ARTHUR: A Collaborative Augmented Environment for Architectural Design and Urban Planning

W. Broll[1], I. Lindt[1], J. Ohlenburg[1], M. Wittkämper[1], C. Yuan[1], T. Novotny[1]
C. Mottram[2], A. Fatah gen. Schieck[2], A. Strothmann[3]
[1]Fraunhofer FIT, Germany,
[2]The Bartlett, University College London, UK,
[3]Linie 4 Architekten, Germany
wolfgang.broll@fit.fraunhofer.de

Abstract

Projects in the area of architectural design and urban planning typically engage several architects as well as experts from other professions. While the design and review meetings thus often involve a large number of cooperating participants, the actual design is still done by the individuals in the time between those meetings using desktop PCs and CAD applications. A real collaborative approach to architectural design and urban planning is often limited to early paper-based sketches.

In order to overcome these limitations we designed and realized the Augmented Round Table, a new approach to support complex design and planning decisions for architects. While AR has been applied to this area earlier, our approach does not try to replace the use of CAD systems but rather integrates them seamlessly into the collaborative AR environment. The approach is enhanced by intuitive interaction mechanisms that can be easily configured for different application scenarios.

1. Introduction

Architectural design and urban planning – at least for sophisticated projects - have always involved highly cooperative tasks. The individual phases within a project often change between close cooperative situations for instance during design and review meetings, and individual work by the participants or third parties. While in between these meetings, people involved of course have a common goal they contribute to, they will rather work on individual parts or aspects of it. During the design and review meetings problems are discussed and solutions or alternatives are proposed. However, the actual preparation of the individual solutions is then again performed by the individuals (leaving the final decision to one of the following meetings). Real collaboration – if happening at all – is limited to the creation of very early design sketches. From an architects point of view it would be desirable to have an additional support tool allowing to improve the cooperation in a way that real collaboration within the meetings is supported. This would then allow for much faster design and review cycles.

In this paper we will present our approach to an AR system supporting the individual phases of architectural design or urban planning. The goal of our approach was to provide the necessary environment and tools to support full collaboration between all experts involved in the particular meetings, without altering the established working procedures and accepted tools and mechanisms too much. We tried to enhance the meeting situations by fully integrating them into the work of the individual contributors. This approach is reflected by the use of intuitive interaction mechanisms which allow even untrained users to quickly benefit from the enhancements provided by the AR environment. Additionally, existing tools such as CAD systems and simulation programs already used by the people involved were integrated to provide a rather seamless transition between their individual daily work and the collaborative work at the round table meetings.

Our paper is structured as follows: in section 2 we will discuss related work, before we will provide a brief overview of the ARTHUR AR system in section 3. In section 4 we will describe the major application areas and how the system has been tailored to support them. In section 5 we will present the feedback we received from user tests, followed by the conclusion and look into future work in section 6.

2. Related Work

Over the past few years there have been a number of attempts to develop AR systems in the area of architectural design and urban planning. For construction sites, it has been proposed that an AR system might provide users with an "X-ray vision" inside a building, allowing them to see, for instance, where the pipes, electric ducting, and structural supports are situated inside walls and above ceilings [9] [31]. Such systems visualize hidden features of a building and are well suited to support maintenance and repair. AR is also useful to get a realistic impression of existing plans. Thomas et al. [29] developed, for example, a system for outdoor visualization of construction plans. Designs are exported from a CAD application and displayed in their physical outdoor context using the *TINMITH2 system*. Other AR systems in the area of architectural design and urban planning support the assembly of complex systems. *Augmented Reality for Construction* [10] supports e.g. the construction of spaceframes by indicating the position of each structural element in space. Another example is the *Assembly Instructor* [32] that allows for an easy authoring and context-based visualization of hierarchical assembly tasks.

An approach that goes beyond the visualization of spatial designs is the support of collaborative design and planning tasks. Common meetings are enhanced with AR technology to allow for a joint view and collaborative manipulation of complex spatial problems. The following subsections summarize AR systems for the collaborative design of products and production lines as well as for architecture and urban planning.

An early prototype for collaborative planning is *BUILD-IT* [24]. BUILD-IT supports engineers in designing assembly lines and building plants. The technical infrastructure is based upon a table top interaction area, enhanced by a projection of a 2D computer scene on the table top. Additionally, a video camera is used to track small, specialized bricks, that can be used as “universal interaction handlers”. A second, vertical projection screen provides a 3D view of the virtual scene.

MagicMeeting [25] supports product review meetings by augmenting a real meeting location. Instead of real mock-ups virtual 3D models are used that may be loaded into the environment from usual desktop applications or from Personal Digital Assistants (PDAs). The MagicMeeting system explores several interaction techniques, such as the MagicBook metaphor [5], annotations and a clipping plane tool. The 3D models are linked to physical placeholder objects and realize a tangible interface.

MARE [14] is a multi-user augmented reality system. The table space is divided into two parts: a personal area for the private real objects and virtual ones (private menus); and a shared interactive space.

MIXDesign [7] provides a Mixed-Reality system oriented towards tasks in Architectural Design. It explores tangible interfaces using AR Toolkit patterns on a paddle and gestures.

Tiles [23] implements a generic user interface based on a loose coupling of visual markers (here: tiles) to synthetic content. The system distinguishes between two classes of tiles: data tiles and operation tiles. Data tiles may contain arbitrary digital content, whereas operation tiles define certain operations that may be applied to data tiles. The Tiles system has been used for rapid prototyping and collaborative evaluation of aircraft instrument panels and is well-suited to compose complex spatial designs from a variety of smaller components.

AR-Planning Tool [13] supports the collaborative planning of production lines. Machines are modelled as virtual building blocks and can be positioned by the user with a visually tracked paddle. The system checks the validity of the planed production line using a database with metadata for each machine. The user wears a video-augmented HMD to see the virtual machines.

The *Luminous Table* [18] developed by the Tangible Media Group integrates sketches, physical models, and computational simulations into a single workspace. 2D drawings and 3D physical models are augmented with a 2D video projection to simulate sunlight shadows, wind patterns and traffic. The physical objects are tracked with cameras.

ARVIKA [12] realizes AR applications for development, production and servicing. An application developed within ARVIKA allows for collaborative plant design. Paper-based floor plans are augmented with virtual objects to get a better impression of the current planning. Placeholder objects are used to position virtual objects; menus can be used to insert or delete objects.

AR systems for collaborative design and planning typically support the spatial composition of larger designs from existing building blocks (compare BUILD-IT and Tiles). They integrate planning rules (compare AR Planning Tool) and sophisticated interaction metaphors (compare MagicMeeting and Tiles) and they have mature concepts for integrating physical and digital workspaces (compare Luminous Table and ARVIKA).

But still, they are restricted regarding intuitive interaction mechanisms and functionality. ARTHUR tries to preserve the natural communication and collaboration between meeting participants. We are using optical augmentation and wireless computer-vision based trackers to allow for a natural 3D collaboration. Virtual objects are displayed using stereoscopic visualization to seamlessly integrate them into the physical environment.

Another point is the limited functionality of most systems. Typically, AR planning system are limited to basic manipulation of imported geometry. ARTHUR integrates a CAD system to support more advanced functionality such as 3D sketching and extrusion. Geometry is not just imported from modelling software, but can be directly created within ARTHUR.

3. The ARTHUR System

To realize our approach to support complex architectural design and planning decisions, several major parts have been developed and extended, such as the Morgan framework, the AR displays, the computer-vision based object and head tracking and the Graphical User Interface configuration. The individual system components are discussed in more detail in the following sub-sections. For further information on the ARTHUR system see also [20] and [14].

3.1. The MORGAN Framework

Our AR framework MORGAN consists of three major parts: the 3D visualization component, the distribution and communication framework and the developers' application interface.

Figure 1. ARTHUR See-through head mounted display (HMD).

3.1.1. 3D-Visualization. The 3D visualization component allows the user to see the ARTHUR scenarios in 3D using head mounted displays or other output devices. Rendering can be either stereoscopic (quad-buffered or dual screen) or monoscopic. Augmentation within ARTHUR is usually done using see-through augmentation, i.e. the virtual image is optically superimposing the real environment. The 3D visualization component, however, additionally supports video augmentation, where the image from a head-mounted camera is superimposed by the virtual scene components. This is for example used for screen-based presentations or larger projections to show people not wearing an HMD, what is currently visible for those participants.

The 3D visualization component is based on a component based scene graph architecture. While an optimized internal scene graph is used to perform the actual rendering, external scene graphs are attached to support and conserve individual native scene graph structures. This architecture allows us to use VRML'97 / X3D for the description of user interface elements and ARTHUR specific objects, while e.g. CAD objects may use their own individual external scene graph (see also section 4.3). The ISO standard VRML'97 has been enhanced by features required for AR (such as phantom objects) and interaction (such as high-lighting, mouse pointer independent picking and collision detection). In the overall system, one visualization component exists for each individual user, thus rendering is performed locally. In order to achieve this, the scene graphs are replicated among the individual visualization components and kept synchronized upon changes.

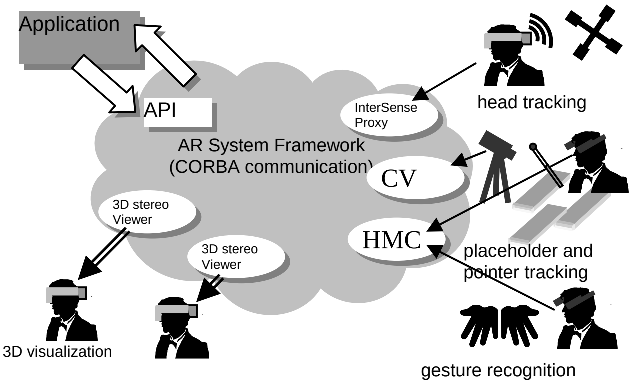

Figure 2. The AR framework MORGAN

3.1.2. Distribution and Communication. Our AR framework MORGAN provides the distribution and communication mechanisms required to connect the input devices such as head tracking (i.e. InterSense IS900 or InertiaCube^2), computer vision input (placeholder tracking, pointer and finger tracking, gesture recognition, alternative head tracking) to other system components (e.g. the 3D stereo visualization components) (see Figure 2). This allows the visualization components to adapt their current viewing position and orientation to the tracking input. Additionally, this distribution mechanism is also used to keep the scene graphs of multiple users synchronized (i.e. upon changes to one local scene graph, these changes are immediately distributed to all other replicated scene graphs). Synchronized clocks between all PCs involved are required in order to resolve ambiguities. The general communication mechanism used within the AR framework is CORBA. A universal sequencer mechanism provides an overall virtual time management for all components based on virtual time.

3.1.3. **The Application Programming Interface.** The MORGAN API is a C++ based interface to the AR framework. It provides application programmers with an interface to the input devices connected to the system (tracking, object and gesture recognition) as well as to standard devices such as mouse and keyboard. Information on current scene graph objects can be queried; objects may be created, modified, replaced or deleted. Additionally, the application programmer can access more advanced scene graph operations required to realize user interface operations such as picking of objects along a ray or collision detection between objects.

3.2. AR Displays

As part of the ARTHUR project a new head mounted display (HMD) has been developed (see Figure 1) to fit the particular needs of the environment and the users. The developed HMD is a transparent high-resolution binocular (stereoscopic) display. The image is generated on two independent full color 1280x1024 pixels (SXGA) liquid crystal on silicon micro displays. The display features excellent image quality, high brightness and contrast.

The image can be shown superimposed on the environment with up to 35% see-through or fully immersed. The ARTHUR HMD is designed for a 46 degree diagonal 100% stereo overlap field of view. A 50% overlap can also be used, giving a 54-degree horizontal or 60 degree diagonal field of view. It features a patented optical design that combines a wide field of view with high transparency see-through and a patent pending head fitting system with easy adjustments, low weight and eyeglass compatibility for most users.

3.3. Input Mechanisms

In our approach computer vision (CV) is used for user input and it may also be used for head tracking. This section describes first the CV based input mechanisms, and secondly comments briefly on the implementation of the CV system.

Two main types of input mechanisms are available for user interaction; 1) devices such as tangible interfaces and wands, and 2) hand gestures and fingertip tracking.

Input Devices

The user interface devices are dedicated objects that are tracked by the CV system using colour and shape information. There are two types of devices: placeholder objects and wand-like pointers.

Placeholder objects (PHO) are tracked in the table plane, in two translational and one rotational degrees-of-freedom (3DOF). They are of a convenient size to be grabbed and moved by the users (see Figure 3). More than ten PHO may be used concurrently. The users may take a PHO, associate it with any virtual object and move this virtual object by moving the PHO, which is described in more detail in the next section.

The *pointers* are tracked in 5 DOF – all except roll. The pointers have three buttons for functionalities such as pick or select. Users may select and manipulate the shape of virtual objects with a pointer or use it to navigate in virtual menus (see).

Gestures

Two types of gestures can be used, static command gestures and 3D fingertip tracking.

Currently a set of five *static command gestures* may be used. The number of fingers shown to a camera corresponds to a gesture. The command gestures may be used to get a pop-up menu or for functions such as copy-cut-paste.

3D fingertip tracking may be used to draw a line in space or to navigate in pop-up menus, select items and execute actions.

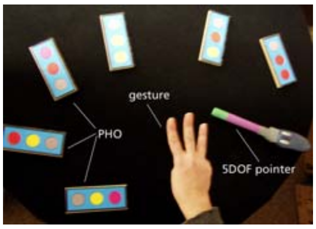

Figure 3. Input image for CV-based recognition of PHOs, pointers, and finger gestures

3.4. Graphical User Interface Configuration

A graphical language (GRAIL) has been developed to allow users to configure the relationships between the 3DOF PHO, the 5DOF pointers, the command gestures and the virtual objects. GRAIL also provides the user with the ability to link the ARTHUR system to external applications, such as pedestrian and environmental analysis applications and project planning and cost estimating applications, using scripting commands. The GRAIL application sits on top of the MORGAN API (see section 3.1.3) and acts as a ‘tool building tool’ to develop AR user interfaces that define the properties and characteristics of the overall interaction environment. It allows users to create a range of tools by graphically defining the relationship between the 3DOF PHOs or 5DOF pointers and the command gestures or virtual objects. A set of five command gestures is implemented For instance, a two-finger gesture could be used as a “tool” for creating virtual boxes in the AR environment. This can be achieved by linking the command “create box" with the icon, which represents the two-finger gesture in the GRAIL window graphically. Similarly, a virtual object "a box" could be converted into a "tool" that creates virtual boxes in the AR environment by linking the command "create box" with the VRML object "Box" from the list of VRML objects in the GRAIL window. The relationships within a GRAIL configuration can easily be changed and reassigned providing the ARTHUR system with a flexible interface that supports a rich set of interactions with the virtual scene.

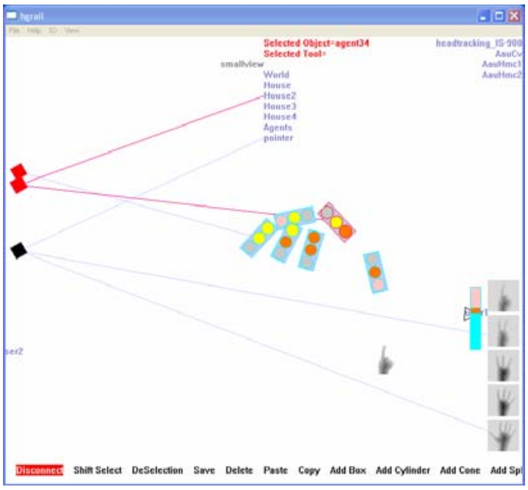

Figure 4. GRAIL application

4. Application Scenarios

In this section we will describe three application scenarios that we have developed and tested so far. While the ARTHUR system in general is not limited to these scenarios, those were the ones tested by architects and other users.

4.1. Natural 3D Collaboration for Design and Planning

The task of creating a system such as ARTHUR demands a high degree of communication and collaboration with potential users, in this case trained architects.

In order to ensure the success of this undertaking the technique of scenario-based design [26] was adopted for system development.

The scenarios were designed collaboratively by the application partners as well as designers in order to provide for feedback on a user as well as on a technical level.

A 3D model of the corner of a building was created with representations of the different material layers within the walls and the floor. Users can investigate this detailed section by rotating and moving the model. Also, they can go beyond the outside surface by rendering layer by layer invisible thus gaining further insight into the structure of the design.

A 3D model of a room is used as a means of testing spatial orientation and interaction modes. Different types of beams can be selected from a 3D menu and varied in number by using a second menu. The user positions the beams within the room, altering type and number to test different design options.

Finally, an urban context model of London serves as testing grounds for both architectural as well as urban design. This scenario exists in a number of variations. Among them are the scaling and positioning of a simple model of the recently completed SwissRe high-rise building in London as well as the testing of system performance using a model of the same building with substantially higher level of detail. The most complex of these variations is the creation of a church on a freed-up space in the centre of the urban model. All forms of interaction – PHO, 3D pointer and gestures – are combined to allow users to operate 3D menus, create new geometries and manipulate them as desired. Form creation is achieved by these means in a quality not yet available to architects and urban designer.

4.2. Integration with Simulation Software

Urban planning decisions typically require a thorough consideration of different alternatives involving aspects such as the overall design, the affordability, or the impact on the local environment. While costs can be easily calculated, design decisions are more involved. They are often based on real models and their impact is investigated by studies or simulations. The decision process itself, however, is often lacking a comprehensive mechanism to combine the relevant information. In our approach we integrate the various aspects into a single Augmented Reality table-top planning environment facilitating more efficient decision finding. We demonstrate the approach with a pedestrian simulation within a cityscape scenario.

We have implemented spatial agents that respond dynamically to the changes in locations of objects on the table. The movement rules for these agents are based on space syntax theory. Space syntax deals with the configurational properties of environments. Spatial agents use vision to assess the configuration, and move towards open space by a stochastic process: choosing a destination at random from the available space, and walking towards it.

Figure 5. Users positioning agents with the 3D pointer.

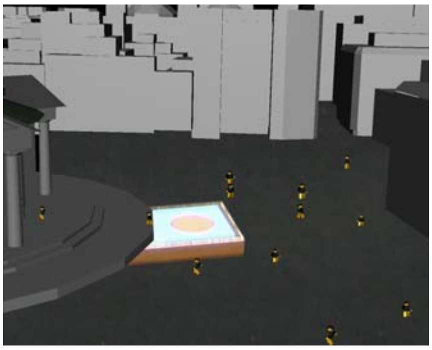

Figure 6. Near view of the Augmented Reality based interactive pedestrian simulation.

In this way, they are configurational explorers. The rules are: walk 3 steps, look around and choose a new destination, walk 3 steps, and so on. If their field of view is set to 170º (approximately human vision) the agents start to move, on aggregate, in a human like manner. In an experiment within a 1.5 km square area of the City of London, there is a correlation coefficient of $R^2 = 0.67$ between agents and actual pedestrian numbers measured at 78 "gates" randomly located within the area [30] [20]. Thus, while the movement of an individual agent will typically not resemble those of a human equivalent that much, the overall movement of a large number of agents is quite meaningful and allow the users to draw conclusions regarding the position of new buildings.

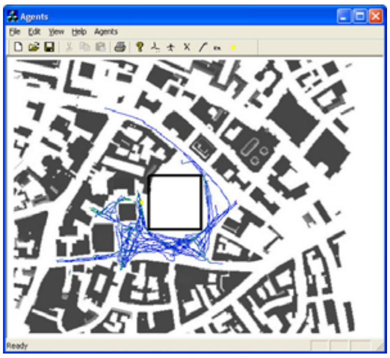

Figure 7. Screenshot of the pedestrian simulation application

The system allows placement of a specified number of agents within the cityscape scenario. The MORGAN API allows the agents simulation engine to talk to the augmented reality application. A simple object model feeds the new agent locations to the front end whilst the front end feeds current building locations to the simulation engine. Agent animations are then executed locally at the front-end using a dead-reckoning algorithm. The actual walking speed is determined by the distance of the new position. As soon as a new position update is received, the new direction is locally re-calculated for the individual agent and the new travel path is started. All agents share a single walking animation in order to keep the visualization scalable to even several hundred agents at the same time. It is possible to set initial starting locations and directions of the agents, for instance simulating the effect of transportation terminals within the system. Buildings are usually attached to a PHO to be moved (e.g. the cathedral, see Figure 5 and Figure 6). The agent simulation program will prevent agents to choose a path through dynamic objects (see Figure 7). Currently the representation of these dynamic objects within the agent simulation is limited to the bounding boxes of these objects.

4.3. Seamless Integration with a CAD System

Architectural design without the possibility to make changes to the virtual model is not better than real static models. Since the results of a review have to be integrated into the virtual model by individuals using CAD software. Often these designers did not even take part in the review process. To overcome this limitation, a major CAD system has been integrated into the ARTHUR system. Currently, CAD system integration into AR or VR usually means exporting the CAD data into a different 3D graphics format like VRML97 [4]. This integration has several drawbacks, like loss of geometry or precision and the loss of semantics. Most importantly no real-time interaction or modification is possible without recursively converting between the different file formats.

Either by using the 3D pointer or finger gestures, the user is able to draw directly in front of him without sitting at a CAD workbench. Virtual menus located above the augmented round table allow selecting different operations within the CAD software, e.g. drawing a sphere, a torus or a b-spline as well as changing colors and extruding surfaces. After selecting an operation, the user is able to draw these objects in 3D space. Apart from drawing objects the user is also able to move and delete objects. Basically all operations of the CAD software can be integrated into the ARTHUR system.

To ensure that the CAD model stays consistent all the time, only one instance of the CAD system is running. Each user interaction is send to the CAD integrating component, which ensures synchronization of different users. Using the update mechanism of the CAD system, the component distributes the results of the interaction to the attached visualization components. By sending the current position of the 3D pointer or the fingers, the use of a mouse inside the CAD system could be simulated, allowing the ARTHUR system to show dynamically, how shapes would look like during their creation.

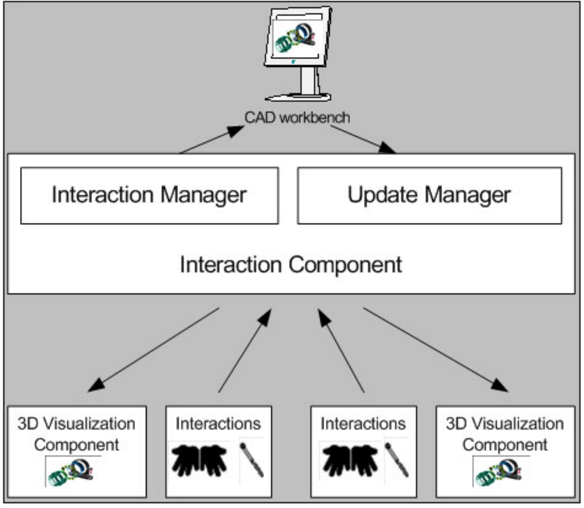

Figure 8. Information flow for CAD integration

As mentioned in section 3.1.1 the CAD integration component is an external scene graph within the AR framework. While the internal scene graph does not care about object identifiers within the CAD system, these are stored in the integration component. This mapping between the two internal references allows minimal data transfer between the components, since operations like deleting a CAD object only results in deleting the corresponding object in the internal scene graph. Additionally, users can interact with the representations in 3D and actually interact with the CAD objects; since the integration component can tell which internal scene graph object represents the CAD object. Representations of the CAD objects does not mean that the objects in the internal scene graph are less complex, it just means they are stored in a different way within the internal scene graph and the integration component.

Since the update mechanism of the CAD system is used, the interaction is not limited to the interaction with 3D pointers or finger gestures. During the same session user can also apply changes to the CAD model using the CAD software, this way more complex changes can be made and other CAD models can be loaded or referenced inside the current file. Depending on the size of these CAD models, the distribution can take several seconds up to minutes if the model is large. Architectural data is typically very detailed and complex, models of several million polygons are common. But because architects are used to wait for realistic rendering results, this delay is accepted.

Additionally, since the CAD data is not converted, all semantic information is preserved. This is also true for behaviour which is attached to objects, e.g. kinematics. The update mechanism is sufficient to filter out changes being made since the last update has occurred, therefore ARTHUR system is able to display even interactive data which originates from the CAD system, this not only includes kinematics, but also the dynamics during the creation of shapes. Of course, the amount of interactive data is limited by means of network bandwidth and network latency.

User tests have shown, that experienced CAD users where able to use the integration without training. The look-a-like of the menus and the dynamics during the creation of shapes let them think to work inside the CAD software.

5. User Evaluation

Architectural design has always been linked to the available design tools. Although it is difficult to describe precisely the relation between the two, it is clear that specific projects today could not be realized without the computer [28].

The ARTHUR system bridges the long-time gap between today’s common design tools and provides architects with new design and interaction possibilities.

Figure 9. Two users collaborating at the augmented round table

Although CAD-systems have been widely used by architects since the eighties [11], their potential has never truly exceeded that of a powerful drawing and visualisation tool. Programmes such as “SketchUp” have improved the situation to a degree, offering a quick and easy-to-use interface better suited for design.

Nonetheless, Design itself is still predominantly achieved by use of hand drawings and physical sketch models. Decision makers in architectural practices for a long time remained sceptical or unaware of the potential that remained unused by limiting the use of computers [26].

ARTHUR provides designers with a new instrument that links digital 3D-models to interaction mechanisms similar to those of the real world. Interaction is simple and intuitive. Contrary to existing, highly complex, often confusing CAD interfaces that stand in the way of design, ARTHUR provides a simple and intuitive interface ideal for design creation.

Furthermore, it enables designers to truly enter into a collaborative form of design beyond the mode of taking turns or creating individually thus far not provided by any other design tool. It seems that one of the most interesting aspects of the ARTHUR development lies in what it seems to tell us about the way that designers collaborate [Penn et. al]. Two different kinds of behaviour were noticeable in our user tests of collaboration. In the first, one member of the team would take charge of the process, and direct actions. This is common in design teams in most architects’ offices. In the second, collaborators began to play games, particularly when users were faced with simulated pedestrian movement. We believe that creating architectural forms and working on a task collaboratively became a game that users enjoyed and as a result this enhanced the level of collaboration [22] [8].“The actual making of decisions about forms in space - had a strong and inevitable social dimension and as such was influenced by the way in which involved parties interacted” [16].In addition, the appearance of the agents on the design table encouraged the users to understand structures within space as a dynamic experience rather than a static one (through agents moving between spaces) and the interaction with the agents became an integral part of the designers’ conversation with the emerging design on the design table [8].

Our findings to date suggest this and indicate that the ARTHUR-systems defines new standards for human-computer-interaction in design whilst staying true to core requirements of architectural form creation based on visual perception, spatial relations [1] and social behaviour

6. Conclusions and Future Work

In this paper we introduced the ARTHUR system to support close collaboration between architects for complex design and planning decisions. We have explained the ARTHUR system with its major components, such as the MORGAN framework, the AR HMDs, the computer-vision based head and object tracking and GRAIL for configuring the user interface. In our tests with architects and other users we experienced that ARTHUR supports the understanding of spatial structures and close collaboration between meeting participants.

In our future work we will enhance the ARTHUR system by using more flexible and light-weight components (such as lighter displays, fire wire cameras) and further reduce the necessary set-up and preparation time (e.g. by using camera-based position tracking only). The application scenarios will be further evaluated and improved based on the user feedback. So far this will include the integration of additional more advanced simulation programs (including but not limited to solar heat gain and wind simulations) as well as a more advanced integration with the CAD system (and possibly the extension to other CAD systems).

7. Acknowledgements

The ARTHUR project is partially funded by the European Commission as part of the IST programme within the 5th framework (project no. IST-2000-28559).

8. References

[1] Aicher, O., “Analog und Digital”, *Analog und Digital*, Ernst & Sohn Verlag, 1991, pp. 45-50.

[2] Azuma, R.T., “A Survey of Augmented Reality”, *Presence: Teleoperators and Virtual Environments*, 6 (4), 1997, pp. 355-385.

[3] Azuma, R.T., Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre, “Recent Advances in Augmented Reality”,

IEEE Computer Graphics and Applications, 21 (6), 2001, pp. 34-47.

[4] Berta, J., “Integrating VR and CAD”, *IEEE Computer Graphics and Applications,* 19 (5), 1999, pp. 14-19.

[5] Billinghurst, M., H. Kato, and I. Poupyrev, “The MagicBook - Moving Seamlessly between Reality and Virtuality”, *IEEE Computer Graphics and Applications,* 21 (3), 2001, pp. 2-4.

[6] Broll, W., M. Störring, and C. Mottram, “The Augmented Round Table - a new Interface to Urban Planning and Architectural Design”, *Proceedings of INTERACT 2003*, 2003, pp.1103-1104.

[7] Dias, J.M.S., P. Santos, and N. Diniz, “Tangible Interaction for Conceptual Architectural Design”, *Proceedings of ART 2002*, 2002.

[8] Fatah gen. Schieck, A. Penn, C. Mottram ”Interactive Space Creation through Play”, In the 8th International Conference: Information Visualisation IV04, London, VGRU, (to be published), 2004.

[9] Feiner, S., A. Webster, T. Krueger, B. MacIntyre, and E. Keller, “Architectural Anatomy”, *Presence: Teleoperators and Virtual Environments*, 4(3), 1995, pp. 318-325.

[10] Feiner, S., T. Höllerer, B. MacIntyre, and A. Webster, ”Augmented Reality for construction - A collaborative project at Columbia University between the Graphics and User Interfaces Lab in the Computer Science Department and the Building Technologies Group in the Graduate School of Architecture“, 1996, www1.cs.columbia.edu/graphics/projects/arc/arc.html.

[11] Feldhusen, G.,“Zur Geschichte von Theorie und Praxis des CAD“, *CAD: Architektur automatisch?*, 1986, pp. 93–103.

[12] Friedrich, W., ARVIKA — Augmented Reality for Development, Production and Service, *Proceedings of ISMAR 2002*, Darmstadt, Germany, 2002.

[13] Gausemeier, J. , J. Fruend, C. Matysczok, “AR-Planning Tool – Designing Flexible Manufacturing Systems with Augmented Reality”, *Proceedings of Eurographics Workshop on Virtual Environments 2002, Barcelona, Spain, 2002.*

[14] Granum, E., T. Moeslund, M. Stoerring, W. Broll, and M. Wittkaemper. Facilitating the Presence of Users and 3D Models by the Augmented Round Table, *In PRESENCE Conference*, Aalborg, Denmark, 2003.

[15] Grasset, R., and J.D. Gascuel, “MARE: Multiuser Augmented Reality Environment on table setup”, Proceedings of *SIGGRAPH 2002, Conference Abstracts and Applications, Computer Graphics Annual Conference Series 2002*, 2002, p. 213.

[16] Habraken, N.J., and M.D. Gross, “Concept Design Games”, *Design Studies*, 9(3), 1988, p. 181.

[17] Ishii, H., and B. Ullmer, “Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms”, *Proceedings of CHI 1997,* Atlanta, Georgia, 1997, p. 234-241.

[18] Ishii, H., J. Underkoffler, D. Chak, and B. Piper, “Augmented Urban Planning Workbench: Overlaying Drawings, Physical Models and Digital Simulation”, *Proceedings of IEEE and ACM International Symposium on Mixed and Augmented Reality*, Darmstadt, 2002.

[19] Liu, Y., M. Stoerring, T.B. Moeslund, C.B. Madsen, E. Granum, Computer Vision Based Head Tracking from Re-configurable 2D Markers for AR, *Proceedings of the Second IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2003)*, Tokyo, Japan, 2003, pp. 264 -265.

[20] Moeslund, T.B., M. Stoerring, Y. Liu, W. Broll, I. Lindt, C. Yuan, and M. Wittkaemper. Towards Natural, Intuitive and Non-intrusive HCI Devices for Roundtable Meetings. *Workshop on Multi-User and Ubiquitous User Interfaces (MU3I)*, Funchal, Madeira, Portugal, 2004.

[21] Penn, A., and A. Turner, “Space Syntax based Agent Simulation”, *Pedestrian and Evacuation Dynamics*, Springer-Verlag, 2002, pp. 99-114.

[22] Penn, A., C. Mottram, A. Fatah gen. Schieck, M. Wittkämper, M. Störring, O. Romell, A. Strothmann, and F. Aish, “Augmented reality meeting table: a novel multi-user interface for architectural design”, *Recent Advances in Design and Decision Support Systems in Architecture and Urban Planning – selected papers*, Kluwer Academic Publishers (to be published), 2004.

[23] Poupyrev, I., D.S. Tan, M. Billinghurst, H. Kato, H. Regenbrecht, and N. Tetsutani, “Tiles: A Mixed Reality Authoring Interface”, *INTERACT 2001 Conference on Human Computer Interaction,* Tokyo, Japan, 2001.

[24] Rauterberg, M., M. Fjeld, H. Krueger, M. Bichsel, U. Leonhardt, and M. Meier, “BUILD-IT: a Video-based Interaction Technique of a Planning Tool for Construction and Design”, *Proceedings of work With Display Units - WWDU’97,* Tokyo, 1997, pp. 175-176.

[25] Regenbrecht, H.T., M.T. Wagner, and G. Baratoff, “MagicMeeting: A Collaborative Tangible Augmented Reality System”, *Virtual Reality* (6), 2002, pp. 151-166.

[26] Rosson, M.B., J. M. Carroll, Scenario-based usability engineering. *In Proceedings of Symposium on Designing Interactive Systems*, 2002.

[27] Schmitt (a), G.N., “Computer als Medium”, *Architektur mit dem Computer*, Vieweg Verlagsgesellschaft, 1996, p. 53.

[28] Schmitt (b), G.N., “Neue Instrumente der Architekturinformatik”, *Architektur mit dem Computer*, Vieweg Verlagsgesellschaft, 1996, p. 54.

[29] Thomas, B., W. Piekarski, and B. Gunther, “Using Augmented Reality to Visualise Architecture Designs in an Outdoor Environment“, *Design Computing on the Net*, http://www.arch.usyd.edu.au/kcdc/conferences/dcnet99, Sydney, 1999.

[30] Turner, A. “Analysing the Visual Dynamics of Spatial Morphology”. *Environment and Planning B: Planning and Design*, 30, 5, 2003, pp. 657–676.

[31] Webster, A., S. Feiner, B. MacIntyre, W. Massie, and T. Krueger, “Augmented Reality in Architectural Construction, Inspection and Renovation”, *Proceedings of the ASCE 3rd Congress on Computing in Civil Engineering,* Anaheim, CA, 1996, pp. 913-919.

[32] Zauner, J., M. Haller, A. Brandl, “Authoring of a Mixed Reality Assembly Instructor for Hierarchical Structures”, *Proceedings of the Second IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2003)*, Tokyo, Japan, 2003, pp. 237-247.