the PDA can be tracked using fiducial markers observed by the helmet-mounted camera, if 3D input is desired.

While the touch screen interaction clearly hints at its origins in desktop 2D and 3D user interfaces, the iOrb device (Fig. 4) was specifically designed for unified command and spatial input for mobile AR [RCK*05]. It consists of a single 3DOF inertial tracker (XSens MT9) embedded in a shell composed of two hemispheres of about 8cm diameter. By turning the sphere and then pressing the two hemispheres together, the user can issue application commands using variants of 1D and 2D pie menu techniques. Similarly, spatial selections can be made using a picking ray or cone. All interactions use only relative rotational measurements and are therefore mostly insensitive to measurement inaccuracies and drifting.

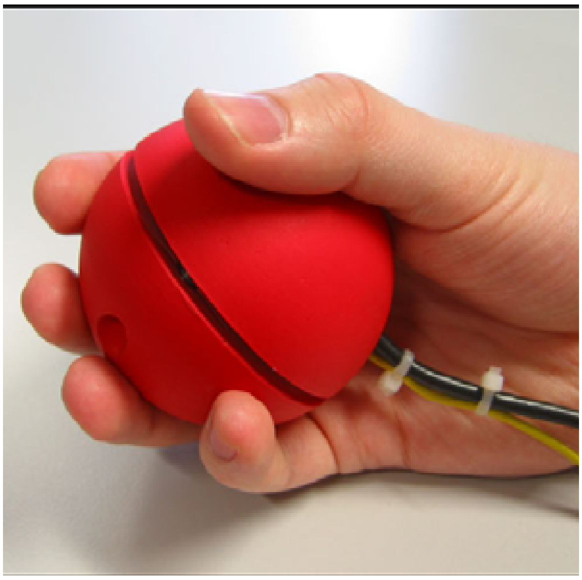

Fig. 4: The iOrb, an interaction device specifically designed for mobile AR users, unifies command input and 3D spatial selection at a distance through raypicking.

2.3. Tracking infrastructure

Wide area tracking cannot generally be done with a single sensor, because no single tracking technology can provide the range and accuracy required by a general mobile AR system. Therefore a significant body of work exists on hybrid tracking systems, combining multiple tracking technologies through sensor fusion techniques. However, most research focuses on building a single, improved tracking system through sensor fusion, rather than on alternating between different sources depending on availability, location, and context. Notable exceptions are the systems built by Hallaway et al. [HHF*04] and by Piekarski et al. [PAT*04], which are the only systems we are aware of that are capable of alternating between indoor and outdoor operation.

Several previous approaches combine multiple sensors popular for AR setups, such as fiducial tracking, inertial and GPS sensors. However, these approaches lack a general approach for management of arbitrary sensors. We have therefore developed a ubiquitous tracking framework [NWB*04], that addresses the problem of tracker integration and arbitrary sensor management. The most recent integrated solution that uses this approach executes on a handheld computer (U70), and combines five tracking systems: Inertial (XSens MT9), infrared vision (ARTTrack), magnetic (Flock of Birds), optical (ARToolKit), and GPS (Garmin GPS18 USB).

Each tracking technology has a dedicated working volume, with the exception of the inertial orientation tracker. The inertial tracker is used to assist other tracking systems with dead reckoning information, in particular the outdoor GPS system, which does not deliver any estimates of orientation.

As the core tracking software, we use OpenTracker (OT), which implements a pipes-and-filters network for connecting producers and consumers of tracking information [RS01]. The nodes of this network can execute on different hosts in a network. In particular, ARTTrack and Flock of Birds are stationary systems with a dedicated device server, each executing an instance of OT. The OT server will then communicate the tracking information to another instance of OT on the handheld computer over a wireless network.

Outdoor tracking: information is provided by a GPS receiver with the XSens inertial tracker providing complementary orientation estimates. Differential GPS corrections are obtained through a wireless internet connection from a local base station service or the new global correction service.

ARTTrack is a commercial multi-camera system capable of tracking target bodies composed of 4-5 small retro-reflective balls. These lightweight target bodies can easily be mounted on a helmet or handheld device, which can then be tracked in an outside-in mode while in view of the cameras. The mobile device itself is completely passive and does not require batteries or tethered cables for tracking. The pose data is transmitted from a stationary tracking server that performs the online pose estimation to the mobile system using wireless networking.

Flock of Birds: Since the Bird is wired, when the user with the handheld enters the effective envelope of the Bird, it is necessary to physically attach the Bird to the object of interest using velcro straps. Currently, the user triggers an event, by pressing a button, to acknowledge the presence of the Bird. Future versions will automatically determine Bird activity using correlated motion from an inertial tracker permanently mounted to the handheld device and the Bird, once it is attached to the handheld device. Similarly to the ARTTrack, the tracking data is transmitted wirelessly from a stationary server to the mobile device.

ARToolKit: A significantly modified version of the popular vision tracking library *ARToolKit* is used for tracking indoor regions beyond the reach of the magnetic and infrared technologies. Pose estimation of the mobile camera rigidly mounted on a helmet or handheld device is performed using the 2D location of the corners of one or several square markers visible in the camera image. The identity of the markers is then decoded directly from self-correcting 2D barcodes in the marker's interior area, which makes it possible to uniquely discriminate a large number of markers dispersed throughout the indoor environment. By looking up the geometric position of each observed marker in a previously surveyed model of the indoor area, the global pose of the mobile device can be computed from the camera pose estimation. The inertial tracker provides orientation dead reckoning if no markers are observed.

Tracker selection: The OT configuration in the client is responsible for making the tracking "ubiquitous", i.e., permitting online selection of the best available tracking technology. The selection mechanism is currently based on priorities: The stationary tracking technologies ARTTrack, Flock of Birds, and ARToolKit are selected in this order if data from the corresponding sensors is available. The systems attempts to fall back to GPS if none of these technologies is available, and finally reverts to a static map when there is no location data whatsoever. An improved version of the tracking framework is currently under development, which will permit fully automatic discovery of new tracking services based on a tracking service characterization.

2.4. Applications

Outdoor applications. The needs and requirements of a tourist are a suitable starting point for testing location-based applications. A tourist typically has 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. Guided tours are also popular for tourists. Consequently, we have chosen a tourist guide application for the City of Vienna as an example scenario for an AR application that integrates a large amount of data from different sources.

Fig. 5: An outdoor navigation guide for a pedestrian visualizes the selected route as a series of waypoints. Note the correct occlusion between the route and the archway .

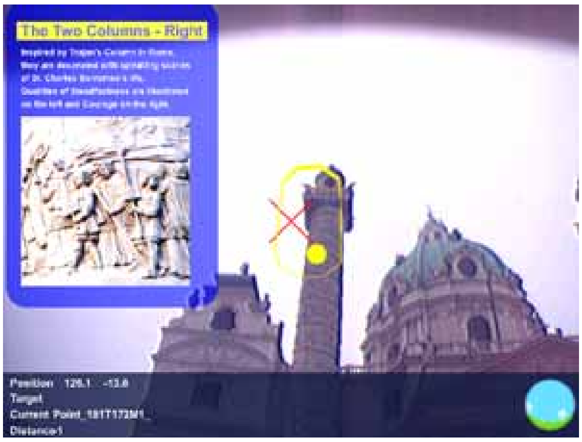

Fig. 6: By gazing at a cultural artefact of interest, a user can instantly recall historic multimedia information.

Fig. 7: Indoors, a combination of directional arrows, highlighting of exit, compass (upper right) and world-in-miniature (lower middle) is used to help a user navigate the environment.

Fig. 8: The "Augmented Library" allows a user to quickly locate the shelf on which a book of interest can be found, by emphasizing the relevant location.

The system provides a navigational aid that directs the user to a target location. An information browser displays location-referenced information icons that can be selected to present more detailed information in a variety of formats. Both functions support collaboration between multiple mobile users.

In 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. It is interactive and reacts to the user's movements. It continuously re-computes 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 (Fig. 5). These cylinders are connected by arrows indicating the direction the user should take between waypoints. Together they become a visible line through the environment that is easy to follow. 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. Finally, the system displays simple directional information, if the user is not able to perceive the next waypoint because she is looking in the wrong direction.

The information browsing mode presents the user with location-based information. Location-referenced information icons appear in view and are selected by looking at them (Fig. 6). They then present additional information associated with that location. The application conveys historical and cultural information about sights in the City of Vienna.

Indoor applications. Indoors, the mobile AR system guides a user on the way through the long and winding corridors of a university building with direction and map based contextual information [RS03]. The system continuously provides the user with two modes of visual feedback: a heads-up display with directional arrows and a world in miniature model of the environment. The heads-up display shows a wire frame model of the current room superimposed on top of the real scene. The application uses a shortest path search on an adjacency graph of the building to determine the next door/portal on the way to the destination, which is then visually highlighted. In addition, an arrow shows the direction to the next door or portal turn (Fig. 7). The application also presents a world in miniature model of the building to the user in the lower area of the heads-up display. While the 3D overlay only shows the next exit, the miniature model shows a overview of the user's current environment including the complete path.

In the library, the mobile AR system may assist a user in locating and retrieving a book. Using a menu system, the user can select a book from a database. The corresponding bookshelf is highlighted in the heads-up display to aid the user in finding the book (Fig. 8). The system can also help in the return of a book. If a marked book is identified, the book's designated shelf is once again highlighted aiding the user to return the book to its correct position. This enables the user to simply look at the book in her hand and trigger the appropriate application behavior.

3. First experiences with large scale data models for AR

The indoor and outdoor applications presented in the last section require extensive 3D models and information presented to the user. Accurate and complete models of the buildings interiors and overall shape are required for rendering occlusions and highlights of buildings. Different navigation models for indoor or outdoor use were developed to fit the specific requirements.

Both applications are supported by a common world model based on an XML description of the geometry of world features. The XML tree is interpreted in the standard geometrical way, by defining a child's pose relative to its parent. 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 and annotations.

An outdoor AR system can be considered as a special case of a geographic information system (GIS). It presents geo-referenced information in real-time and in 3D, based on the physical location of the user, user preferences, and other context-dependent information. Large amounts of geo-referenced information, such as a 3D world model, require a database system for efficient storage and retrieval. The introduction of a GIS database also solves the problem of providing a consistent view of the 3D world model for a potentially large number of wirelessly connected clients.

Depending on location and context, only a small subset of the information contained in the GIS database is necessary for the client. Information is therefore retrieved dynamically by querying a database server. The response to such a GIS database query typically undergoes a series of transformation steps. Common operations are filtering according to geographic and logical constraints (e.g., return all coffee shops in a radius of 100m), and translation from a more generic format to a data structure that can be directly visualized [VZ04].

When we started with the implementation of our mobile AR framework, solutions for 3D GIS visualization were not sufficiently advanced for our purposes. Consequently, we developed our own XML based data format and processing pipeline using XSLT for data translation. Recently, the emergent Geographic Markup Language version 3 (GML3) standard together with the Web Feature Service (WFS) standard provide a standardized and extensible interface for accessing GIS information. CityGML [KG03], an application profile (extension) to GML3, is similar in spirit to our own custom XML dialect. Since it can be expected that these new standards will be widely supported in the near future, it is advisable to adopt them for AR applications as well.

4. Automating visualization generation

The complexity of interactive visualizations demands automated methods for generating engaging presentations. The fundamental idea of automated visualization generation for AR is quite old [FMS93], but few or no tools for this purpose exist, and GIS technology does not directly resolve this problem either.

Consider, for example, recent research which focuses on specialized visualization techniques for augmented reality, such as communicating the distance of occluded objects [FAD02], improving the readability of text overlays [LT04], providing automated layout of presentation items [HFH*01], filtering information [JLB*00], or adapting the visualization in the presence of tracking errors [CMJ04]. All these techniques could be made generally available to AR applications by an automated approach for visualization generation. However, a simple XML-based world model as presented in the last section is not flexible enough for this purpose.

GML's approach of an explicit representation of the geometric and other relations through so called features is better suited to provide the flexibility necessary for combining a large variety of applications and data sources. However, GML defines the syntactical aspects of geographic data exchange by fixing low-level data types and describing what information features contain and what relations exist. Any application using a specific GML data source therefore needs to know in advance how to interpret these features and relations. This means that these relations, feature names, and resultant structure are hard-wired into the application itself. Moreover, only traditional relational queries are supported. More complex queries, incorporating transitive closures of relations, require query iterations to implement. However, such requests are commonplace in graph search algorithms such as finding a path through a navigation network.

In order to further decouple applications from this inherent structure and to support a more expressive query language, we are investigating the use of another layer of information encapsulated in a semantic network on which applications operate. The semantic network layer includes the application's view of the world and the data sources' view and is able to integrate both. As a result, visualizations can be described independently of the structure of the underlying data source.

In the remainder of this section, it will be demonstrated with a set of examples how an integrated and semantic world model enables these methods. The semantic aspect of the world model entails that such visualizations can be developed independently from the underlying data model.

4.1. Example: Gas utility company

Professionals dealing with hidden and embedded structures can be supported in their field work with the integration of administrative information with detailed models of the physical environment. For example, a worker for a gas utility company has to find and repair a leak in a gas pipe in the field.

The worker will require accurate 3D information to locate access points to the pipe where he can measure various operational parameters of the pipe in order to locate the leak. The parameters are referenced to the points and can be queried by the system on the fly from the combined spatial-semantic database. A handheld visualization device will use the stored 3D model at the same time to display the layout and placement of the hidden pipe in the ground to facilitate planning of the repair. The worker can then proceed to carry out the maintenance.

However, the displayed information is not limited exclusively to the task at hand. Any other subsurface structure in close proximity to the site is also extracted from the model and its relevance is assessed. Obstructing structures are identified by analyzing the geometric relations they have with the pipe. For example, if another pipe is above the gas pipe of interest, the visualization will include it and show any possible areas of intersection with the work path. Similarly, other structures are checked for dangers to the planned work, such as electric fields that could trigger sparks in tools. Again the visualization will highlight such structures to inform the worker of possible risks.

After the task is completed the relevant administrative information including work time, material, client information and nature of the defect are entered by the worker and automatically related to the pipe and location by the system again. The collected information can then enter the business logic workflow of the company without further overhead. The relation to the 3D model enables automatic cross-referencing of administrative data and real-world locations and artefacts. Finally such a model can be reused for information visualization in report and analysis work.

4.2. Example: Pedestrian navigation

Navigation systems for cars have become standard equipment due to accurate and affordable tracking and high-quality road maps. Similarly, one can expect that pedestrian navigation covering outdoor city use but also indoor areas will become a generic feature of future mobile systems. Various systems have demonstrated partial results in this area. However universal navigation from room to room across two buildings, streets and even city districts remains out of reach. Besides the unavailability of ubiquitous tracking, a generic model covering all levels of details involved in such a task also does not exist.

Consider a user having an appointment with a customer in an office building across the city from her current location. Her PDA queries a web service ahead of the meeting time to compute a route from her own office to the customer's. Such a route will not only include accurate driving instructions but also information on possible parking spaces in the vicinity of the destination area, the path she has to walk to get to the office building and information on how to get to the customer's office.

A display included into her sunglasses conveys navigational information as she leaves her car and walks towards the office building. The entrance she has to use is highlighted and to give some impression of the location of the customer's office, the system highlights the windows facing out of the room on the building façade. Within the building, the system directs her to an elevator and displays, or automatically selects, the floor level she has to go to. Once on the right floor, the system points her in the right direction along the corridor.

4.3. Adaptive visualization engine

Within the scenarios described above, we identified several tasks that should be delegated to an automated adaptive visualization engine. Such an engine will operate on the given world model and its meta-data to derive which elements of the model are of interest and what the appropriate style of presentation should be.

Deriving styles and transformations. The adaptive visualization engine queries the world model for the structures directly relevant to the worker's task. From the given work area it also derives other structures that intersect the planned excavation volume. Based on the attributes associated with the structures it finds, it can assign appropriate visualization styles to them.

For example, structures that lie above the designated pipe are rendered in a bright color to draw attention to them. Nevertheless, they are rendered translucent enhancing the perception of the main structure. Structures within the volume but below the pipe are rendered with darker colors as they are not as important.

Similarly, a projection of the main structure and the excavation volume to the ground model is computed to throw a "shadow" on the real ground. Such a "shadow" maps the 3D location directly to the visible surface area and delineates the required excavation area.

Displaying dangerous areas. The worker's system should automatically identify dangerous areas or structures in the working area and notify the worker with appropriate signals. To do so it needs a model of the possible hazards related to the task at hand and a method of querying the world model to retrieve structures that fit the model.

To implement such functionality the application has a list of attributes that relate hazardous structures to the selected task. Then a query is formulated to search for structures intersecting the working area and being annotated with attributes matching the possible hazards. The query is translated into one or more WFS queries and is send to the server.

The results are converted into a scene graph suitable for rendering and interaction. Based on the returned attribute values, different levels of severity are assigned to the structures and presentation styles are set accordingly. The resulting scene graph is then added to the viewer's graph and becomes part of the presentation.

Deriving related objects of interest. For a mobile task like navigation, objects of interest are often occluded or not fully or directly visible. In such cases it is not simple for the user to interpret the visualization correctly, even if various techniques such as transparency or cut-away views are employed. Therefore the adaptive visualization engine can substitute currently visible structures for the actual target structures and use these in the presentation.

For example, as the user is coming close to the target building both the building entrance and a window associated with the target room are highlighted. The system emphasises the entrance in order to guide the user in the right direction, but also, it draws some attention to the window to provide some overview of the planned path. The user can therefore build a mental map of where she will go.

Within the building, doorways are again used as direct navigation aids, but an outline of the intersection of the room with the adjacent walls can provide more information about the location and size of the target room. Such subtle additions to the presentation could help users to gain a better understanding of the structure of their environment.

To implement these functions, the system will first compute a path from the current location of the user to the destination. Moreover, it queries the world model for potential occluding objects in the area between the user and the destination object. If such objects exist, it will further derive visible sub-structures of these objects and try to relate them to the destination room. Such relations can be topological such as windows connecting a room volume to the outside or geometrical such as proximity of walls. If such a related object is found, it can be used in the visualization instead of the destination object.

5. Semantic reasoning engine

More complex automation is achieved by incorporating knowledge-based query mechanisms into the application. Our aim is to provide a software component that reasons over the available world model to identify and select objects described by more complex propositions than simple query-based assertions.

The semantic reasoning engine maintains a representation of the world and application state as a semantic network which contains information on the individual objects in the scene, the tasks of the user and the selected visualization modes. The semantic network is based on an ontological description of the properties of objects.

It combines a set of formal descriptions of application knowledge:

- A basic ontology like OWL [HPH03] for defining the basic relationships between objects, such as membership of a class, subclasses, aggregation and attributes.
- An ontological description of the GML-based data in order to interpret and map the GML features to the semantic network representation.
- Application specific ontologies that further refine the basic ontology with the relations that hold in the application domain.
- A description of the possible user interface representations and their relations to domain specific object attributes.
- A set of mapping ontologies that translate between the GML relations and the application-specific or user interface concepts.

Finally, a knowledge base of facts about the objects is also maintained in the network. These facts are representations of the underlying GML feature data retrieved from the Web Feature Service back-end.

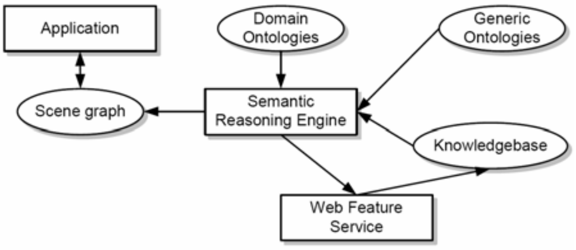

Fig. 9: The semantic reasoning engine relies on a knowledge base interpreted according to various ontologies.

An inference engine such as Racer [HM03] or FaCT [Hor98] processes queries defined by the application and returns the results with respect to the semantic network. Relations that include geometric semantics such as distance or visibility are implemented as extensions to the inference engine that operate on the geometric data itself, rather than on the representations in the semantic network. If further facts are required, the engine also formulates queries to the WFS back-end to replenish the knowledge base with more information. For example, a prerequisite to certain queries could be that all objects within a certain distance of the user are kept in the knowledge base. As the user moves, the engine will update the knowledge base as required.

5.1. Building visualizations

The next step uses the knowledge encoded in the semantic network to derive convincing visualizations. Here we use the ontology describing the properties of types of visualization. For example, we could define that objects that are occluded but important for user orientation are of type *I* and are to be rendered with a bright yellow stippled outline, while objects that are occluded by any object of type *I* are themselves of type *O* and are rendered with a black translucent color to modify the luminance of occluded objects.

The inference engine can now deduce from assertions in the knowledge base whether a certain object meets the criteria to be classified as either type *I* or *O*. Similarly, we could also formulate a query that returns all objects of a certain type. Note that the decision of whether an object is of type *I* is not solely based on geometric calculations, because it must also be important to user orientation. A formalization of this concept is in fact encoded in the domain specific application ontology.

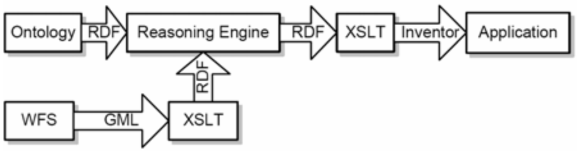

Fig. 10: XSLT is used as the general translation mechanism that interfaces the individual parts of the semantic reasoning system for AR. It can convert between geometric information (GML), semantic information (RDF) and graphical data for visualization, such as the scene graph standard Open Inventor.

The connection between the semantic reasoning engine and the user interface component is another crucial aspect of the system. The output of the engine – the results of queries to it – is transformed into scene graph representations for the rendering engine. However, due to the possible complex computations of query results, we cannot expect the reasoning engine to operate at real time in direct response to the user's interactions.

The output scene graph will rather describe the possible visualizations of all objects with respect to certain changes such as visibility or distance to the user or actions by the user such as selection. The visualization will be rendered with appropriate parameters driven by the sensor input and adapt at render time to such simple changes. Only larger movements exceeding certain thresholds or interactions with the application such as selecting a different mode or target will trigger a re-evaluation by the semantic reasoning engine.

5.2. System integration

The integration of the semantic reasoning engine into the overall system is supported by the extensive use of XML-based data exchange formats. Ontologies are expressed in RDF [MM03] and can therefore be served by web services. The knowledge base of facts in the reasoning engine is typically also expressed in RDF. Therefore XML-based transformation technologies such as XSLT can be applied to the GML query results of the Web Feature Service to extract and formulate the assertions in RDF and feed them to the reasoning engine.

The output to the application is mainly entailing the creation of scene graphs for 3D rendering. Here we have already successfully employed the same techniques to generate scene graph descriptions from XML data structures. As the output of the reasoning engine consists of facts expressed in RDF too, we can directly apply static transformations to it and create the corresponding graph structures.

5.3. Example: Tracking target objects for navigation

To demonstrate the use of the semantic reasoning engine we will describe its operation when applied to universal navigation. Here we always want to present the user with an object that is "semantically" related to the target object, even if the target is not directly visible.

Within our application specific rule base, we define a set of rules that always compute a visible place-holder for the target object. A first rule is that the best place-holder is the target object itself, if visible. Another rule states that an object belongs to the type class *IsInteresing*, if it is visible and stands in the relation *SubstituteFor* with the target object. The relation *SubstituteFor* is defined in terms of a set of logical disjunctions and conjunctions of assertions on properties of both the target and the candidate objects. Among the assertions is the topological relation *neighboursWith* meaning a spatial relation that objects are adjacent to each other. The concrete implementation in terms of testable attributes or relations depends on the world model. Therefore it is either defined in the world model itself or as a set of geometric relations *Touches* which is part of the query language for the Web Feature Service.

For example, in order to find windows that may act as substitutes for a target room, we define the relation *SubstituteFor* to include objects of a size not larger than the target, that stand in the relation *neighboursWith* to it and that are potentially visible, because the user is currently outside of the building. The semantic reasoning engine is able to map the relations to real underlying relations in the current GML data set without further interaction with the application. Therefore, the visualizations depend on high-level abstract descriptions of interesting objects rather than on the direct low-level expressions used to compute them.

The result of the reasoning step is the set of all objects belonging to the class *IsInteresting*. The result set creates a scene graph representation that includes the objects' geometry and appropriate rendering styles. Moreover, the scene graph also includes some logic to react to individual objects becoming visible or invisible as the user moves within a certain range, such that only one object is actually rendered at any point in time. When the user moves beyond this range or selects a different target, the application re-evaluates the set by updating the knowledge base accordingly and applying the reasoning again.

6. Conclusions and future work

Mobile AR has outgrown its infancy and is getting ready for early commercial deployment. We have presented a series of platforms and application prototypes developed to assess the feasibility of key technologies in mobile AR. One noteworthy aspect of our experiments is that they use probably the largest and most systematic AR model to date. From our experiences with this model and its creation process we have learned what today's 3D modeling technologies do not provide: truly flexible interpretation of the data, which makes applications independent of assumptions concerning model structure and the relations between model entities. Semantic web technology aims to overcome this problem in the domain of online information systems.

In this paper we explained why mobile AR, with its potentially large number of clients and location-based service providers, has essentially the same requirements as document-oriented semantic web applications, but in the domain of real-time 3D information. We derive a data model which allows a suitable degree of semantic reasoning for mobile AR, and describe how it can be used in practical examples. While we already have many tools in place for the implementation of such a model through our own work and the resources available in the GIS and semantic web communities, the verification of the approach with real world scenarios will be the subject of future work.

Acknowledgements. This work was sponsored by the Austrian Science Fund *FWF* (contract Y193). Thanks to Joseph Newman and Erick Mendez for proofreading and for useful suggestions.

References

[Azu97] Azuma R. T.: A survey of augmented reality. *Presence Teleoperators and Virtual Environments* 6, 4 (August 1997), pp. 355–385.

[CMJ04] Coelho E. M., MacIntyre B., Julier S. J.: Osgar: A scene graph with uncertain transformations. In *Proc. ISMAR 2004* (Arlington, VA, USA, November 2–5 2004), IEEE.

[FAD02] Furmanski C., Azuma R., Daily M.: Augmented-reality visualizations guided by cognition: Perceptual heuristics for combining visible and obscured information. In *Proc. ISMAR 2002* (Darmstadt, Germany, September 30 – October 1 2002), IEEE and ACM, pp. 215–224.

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

[FMS93] Feiner S., MacIntyre B., Seligmann D.: Knowledge-based augmented reality. *Communications of the ACM* 36, 7 (July 1993), pp. 53–62.

[Hor98] Horrocks I. Using an expressive description logic: FaCT or fiction? In *Proc. Principles of Knowledge Representation and Reasoning KR'98* (San Francisco, CA, USA, June 1998), pp. 636-647.

[HFH*01] Höllerer T., Feiner S., Hallaway D., Bell B., Lanzagorta M., Brown D., Julier S., Baillot Y., Rosenblum L.: User interface management techniques for collaborative mobile augmented reality. *Computer & Graphics* 25, 25 (2001), pp. 799–810.

[HHF*04] Hallaway D., Höllerer T., Feiner S., Bridging the gaps: Hybrid tracking for adaptive mobile augmented reality. In: Applied Artificial Intelligence, Special Edition on Artificial Intelligence in Mobile Systems, 25, 5 (July 2004).

[HM03] Haarslev V., Möller R., Racer: A core inference engine for the semantic web. In York Sure and Óscar Corcho, editors, *Proc. EON2003*, (Sanibel Island, FL, USA, October 20 2003).

[HPH03] Horrocks I., Patel-Schneider P.F., van Harmelen F. From SHIQ and RDF to OWL: The making of a web ontology language. *Journal of Web Semantics* 1, 1 (2003), pp. 7-26.

[JLB*00] Julier S., Lanzagorta M., Baillot Y., Rosenblum L.,Feiner S., Höllerer T.: Information filtering for mobile augmented reality. In *Proc. ISAR 2000* (Munich, Germany, October 5–6 2000), IEEE and ACM, pp. 3–11.

[KG03] Kolbe T. H., Gröger G.: Towards unified 3d city models. In *Proc. of the ISPRS Comm. IV Joint Workshop on Challenges in Geospatial Analysis*, Integration and Visualization II (Stuttgart, Germany, 2003).

[LT04] Leykin A., Tuceryan M.: Automatic determination of text readability over textured backgrounds for augmented reality systems. In *Proc. ISMAR 2004* (Arlington, VA, USA, November 2–5 2004), IEEE.

[MM03] Manola F., Miller E. Rdf Primer. *W3C*, http://www.w3c.org/TR/rdf-primer/, visited May 3 2003.

[NWB*04] Newman J., Wagner M., Bauer M., MacWilliams A., Pintaric T., Beyer D., Pustka D., Strasser F., Schmalstieg D., Klinker G.: Ubiquitous tracking for augmented reality. In *Proc. ISMAR 2004* (Arlington, VA, USA, November 2–5 2004), IEEE.

[PAT*04] Piekarski W., Avery B., Thomas B. H., and Malbezin P., *Integrated Head and Hand Tracking for Indoor and Outdoor Augmented Reality* in Proc. IEEE Virtual Reality Conference, (Chicago, IL, USA, March 2004), pp. 11-276.

[RCK*05] Reitmayr G., Chiu C., Kusternig A., Kusternig M., Witzmann H.: iOrb - unifying command and 3d input for mobile augmented reality. In *Proc. IEEE VR Workshop on New Directions in 3D User Interfaces* (Bonn, Germany, March 12 2005), IEEE, pp. 7–10.

[RS01] Reitmayr G., Schmalstieg D.: OpenTracker – an open software architecture for reconfigurable tracking based on XML. In *Proc. IEEE Virtual Reality 2001* (Yokohama, Japan, March 13–17 2001), pp. 285–286.

[RS03] Reitmayr G., Schmalstieg D.: Location based applications for mobile augmented reality. In *Proc. AUIC 2003* (Adelaide, Australia, February 4 – 7 2003), Biddle R., Thomas B., (Eds.), vol. 25 (3) of Australian Computer Science Communications, ACS, pp. 65–73.

[RS04] Reitmayr G., Schmalstieg D.: Collaborative augmented reality for outdoor navigation and information browsing. In *Proc. Symposium Location Based Services and TeleCartography – Geowissenschaftliche Mitteilungen* (Vienna, Austria, January 28–29 2004), vol. 66, Wiley, pp. 53–62.

[SFH*02] Schmalstieg D., Fuhrmann A., Hesina G., Szalavari Z., Encarnacao L. M., Gervautz M., Purgathofer W.: The Studierstube augmented reality project. *PRESENCE – Teleoperators and Virtual Environments* 11, 1 (2002), pp. 32-54.

[TDP*98] Bruce H. Thomas, V.Demczuk, Wayne Piekarski, David Hepworth, and Bernard Gunther. A wearable computer system with augmented reality to support terrestrial navigation. In Proc. *ISWC'98*, (Pittsburgh, USA, October 19–20 1998), IEEE and ACM, pp. 168–171.

[VZ03] de Vries M., Zlatanova S., Interoperability on the Web: The Case of 3D Geo-Data. In *Proc. IADIS International Conference e-Society 2004* (Avila, Spain, July 16-19), pp. 667-674.

[WS03] Wagner D., Schmalstieg D.: First Steps Towards Handheld Augmented Reality. Proc. *7th International Conference on Wearable Computers*, (White Plains, NY, USA, October 21-23, 2003), pp. 127-135.