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

Wolfgang Broll[*], Irma Lindt[*], Jan Ohlenburg[*], Michael Wittkämper[*], Chunrong Yuan[*],
Thomas Novotny[*], Ava Fatah gen. Schieck[†], Chiron Mottram[†], Andreas Strothmann[‡]

[*] Fraunhofer Institute for Applied Information Technology (FIT)
Sankt Augustin, Germany
email: `wolfgang.broll@fit.fraunhofer.de`
[†] The Bartlett, University College London
London, UK
[‡] Linie 4 Architekten
Aachen, Germany

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 in 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 ARTHUR system, an Augmented Reality (AR) enhanced round table 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.

First presented at the International Conference of Human and Computer 2004, extended and revised for JVRB

Keywords: Augmented Reality, Tangible User Interfaces, Architectural Design, Urban Planning, Simulation, CAD

1 Introduction

Architectural design and urban planning – at least for sophisticated projects – have always involved highly cooperative tasks. Individual phases within a project often change between close cooperative situations, for instance, during design and review meetings, and individual work carried out by the participants or third parties. While between these meetings, people involved of course have a common goal they contribute to, they will however focus 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 particular solutions is once more performed by the individuals (leaving the final decision to one of the following meetings). Real collaboration is often limited to the creation of very early design sketches. From an architect's point of view it would be desirable to have an additional support tool allowing to improve the cooperation in a way that supports real collaboration within meetings. This would in turn allow for much faster design and review cycles.

In this paper we will present our approach to an AR system supporting the different phases of architectural design and urban planning. The goal of our approach was to provide an environment and tools to support the collaboration between the experts involved in these particular meetings, without radically altering or even replacing established working procedures, or accepted tools and mechanisms. Rather, we tried to enhance the meeting situations by 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 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 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 a look into future work in section 6.

2 Related Work

Recently, several approaches to deploy AR in the area of architectural design and urban planning have been made. For construction sites, for instance, 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 [FWK$^+$95, WFM$^+$96]. 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. [TPG99] developed, for example, a system for an outdoor visualization of construction plans. Designs are exported from a CAD application and displayed in their physical outdoor context using the *TINMITH2 system*.

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* [RFK$^+$97]. 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 [RWB02] supports product review meetings by augmenting a real meeting location. Instead of real mock-ups, virtual 3D models are used, which 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 [BKP01], annotations and a clipping plane tool. The 3D models are linked to physical placeholder objects and realize a tangible interface.

AR-Planning Tool [GFM02] 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 planned 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* [IBJU$^+$02], 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 [Fri02] realized AR applications for development, production and servicing. One of the applications 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). They integrate planning rules (compare AR Planning Tool) and sophisticated interaction metaphors (compare MagicMeeting), 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 important issue is the amount of modifications which may be applied to the original model using the facilities of the AR system. In ARTHUR geometry is not just imported from modelling software, but can be directly created within the AR environment. Additionally, ARTHUR integrates a full professional CAD system to support more advanced functionality, such as 3D sketching and extrusions.

3 The ARTHUR System

The ARTHUR system is based on several individual components, including the MORGAN AR framework, AR displays, computer vision (CV) based input mechanisms, as well as a Graphical User Interface configuration. The major system components are discussed in more detail in the following subsections. For further information on the individual components of the ARTHUR system see also [MSL+04] and [GMS+03].

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 [OHL+04].

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 an optical see–through augmentation, i.e., the virtual image is optically superimposes the real environment. The 3D visualization component, additionally supports see–through video augmentation, whereby the image from a head–mounted camera is superimposed by the virtual scene components. This technique is used for screen–based, or projection–based presentations, as well as for non–see–through head mounted displays.

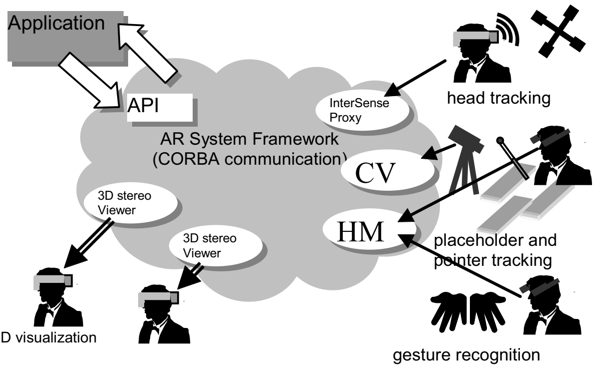


Figure 1: The basic components of the AR framework MORGAN

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 preserve individual native scene graph structures. This architecture allows us to use, for instance, VRML'97 for the description of the 3D user interface elements, as well as for objects created within ARTHUR, while e.g. CAD objects may use their own individual external scene graph (see also section 4.3). The scene graph has been enhanced in order to specifically support AR (phantom objects, video and image backgrounds) and fundamental AR/VR interaction techniques (highlighting of objects, universal picking, and collision detection). In the overall system, there exists one visualization component 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.

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 InertiaCube2), 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 1).

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, SaabTech developed the AddVisor 150 – a new head mounted display (HMD) (see Figure 2) to fit the particular needs of the environment and the users. The developed HMD is a see-through 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 HMD was 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, which 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.

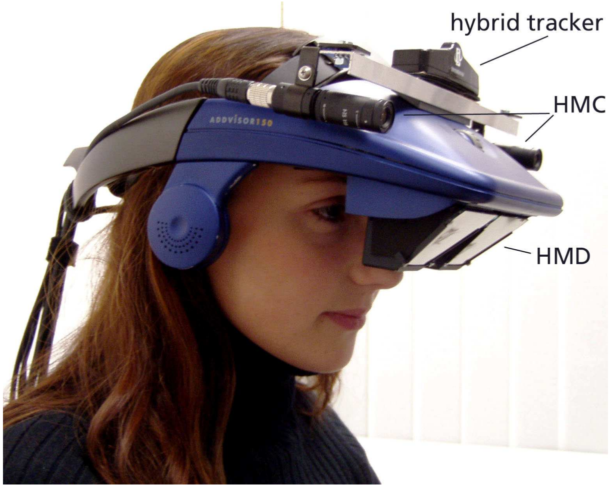


Figure 2: ARTHUR See–through head mounted display (HMD).

3.3 Input Mechanisms

In our approach, computer vision (CV) provides the basis for user input. We distinguish between marker based input devices, such as placeholder objects and wands, and natural input, such as hand gestures and fingertip tracking.

The user interface devices are dedicated objects that are tracked by the CV system [MSL$^+$04] using color 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 can be grabbed and moved by the users (see Figure 3). By associating PHOs to one or several virtual objects or functions, a tangible user interface is created.

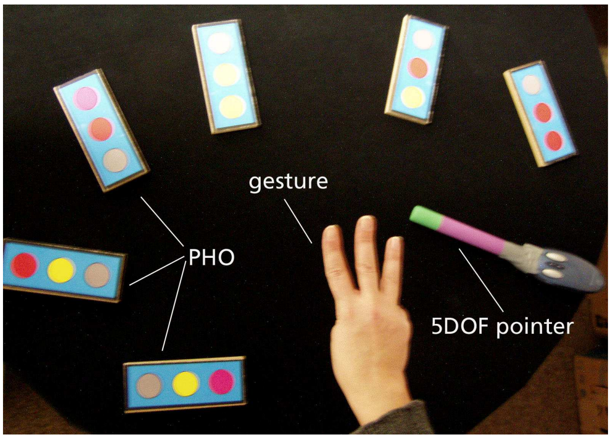


Figure 3: Input image for CV–based recognition of PHOs, pointers, and finger gestures.

Pointers are tracked in 5 DOF and provide three buttons, which may be used to initiate operations, such as picking or selecting. Users may select and manipulate the shape of virtual objects with a pointer, or use it to navigate in virtual menus.

Additionally, two types of hand-based input may be used, namely static command gestures and 3D fingertip tracking. 3D fingertip tracking may be used to draw a line in space, or, to navigate in pop–up menus, select items and execute actions.

3.4 Graphical User Interface Configuration

A graphical language (GRAIL) was developed to allow users to configure the relationships between the input mechanisms (3DOF PHO, the 5DOF pointers, and the command gestures) and the virtual objects. GRAIL also provides the user with the ability to link

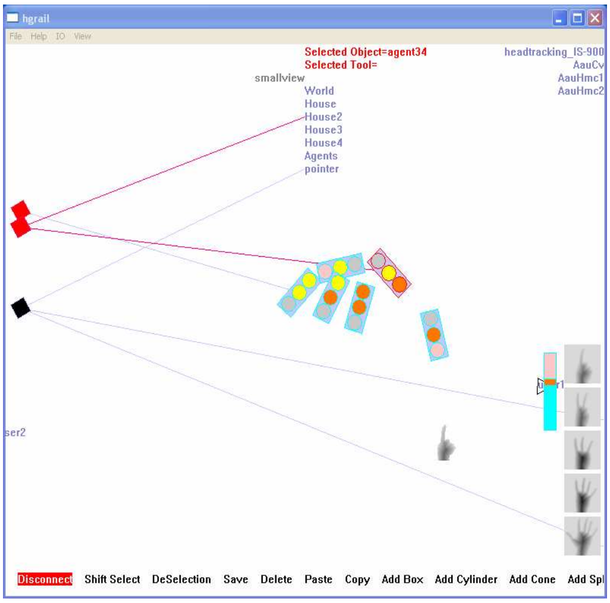


Figure 4: GRAIL application

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 resides on top of the MORGAN API (see section 3.1) 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 input mechanisms and virtual objects (see Figure 4). For instance, a two–finger gesture could be used as a *tool* for creating virtual boxes in the AR environment.

4 Application Scenarios

In this section we will describe three application scenarios, which are typical for the use of the ARTHUR environment. The first scenario represents a typical design or review meeting situation. In the second scenario the presentation of complex data enhanced by simulations is demonstrated. The third scenario finally demonstrates the close integration with existing CAD software.

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 [RC02] was adopted for developing the system. The three 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.

4.1 Supporting Architectural Design and Review Meetings

In this scenario, an urban context model of the City of London served as a testing ground for both architectural as well as urban design. The scenario is based on a real-world example, namely the recently completed Swiss-Re high–rise building. Issues involved in designing and planning this building, and which can be supported by the ARTHUR environment, included the decision about the building area, the basic shape of the building, the adaptation of this shape with respect to the environment (i.e. surrounding buildings), the overall review of the final model, and the final presentation.

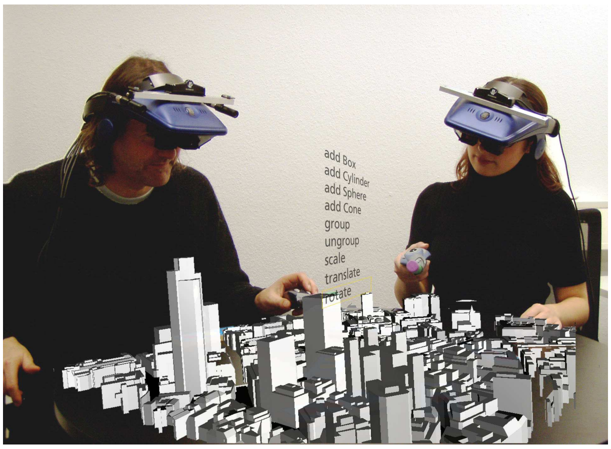


Figure 5: Two users collaborating at the augmented round table

For the first task, the model of London was displayed on the table–top, allowing the users to find a suitable building space. This included the possibility to remove existing buildings in order to create an appropriate freed–up space within the model. User were then free to introduce new basic 3D objects and manipulate them as a matter of 3D sketching. While the final basic shape ("‘the cigar"’) was created outside the ARTHUR environment, the system allowed users to manipulate this shape interactively by using placeholder objects on the table or the 3D wand. Once the final shape was decided upon, an additional fine tuning regarding the size and exact location within the desired space was performed. For the final review, the rather coarse design model was replaced by a more refined model of the actual building (see Figure 5).

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 complex. They are often based on real models, and their impact needs to be investigated by studies or simulations. The decision process itself, however, is often lacking a comprehensive mechanism to combine all relevant information. In our approach, we integrate the various aspects into the ARTHUR system, facilitating a more efficient decision finding. We will demonstrate the approach with a pedestrian simulation within a cityscape scenario.

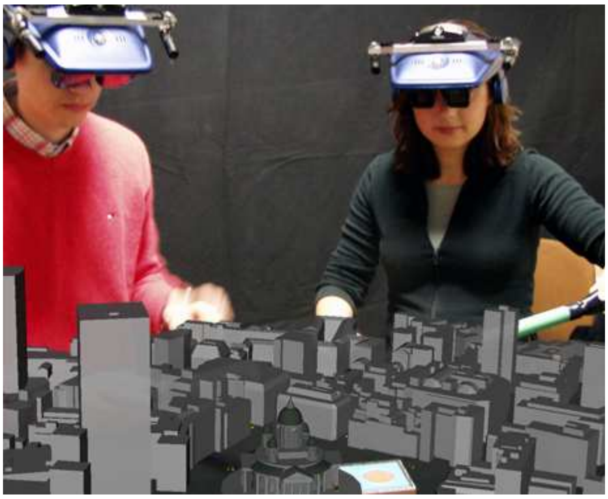


Figure 6: Users adding pedestrian agents at the 3D pointer location.

We have implemented spatial agents that respond dynamically to changes in the 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, i.e. by choosing a destination at random from the available space, and walking towards it.

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 170° (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 [Tur03, MSL$^+$04]. Thus, while the movement of an individual agent will typically not that closely resemble those of a human equivalent that much, the overall movement of a large number of agents is quite insightful and allow the users to draw conclusions regarding the position of new buildings.

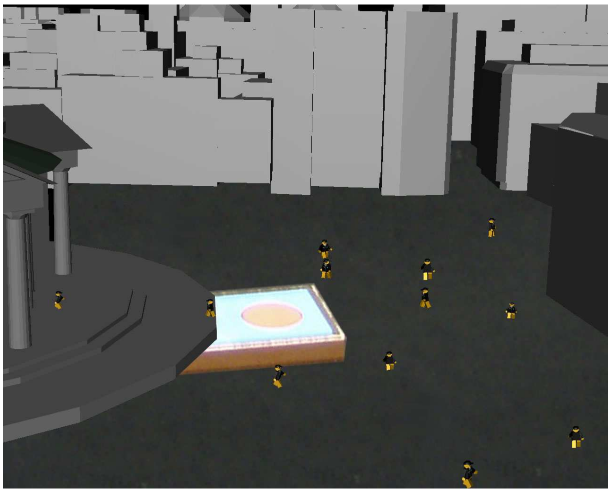


Figure 7: Close-up of the Augmented Reality based interactive pedestrian simulation.

The system allows placement of a specified number of agents within the cityscape scenario. The MORGAN API allows the agent simulation engine to talk to the augmented reality application. A simple object model feeds the agents' new 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 simultaneously. It is possible to set up the agents' starting locations and directions, for instance, for simulating the effect of transportation terminals within the system. Buildings are usually attached to a PHO in order to be moved (e.g. the cathedral, see Figure 6 and Figure 7). The agent simulation program will prevent agents from choosing a path leading through dynamic objects (see Figure 8). Currently, the representation of these dynamic objects within the agent simulation is limited to the bounding boxes of these objects.

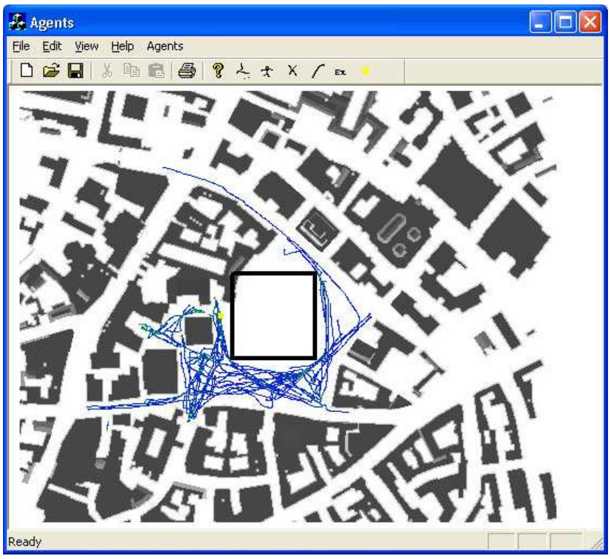


Figure 8: Screenshot of the pedestrian simulation application

4.3 Seamless Integration with a CAD System

Architectural design without the possibility to make changes to the virtual model does not provide any major advantages compared to the use of real static models, since the results of a review have to be integrated into the virtual model individually using CAD software. Frequently, designers actually carrying out this integration task, did not even take part in the review process.

So far, CAD system integration into AR or VR usually meant exporting the CAD data into a specific 3D graphics format, such as VRML'97. This method has several drawbacks, including the loss of geometrical details or precision, and the loss of object semantics. The major drawback however, is that an interactive modification of the original model is not possible, either because it results in multiple back and forth conversions between the individual formats, or, because it requires all changes applied to be repeated within the original CAD system.

To overcome these limitations, a development prototype of the forthcoming version of the Microstation CAD system has been integrated into the ARTHUR system. This allows users to model their objects in 3D above the table–top (see Figure 10) without the need to switch back to the 2D CAD desktop (see Figure 9). Users may create or interact with objects either by using the 3D pointer or finger gestures. Virtual menus located above the augmented round table allow for performing different operations directly within the CAD software, e.g. for creating 3D geometry (such

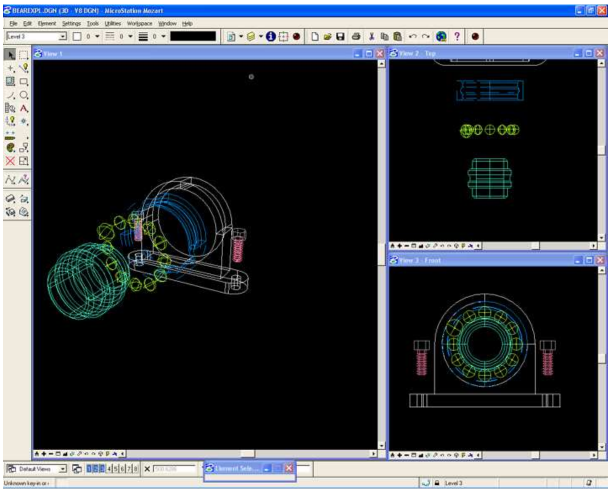


Figure 9: 3D model within the original CAD application

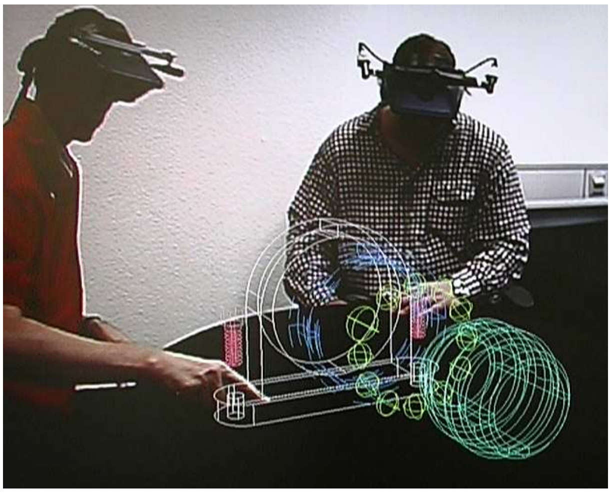


Figure 10: Same 3D model within the ARTHUR environment

as drawing a sphere, a torus, or a b–spline), changing colors, extruding surfaces, and moving, copying or deleting objects. Basically, all operations available within the CAD software can be integrated into the ARTHUR system. Therefore 3D menus providing appropriate entries representing the commands or user interface elements of the CAD system have to be supplied to the ARTHUR system (see Figure 11). This step is rather simple, as it only requires the specification of an appropriate user interface description file using XML (see Figure 12). By selecting or executing the 3D user interface elements, the appropriate commands in conjunction with relevant 3D input data are then sent to the CAD software.

Only a single instance of the CAD system is executed at a time, ensuring a consistent CAD model.

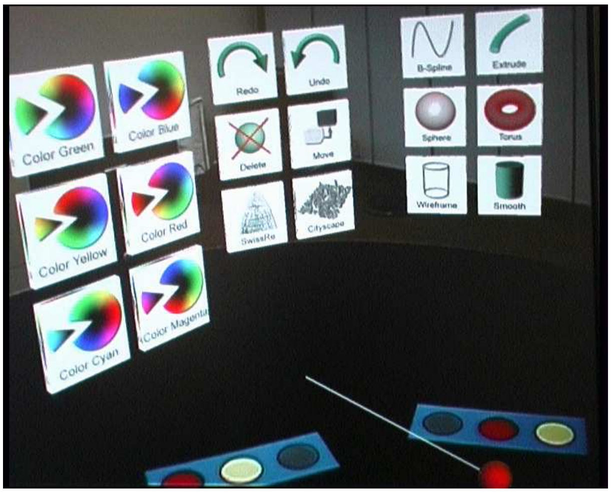
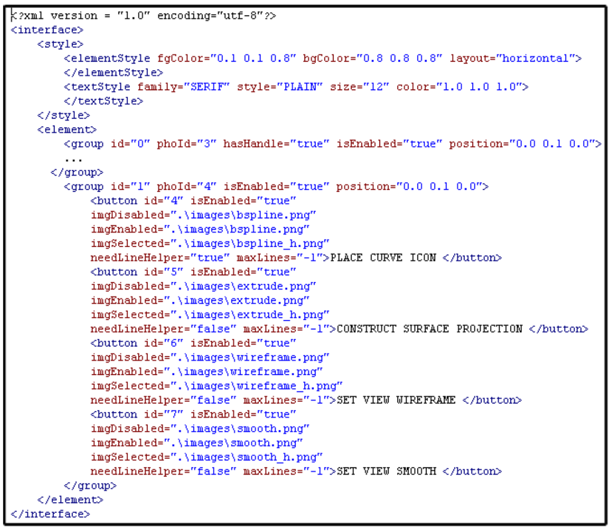


Figure 11: 3D menus representing CAD software commands



```
<?xml version = "1.0" encoding="utf-8"?>
<interface>
    <style>
        <elementStyle fgColor="0.1 0.1 0.8" bgColor="0.8 0.8 0.8" layout="horizontal">
        </elementStyle>
        <textStyle family="SERIF" style="PLAIN" size="12" color="1.0 1.0 1.0">
        </textStyle>
    </style>
    <element>
        <group id="0" phoId="3" hasHandle="true" isEnabled="true" position="0.0 0.1 0.0">
        ...
      </group>
        <group id="1" phoId="4" isEnabled="true" position="0.0 0.1 0.0">
            <button id="4" isEnabled="true"
            imgDisabled=".\images\bspline.png"
            imgEnabled=".\images\bspline.png"
            imgSelected=".\images\bspline_h.png"
            needLineHelper="true" maxLines="-1">PLACE CURVE ICON </button>
            <button id="5" isEnabled="true"
            imgDisabled=".\images\extrude.png"
            imgEnabled=".\images\extrude.png"
            imgSelected=".\images\extrude_h.png"
            needLineHelper="false" maxLines="-1">CONSTRUCT SURFACE PROJECTION </button>
            <button id="6" isEnabled="true"
            imgDisabled=".\images\wireframe.png"
            imgEnabled=".\images\wireframe.png"
            imgSelected=".\images\wireframe_h.png"
            needLineHelper="false" maxLines="-1">SET VIEW WIREFRAME </button>
            <button id="7" isEnabled="true"
            imgDisabled=".\images\smooth.png"
            imgEnabled=".\images\smooth.png"
            imgSelected=".\images\smooth_h.png"
            needLineHelper="false" maxLines="-1">SET VIEW SMOOTH </button>
        </group>
    </element>
</interface>
```

Figure 12: XML sample file

Each user's input is sent to the CAD integration component, which ensures the synchronization of the views of the individual users. Using the update mechanism of the CAD system, the component distributes the results of the interaction to the attached visualization components. Since the CAD data is not converted, all semantic information is preserved. This is also true for behavior attached to objects, such as kinematics.

User tests revealed that experienced CAD users where able to use the integrated system even without training. The main reasons were the similarity in the menu structure as well as its look and feel, and the natural and intuitive 3D input facilities provided by the ARTHUR input mechanisms.

5 User Experience

Although CAD–systems have been widely used by architects, their potential has never truly exceeded that of a powerful drawing and visualization tool. Nonetheless, design itself is still predominantly achieved by means of hand drawings and physical sketch models.

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, and often confusing CAD interfaces that stand in the way of design creativity, ARTHUR provides a simple and intuitive interface for design creation.

Furthermore, it enables designers to truly enter into a collaborative form of design, which is beyond the mode of taking turns or creating individually, thus far not provided by any other design tool. Apparently, one of the most interesting aspects of the ARTHUR development lies in what it seems to reveal to us about the way designers collaborate.

However, it was also noticed that such a system probably will not replace existing CAD or desktop–based 3D–modelling software in the near future. For several tasks, especially as far as a high precision is required, traditional user interfaces still provide significant advantages regarding the quality and the time needed.

Two different kinds of behavior 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 [PMgS^{+}04, gSPM04]. "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" [HG88]. In addition, the agents' appearance 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 [gSPM04].

Our findings to date suggest this and indicate that the ARTHUR system provides new chances, but also challenges, for human–computer–interaction in design, whilst staying true to core requirements of architectural form creation based on visual perception, spatial relations [Aic91] and social behavior.

6 Conclusion and Future Work

In this paper we presented the ARTHUR system supporting close collaboration between architects for complex design and planning decisions. We introduced its major components including the MORGAN AR framework 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, firewire cameras) and further reduce the necessary set-up and preparation time. Based on user feedback, application scenarios will be evaluated and improved.

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 integration of additional CAD systems and 3D modelling tools).

7 Acknowledgements

The authors wish to thank all members of the ARTHUR consortium for their contributions. The ARTHUR project was partially funded by the European Commission as part of the IST programme within the 5th framework (IST–2000–28559).

References

[Aic91] Otl Aicher, *Analog und Digital*, Ernst & Sohn, 1991, ISBN 3-433-02176-7.

[BKP01] Mark Billinghurst, Hirokazu Kato, and Ivan Poupyrev, *The Magic-Book - Moving Seamlessly between Reality and Virtuality*, IEEE Computer Graphics and Applications **21** (2001), no. 3, 6–8, ISSN 0272-1716.

[Fri02] Wolfgang Friedrich, ARVIKA - *Augmented Reality for Development, Production and Service*, Proceedings of the IEEE/ACM International Symposium on Mixed and Augmented Reality ISMAR 2002, IEEE Computer Society, 2002, ISBN 0-7695-1781-1, pp. 3–4.

[FWK+95] Steven K. Feiner, Anthony C. Webster, Theodore Krueger, Blair McIntyre, and Edward J. Keller, *Architectural Anatomy*, Presence: Teleoperators and Virtual Environments **4** (1995), no. 3, 318–325, ISSN 1054-7460.

[GFM02] Jürgen Gausemeier, Jürgen Fründ, and Carsten Matysczok, *AR-Planning Tool - Designing Flexible Manufacturing Systems with Augmented Reality*, Proceedings of the Eighth Eurographics Workshop on Virtual Environments, Eurographics Association, 2002, pp. 19–25.

[GMS+03] Erik Granum, Thomas B. Moeslund, Mortiz Störring, Wolfgang Broll, and Michael Wittkämper, *Faciliating the Presence of Users and 3D Models by the Augmented Round Table*, Proceedings of the 6th Annual International Workshop on Presence, October 2003.

[gSPM04] Ava Fatah gen. Schieck, Alan Penn, and Chiron Mottram, *Interactive Space Creation Through Play*, Proceedings of the 8th International Conference: Information Visualization IV04, 2004.

[HG88] N. John Habraken and Mark D. Gross, *Concept design games*, Design Studies **9** (1988), no. 3, 150–158.

[IBJU+02] Hiroshi Ishii, Eran Ben-Joseph, John Underkoffler, Luke Yeung, Dan Chak, Zahra Kanji, and Ben Piper, *Augmented Urban Planning Workbench: Overlaying Drawings, Physical Models and Digital Simulation*, Proceedings of the IEEE/ACM International Symposium on Mixed and Augmented Reality ISMAR 2002, IEEE Computer Society, 2002, ISBN 0-7695-1781-1, pp. 203–211.

[MSL+04] Thomas B. Moeslund, Moritz Störring, Yong Liu, Wolfgang Broll, Irma Lindt., C. Yuan, and Michael Wittkämper, *Towards Natural, Intuitive and Non-intrusive HCI Devices for Roundtable Meetings*, Proceedings of the workshop on Multi-User and Ubiquitous User Interfaces (MU3I), October 2004, pp. 25–29.

[OHL+04] Jan Ohlenburg, Iris Herbst, Irma Lindt, Torsten Fröhlich, and Wolfgang Broll, *The* MORGAN *Framework: Enabling Dynamic Multi-User AR and VR Projects*, In Proceedings of VRST 2004, 2004, pp. 166–169.

[PMgS+04] Alan Penn, Chiron Mottram, Ava Fatah gen. Schieck, Michael Witkämper, Moritz Störring, Odd Romell, Andreas Strothmann, and Francis 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 (Jos P. van Leeuwen and H. Timmermans, eds.), Kluwer Academic Publishers, 2004, ISBN 1-4020-2409-6, pp. 213–231.

[RC02] Mary Beth Rosson and John M. Carroll, *Scenario-based usability engineering*, Proceedings of the Symposium on Designing Interactive Systems, 2002, ISBN 1-58113-515-7, p. 413.

[RFK+97] Matthias Rautenberg, Morten Fjeld, Helmut Krueger, Martin Bichsel, Uwe Leonhardt, and Markus Meier, BUILD-IT*: A Video-based Interaction Technique of a Planning Tool for Construction and Design*, In Proceedings of Work with Display Units WWDU'97, 1997, pp. 175–176.

[RWB02] Holger Regenbrecht, Michael Wagner, and Gregory Baratoff, *Magic-Meeting: A Collaborative Tangible Augmented Reality System*, Virtual Reality **6** (2002), no. 3, 151–166, ISSN 1434-9957 (online).

[TPG99] Bruce Thomas, Wayne Piekarski, and Bernard K. Gunther, *Using Augmented Reality to Visualise Architecture Designs in Outdoor Environment*, International Journal of Design Computing: Special Issue on Design Computing on the Net (DCNet'99) **2** (1999), ISSN 1329-7147.

[Tur03] Alan Turner, *Analysing the Visual Dynamics of Spatial Morphology*, Environment and Planning B: Planning and Design **30** (2003), no. 5, 657–676.

[WFM+96] Anthony Webster, Steven Feiner, Blair MacIntyre, William Massie, and Theodore Krueger, *Augmented Reality in Architectural Construction, Inspection and Renovation*, In Proceedings of ASCE 3rd Congress on Computing in Civil Engineerings, 1996, pp. 913 – 919.

Citation

Wolfgang Broll, Irma Lindt, Jan Ohlenburg, Michael Wittkämper, Chunrong Yuan, Thomas Novotny, Ava Fatah gen. Schieck, Chiron Mottram, and Andreas Strothmann, ARTHUR*: A Collaborative Augmented Environment for Architectural Design and Urban Planning*, Journal of Virtual Reality and Broadcasting, 1(2004), no. 1, December 2004, urn:nbn:de:0009-6-348, ISSN 1860-2037.