

# **Lessons Learned in Designing Ubiquitous Augmented Reality User Interfaces**

Christian Sandor and Gudrun Klinker  
Technische Universität München, Institut für Informatik  
Boltzmannstraße 3, Garching bei München, Germany  
(sandor, klinker)@in.tum.de

## **1 Introduction**

In recent years, a number of prototypical demonstrators have shown that augmented reality has the potential to improve manual work processes as much as desktop computers and office tools have improved administrative work (Azuma et al., 2001; Ong & Nee, 2004). Yet, it seems that the "classical concept" of augmented reality is not enough (see also <http://www.ismar05.org/IAR>). Stakeholders in industry and medicine are reluctant to adopt it wholeheartedly due to current limitations of head-mounted display technology and due to the overall dangers involved in overwhelming a user's view of the real world with virtual information. It is more likely that moderate amounts of augmented reality will be integrated into a more general interaction environment with many displays and devices, involving tangible, immersive, wearable and hybrid concepts of ubiquitous and wearable computing. We call this emerging paradigm Ubiquitous Augmented Reality (UAR) (Sandor & Klinker, 2005; MacWilliams, 2005; Sandor, 2005).

It is not yet clear which UAR-based human-computer interaction techniques will be most suitable for users to simultaneously work within an environment that combines real and a virtual elements. Their success is influenced by a large number of design parameters. The overall design space is vast and difficult to understand.

In Munich, we have worked on a number of applications for manufacturing, medicine, architecture,

exterior construction, sports and entertainment (a complete list of projects can be found at <http://ar.in.tum.de/Chair/ProjectsOverview>). Although many of these projects were designed in the short-term context of one semester student courses or theses, they provided insight into different aspects of design options, illustrating trade-offs for a number of design parameters. In this chapter, we propose a systematic approach towards identifying, exploring and selecting design parameters at the example of three of our projects, PAARTI (Echtler et al., 2003), FataMorgana (Klinker et al., 2002) and a monitoring tool (Kulas, Sandor, & Klinker, 2004).

Using a systematic approach of enumerating and exploring a defined space of design options is useful, yet not always feasible. In many cases, the dimensionality of the design space is not known a-priori but rather has to be determined as part of the design process. To cover the variety of aspects involved in finding an acceptable solution for a given application scenario, experts with diverse backgrounds (computer science, sensing and display technologies, human factors, psychology, and the application domain) have to collaborate. Due to the highly immersive nature of UAR-based user interfaces, it is difficult for these experts to evaluate the impact of various design options without trying them. Authoring tools and an interactively configurable framework are needed to help experts to quickly set up approximate demonstrators of novel concepts, similar to "back-of-the-envelope" calculations and sketches. We have explored how to provide such first-step support to teams of user interface designers (Sandor, 2005). In this chapter, we report on lessons learned on generating authoring tools and a framework for immersive user interfaces for UAR scenarios.

By reading this chapter, readers should understand the rationale and the concepts for defining a scheme of different classes of design considerations that need to be taken into account when designing UAR-based interfaces. Readers should see how, for classes with finite numbers of design considerations, systematic approaches can be used to analyze such design options. For less well-defined application scenarios, the chapter presents authoring tools and a framework for exploring interaction concepts. Finally, a report on lessons learned from implementing such tools and from discussing them within expert teams of user interface designers is intended to provide an indication of progress made thus far and next steps to be taken.

## 2 Background

In this section, we provide an overview of the current use of UAR-related interaction techniques and general approaches towards systematizing the exploration of design options.

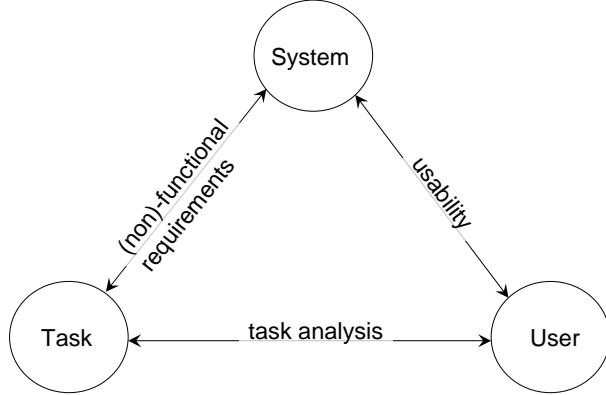
### 2.1 User Interface Techniques for Ubiquitous Augmented Reality

User interfaces in UAR are inspired by related fields, such as virtual reality (VR) (Bowman, Kruijff, LaViola, & Poupyrev, 2004), attentive user interfaces (AUIs) (Vertegaal, 2003), and tangible user interfaces (TUIs) (Ishii & Ullmer, 1997). Several interaction techniques for VR have been adopted to UAR: for example the World-in-Miniature (Bell, Höllerer, & Feiner, 2002), pinch gloves for system control (Piekarski, 2002), and a flexible pointer to grasp virtual objects beyond arm's reach (Olwal & Feiner, 2003). The core idea of TUIs is to use everyday items as input and output simultaneously. This idea has also been applied to UAR (Klinker, Stricker, & Reiners, 1999; Kato, Billinghurst, Poupyrev, Tetsutani, & Tachibana, 2001; MacWilliams et al., 2003). Ideas from AUIs have been used in UAR interfaces by using head tracking (Olwal, Benko, & Feiner, 2003) and eye tracking (Novak, Sandor, & Klinker, 2004).

### 2.2 Implementing New Interaction Techniques

To develop new interaction techniques and visualizations for UAR, several software infrastructures have been created to simplify the development of new interaction techniques by programmers: distributed frameworks, dataflow architectures, user interface management systems, scenegraph-based frameworks, a variety of class libraries, and finally scripting languages. A detailed discussion can be found in (Sandor, 2005).

For novice users, several desktop tools for authoring augmented reality content have been developed: PowerSpace (Haringer & Regenbrecht, 2002), DART (MacIntyre, Gandy, Dow, & Bolter, 2004) and MARS (Güven & Feiner, 2003). Several systems exist that follow an immersive authoring approach (Poupyrev et al., 2001; Lee, Nelles, Billinghurst, & Kim, 2004). Piekarski describes a mobile



**Figure 1. Design criteria clasified according to tasks, systems, and users.**

augmented reality system that can be used to capture the geometries of real objects (Piekarski, 2002). Several hybrid authoring approaches combine immersive authoring with desktop authoring (Zauner, Haller, Brandl, & Hartmann, 2003; Olwal & Feiner, 2004).

### 3 Design Optimization for High-Dimensional Design Spaces

One of the most difficult issues in designing novel interaction techniques for UAR is the wealth of criteria that are potentially involved in finding an optimal solution. We divide such criteria into three classes: criteria pertaining to the task(s) that need to be executed, the knowledge and skills of the user, and the current state-of-the-art of technology. Figure 1 illustrates the classes and their relationships.

#### 3.1 Classes of Design Criteria

**Task-specific criteria** are related to the requirements of specified tasks in an application. According to principles of *software engineering* they are determined from scenarios and use cases, taking the environmental setting and the required technical quality into account. Yet, they may change over time due to changing work processes, which may indirectly depend on evolving technology.

**System-specific criteria** are defined by the state-of-the art of *engineering-related* parameters of sensing and display devices and computer systems. Due to evolving technology, these criteria have to be continuously reevaluated, resulting in ever-changing optimal system configurations (Klinker et al., 1999).

**User-specific criteria** depend on ergonomic issues and the cultural background of users—*human factors* and *anthropology*. They describe current working conditions, habits (working culture), and educational background, as well as specific user-related restrictions.

### 3.2 Criteria Reduction Through Inter-Class Constraints

Finding an overall optimal system that works perfectly with respect to all criteria seems to be impossible. We have thus adopted the approach of selecting specific criteria of one or two classes to impose constraints on design options in other classes. In this section, we analyze the relationship between the classes of criteria from user-, system-, and task-centric specifications. Section 4 illustrates the exploitation of such constraints in specific examples.

The relationships between **task and system** requirements are described by the edge linking the task and system nodes in Figure 1. From the *task*-perspective, they are described as the *functional and non-functional requirements* of software systems. From the *system*-perspective, they need to be matched with the currently available technical options. Trade-offs have to be cast to obtain pragmatically implementable solutions—with an eye towards upcoming requirements and technical developments. In Section 4.1, we will present an example of casting such trade-offs.

The relationships between **task and user** requirements are described by the edge linking the task and user nodes in Figure 1. This case does not involve any considerations of currently available technology. Thus, options that are discussed here should hold true now, as well as 100 years ago or 100 years in the future. They are analyzed by disciplines such as *task analysis* and *system ergonomics* (Bubb, 1993). Yet, they can provide significant constraints upon today’s technically achievable system configurations. In Section 4.2, we will present an example of analyzing how a user (car designer) physically behaves with respect to a number of tasks geared towards analyzing and comparing different automotive designs.

The relationships between **system and user** are described by the edge linking the system and user nodes in Figure 1. From the *user*-perspective, they are described as *usability* criteria, evaluating how users perform, given a specific technical system—in comparison to other technical options. From the *system*-perspective, they describe user requirements that need to be satisfied with currently available

technical means. In Section 4.3, we show an example of how a specific technical device can be evaluated with respect to specific physical user skills in using such a device.

### 3.3 Dealing with Ill-Defined Design Spaces

By applying inter-class constraints on a user interface, the design space can often be reduced considerably. The next step in our proposed design process is to explore the reduced design space with interactive tools that encourage collaboration. In this section, we first give the rationale for our interactive tools. Then, we proceed by highlighting the problems that occur when using this approach. Finally, we give an outlook of how we elaborate on these concepts within this chapter.

To further explore the design space, a collaboration between researchers with different backgrounds is imperative, to yield a solution that is well-balanced according to our three main classes: user, task and system. Thinking about this problem led to the invention of a new development process: *Jam Sessions*. The name Jam Sessions was inspired by the spontaneous collaboration of Jazz musicians that is also named Jam Sessions. However, in our case we collaborate on user interface elements, instead of music. In Jam Sessions, development takes place at system runtime, next to a running system. This allows playful exploration of user interface ideas. Our experience with Jam Sessions was first presented in (MacWilliams et al., 2003); we have already discussed these from a software engineering (MacWilliams, 2005) and user interface (Sandor, 2005) perspective.

To support this development process, interactive tools for novices are an important ingredient, since they foster the interdisciplinary collaboration with other researchers. Desirable would be a set of generic tools that can be applied in all Jam Sessions—*independent of the user interface to be developed*. Although we have achieved this for programmers, for novices this is yet an unsolved problem. We go in line with several other research tools that allow to modify only a quite limited amount of user interface functionality. Since these tools are customized towards the user interface that has to be built, most projects require to write new tools. Thus, a sophisticated software infrastructure that quickly allows to build new tools is very useful.

Section 5 describes a complex user interface that we have designed in Jam Sessions. Additionally, we

first describe briefly our software infrastructure and elaborate on the tools that we have created for this project.

## 4 Projects using Inter-Class Constraints

This section presents three examples of analyzing design options by exploring inter-class constraints.

### 4.1 PAARTI

In the PAARTI project (Practical Applications of Augmented Reality in Technical Integration), we have developed an intelligent welding gun with BMW that is now being used on a regular basis to weld studs in the prototype production of cars (Echtler et al., 2003). It exemplifies the systematic exploitation of constraints between task and system criteria.

The **task** was to assist welders in positioning the tip of a welding gun with very high precision at some hundred predefined welding locations on a car body. The main **system** design issue was to find an immersive solution with maximal precision. An AR-based system would need a display ( $D$ ), a tracking sensor ( $S$ ), and some markers ( $M$ ) that needed to be installed in the environment on the user or on the welding gun in a manner that would yield maximal precision. As a fourth option, we considered the case that one of the objects (esp.: markers) would not be necessary at all. The result was the definition of a 3-dimensional design space,  $S \times M \times D$ , with each dimension spanning a range of four options. In total, there were  $4^3 = 64$  solutions that needed to be considered.

According to an analysis of all options, the highest precision could be achieved by using an outside-in tracking arrangement with sensors placed in the welding environment and markers attached to the welding gun. A small display was attached to the welding gun. The visualization used a notch and bead metaphor of real guns, consisting of several concentric rings. A sphere was positioned three-dimensionally at the next welding location. Welders were requested to capture the sphere within the concentric rings by moving the gun (and the display) to the appropriate location (Echtler et al., 2003).

## 4.2 FataMorgana

In the FataMorgana project, we have developed an AR-based prototypical demonstrator for designers at BMW, helping them compare real mockups of new car designs with virtual models (Klinker et al., 2002). The rationale for building this system was that – although the importance of digital car models is increasing – designers have not yet committed wholeheartedly to a VR-based approach but rather prefer relying on physical mockups. One of the reasons may be that special viewing arrangements such as projection walls do not permit people to view the digital models within a real environment. AR can help alleviate this problem by placing virtual cars next to real (mockup) cars.

We here present this project as an example of a systematic analysis of the relationships between **tasks** and **user** actions. The underlying thesis is that users (designers) behave in specific ways in order to achieve tasks. If a system is expected to support users in achieving their tasks, it has to be designed to function well within the range of typical actions performed by the user. To this end, we have subdivided the task in a set of different approaches and asked a designer to act out each of these tasks within the real car presentation environment. We recorded the designer's motions with a camera that was attached to his head.

**Turning:** The car designer remains in a fixed location and looks at the car rotating on a turn table.

**Overview:** The car designer performs an overview evaluation of the car, by walking around and turning his head to change the lighting conditions.

**Detail:** The car designer focuses on a specific detail of the car, such as a character line on the side of the car or the shape of the front spoiler.

**Discuss:** The car designer discusses the car under evaluation with a colleague.

**Compare:** The car designer compares two cars, for example, an existing car and a new design.

For each scenario, we determined the typical angular range of head rotations, as well as the range of positional changes. Combined with a projection of the field of view onto the environment, this gave us an indication how markers had to be laid out in the room in order to guarantee that enough of them were clearly visible during all user actions.

#### 4.3 Monitoring Tool for Determining Usage Patterns of Novel Interaction Techniques

We have developed a monitoring tool (Kulas et al., 2004) to evaluate the usability of novel input techniques and devices. The monitoring tool allows us to systematically analyze relationships between **user** and **system** criteria, analyzing whether a system is well tailored to the physiological and cognitive skills of its users.

We have used the monitoring tool to evaluate the usability of a novel input device called *TouchGlove* that was developed at Columbia University (Blasko & Feiner, 2002). It consists of a touch-sensitive plate (similar to a touchpad in a laptop) that is attached to the center of a user's palm. It is sensitive to single-finger input, measuring 2D location and pressure. In the evaluation setup, we have compared two techniques of using the TouchGlove to select items from a menu. In the first case, users were asked to make a linear gesture with their fingertip on the TouchGlove to select items from a regular pull-down menu. In the second case, the TouchGlove was combined with a gyroscope to select items from a pie menu. Users were asked to rotate their hands around their wrists, generating only a tapping signal on the touchpad to signal start and end of the gesture rotating the pie menu.

During a usability evaluation, the user is placed at a suitable distance from a usability engineer. The engineer enters observations into a usability logging system and also monitors what the user actually sees on screen. Simultaneously, he also monitors real-time visualizations of measured usability data. The tool provides immediate feedback during an interactive tryout session, thereby supporting Jam Sessions as discussed in Sections 3.3 and 5.

### 5 Interactive Tools for Collaborative Design Space Explorations

This section presents our tools for supporting Jam Sessions. First, we give a brief overview of our tools. Second, we present an interdisciplinary research project, CAR, that uses them. We close with a description of the underlying real-time development environment.

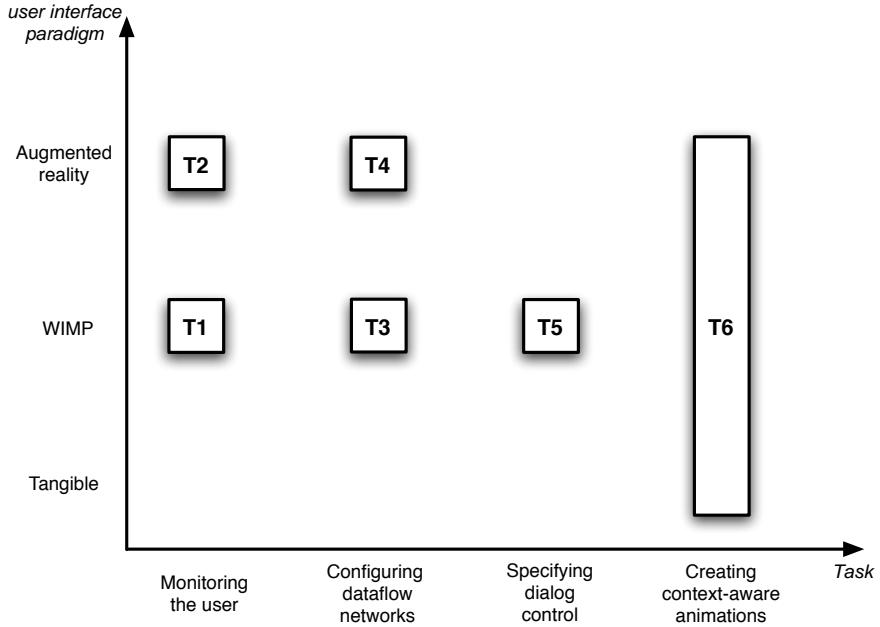
## 5.1 Overview of Tools

To support Jam Sessions, we have created a toolbox of lightweight and flexible tools. They form the basic building blocks which user interface development teams can use to generate, experience and test their novel interaction techniques.

The tools use AR, TUI and WIMP interaction paradigms and are designed to support a number of tasks. The first task concerns **monitoring the user**. (see also the discussion in Section 4.3). The second task involves the **configuration of dataflow networks**. UAR systems need to communicate in real-time with many sensing and display devices, requiring a distributed system approach. A dataflow network connects such devices and components. We provide tools that allow to modify these dataflow graphs during runtime. Another task is related to the **adjustment of dialog control**, i.e., the control of the high-level behavior of a user interface. Tools that enable developers to specify dialog control quickly speed up the development process significantly. The final task involves the **creation context-aware animations**. Conventional animations have *time* as the only parameter that changes the appearance of graphical elements. However, for mobile systems a variety of research projects (e.g., a context-aware World-in-Miniature (Bell et al., 2002)) have explored animations that change their appearance according to *context*.

We have developed 6 tools, T1 – T6, in support of these tasks. **T1** collects and evaluates usability data during system runtime (see Section 4.3).**T2** uses an augmented reality visualization to shows a user's visual focus of attention in a combination of head- and eyetracking (Novak et al., 2004) (see Section 5.2.3 and Figure 7). **T3** is a graphical editor, *DIVE* to adjust dataflow networks (MacWilliams et al., 2003; Pustka, 2003) (see Section 5.3 and Figure 8(a)). **T4** is an immersive visual programming environment (Sandor, Olwal, Bell, & Feiner, 2005) (see Section 6). **T5** is a User Interface Controller Editor, *UIC*, to graphically specify dialog control by composing Petri nets (Hilliges, Sandor, & Klinker, 2004) (see Section 5.3 and Figure 8(b)). **T6** is a collection of tools to experiment with context-aware mobile augmented reality user interfaces (Section 5.2).

Figure 2 classifies our tools with respect to the user interface paradigms they employ and the tasks



**Figure 2. Classification of implemented tools. Development tasks are addressed with tools that use different user interface paradigms.**

they address (Sandor, 2005). It shows that we sometimes developed several tools, addressing the same task, using different interaction paradigms. This reflects our goal of exploring and comparing design options for our own tools as much as for the interaction techniques that will be developed with them.

## 5.2 CAR

CAR is an industry-sponsored multi-disciplinary project to investigate issues pertaining to the design of augmented reality user interfaces in cars. CAR has used most of the tools T1 – T6 to investigate several user interface questions.

### 5.2.1 Motivation

In CAR we have investigated a variety of questions: How can information be presented efficiently across several displays that can be found in a modern car: e.g., the dashboard, the board computer and heads-up displays (HUDs)? How can we prevent that information displayed in a HUD is blocking the driver's

view in crucial situations? Since a wealth of input modalities can be used by a car driver (tactile, speech, head and hand gestures, eye motion): which modalities should be used for which tasks?

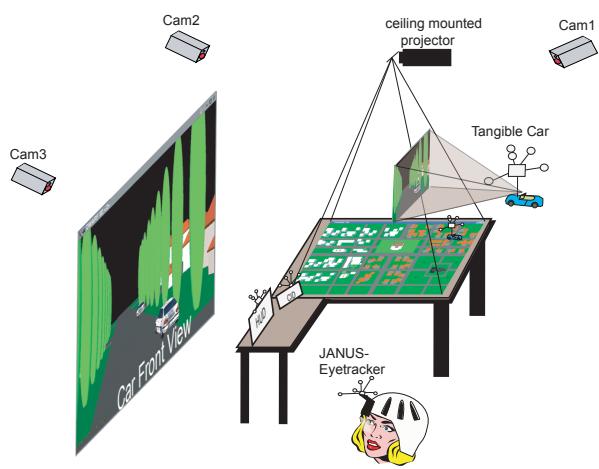
In a multi-disciplinary UI design team, we have discussed, for example, how to present a navigation map on a HUD. Where should it be placed? How large should it be? What level of detail should it provide? Should it be a two-dimensional map or a tilted view onto a three-dimensional environmental model (WIM)? If so, which viewing angle should be selected? Will the angle, as well as the position of the WIM and the size and zoom factor adapt to sensor parameters, such as the current position of the car while approaching a critical traffic area in a town?

### 5.2.2 Physical Setup

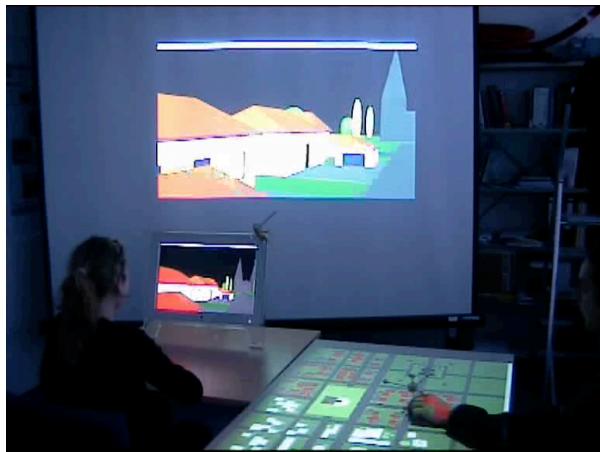
We have set up a simulator for studying car navigation metaphors in traffic scenes (Figure 3). It consists of two separate areas: a simulation control area (large table with a tracked toy car) and a simulation experience area (person sitting at the small table with a movable computer monitor in the front and a stationary large projection screen in the back). In the simulation control area, members of the design team can move one or more toy cars on the city map to simulate traffic situations, thereby controlling a traffic simulator via a tangible object. The simulation experience area represents the cockpit of a car and the driver. The picture projected on the large screen in the front displays the view a driver would have when sitting in the toy car. The monitor in front of the driver provides a mockup for the visualizations to be displayed in a HUD. Further monitors can be added at run-time, if more than one view is needed.

The room is equipped with an outside-in optical tracking system (<http://www.ar-tracking.de>). The cameras track the toy car, the computer monitor and the user (simulating a car driver). Each tracked object is equipped with a marker consisting of a rigid, three-dimensional arrangement of reflective spheres.

Information is presented on several devices and surfaces in the room: A projector at the ceiling projects a bird's eye view of a city onto the large, stationary table on the right. Another projector presents the current, egocentric view of a virtual car driver sitting in the toy car on the large screen at the front wall. A third, location-dependent visualization of the driving scenario is shown on the mobile



(a) Conceptual drawing.



(b) Photo of the actual setup.

**Figure 3. Physical setup.**

computer monitor—our substitute for a HUD.

The system provides tools for a team of design experts with diverse backgrounds to jointly explore various options to present a map (Figure 4).



(a)



(b)

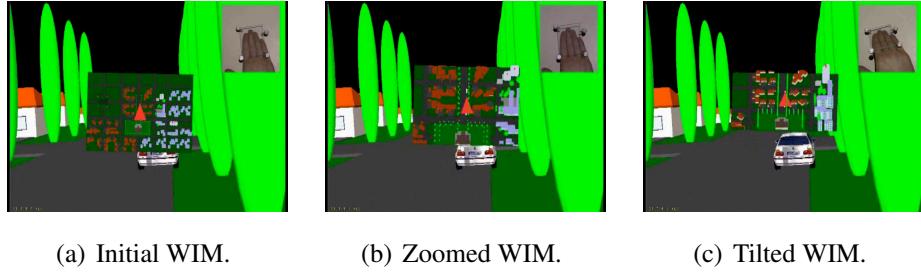
**Figure 4. Discussion of user interface options for car navigation in a design team.**

### 5.2.3 Controlling the Context-Aware Adjustment of a Navigation Map

It is not yet clear, how navigational aids are best presented within a driver's field of view. In the CAR project, we have experimented with various options of placing and orienting a map in a HUD.

Figure 5 shows how our system provides designers with a tangible object—a plate—that is correlated with the **orientation** (tilt) and **zoom** of a 3D map on the HUD: when the user moves the tangible plane, the 3D map is turned and zoomed accordingly on the HUD. Figure 4(a) shows a member of the design team experiment with different map orientations.

The **position** and **size** of a map in a HUD may have to depend on various parameters that depend on the driving context, such as the current position of the car relative to its destination, the driver's viewing direction, and immanent dangers in the environment. Interface designers need to explore schemes for the display system to automatically adapt to context parameters. Figure 6 shows an interactive sketching tool for designers to describe functional dependencies between context parameters and display options. Figure 7 shows first steps towards using tracked head and eye motions to provide a context-dependent interaction scheme (Novak et al., 2004). Figure 4(a) shows the head and eye tracking device. We are in the process of analyzing context-dependent information presentation further. First user studies of selected issues are presented in (Tönnis, Sandor, Klinker, Lange, & Bubb, 2005).

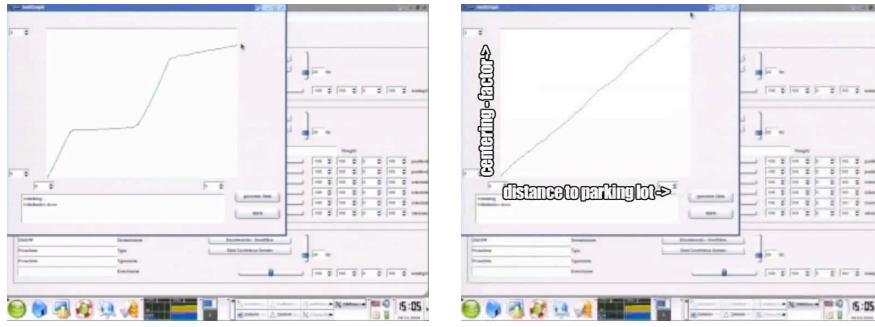


(a) Initial WIM.

(b) Zoomed WIM.

(c) Tilted WIM.

**Figure 5. Tangible interaction for adjustment of a three-dimensional map.**



(a) Staircase function.

(b) Linear function.

**Figure 6. Sketching the context-visualization function.**

### 5.3 Real-Time Development Environment for Interaction Design

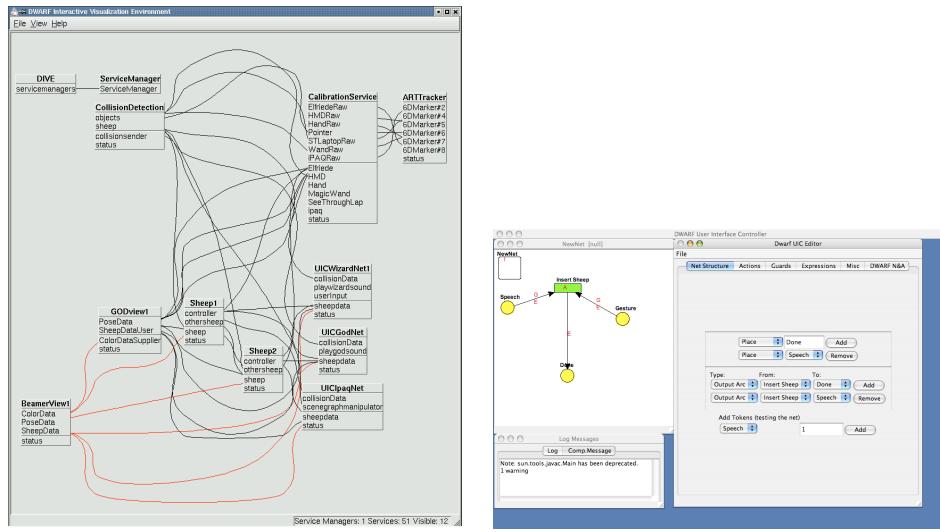
The tools presented in Section 5.2 were geared towards the immediate use by non-programming user interface experts. They mainly address the customization of a set of functionalities and filters, linking context measurements to information presentation schemes. In order to add new functionality to a system, the development team must also be able to modify the underlying network of components, and its dataflow scheme. Tools T3 and T5 of Section 5.1 provide such support.

All system configuration tools are based on DWARF (Distributed Wearable Augmented Reality Framework) (Bauer et al., 2001) and AVANTGUARDE (Sandor & Klinker, 2005; Sandor, 2005). DWARF is the underlying infrastructure that connects a set of distributed components. AVANTGUARDE is composed of DWARF components that address the specific requirements for user interfaces in UAR.

DWARF's Interactive Visualization Environment (MacWilliams et al., 2003) (tool T3, Figure 8(a)) en-



**Figure 7. Attentive user interface, visualizing a driver's eye and head motions**



(a) The DWARF’s Interactive Visualization Environment for managing distributed components.

(b) The User Interface Controller for specifying dialog control.

**Figure 8. Tools for programmers used in CAR.**

ables developers to monitor and modify the dataflow network of distributed components. However, since this requires substantial knowledge of DWARF and distributed programming, novices have difficulties to use this tool.

The core component of AVANTGUARDE is a Petri net based dialog control management system (Hilliges et al., 2004) (Tool T5, Figure 8(b)). We have developed a visual programming environment that eases the modification of the Petri nets (and accordingly the user interface) during system runtime. However, it is still too difficult to use for non-programming design experts, since understanding Petri nets requires knowledge in computer science.

## 6 Conclusion: Lessons Learned

In PAARTI and FataMorgana, we have learned that the reduction of criteria through inter-class constraints is a valuable approach for designing user interfaces. The crucial issue of this method is to determine the most important constraints by talking with domain experts. The subsequent systematic design space exploration is straightforward.

We have presented an example for the inter-class constraint of User and System: the evaluation of the TouchGlove input device. In this project, we have observed the importance of immediate feedback through interactive tools. Our first prototype of the TouchGlove had a loose contact. While we conducted the usability study with the first user, we immediately spotted the problem and solved it. This saved us a lot of valuable time.

Our tool-based approach for further design space explorations has been applied successfully in several projects. The idea of providing user interface developers with a toolbox of flexible, lightweight tools seems feasible. However, one problem has to be pointed out: when creating a variety of tools, a supporting real-time development environment is imperative. Otherwise, too much development time has to be allocated to tool creation – leaving little time for the actual use of the tools. In this respect, we have successfully built our tools on top of DWARF and AVANTGUARDE.

The combination of tools with different user interface paradigms turned out to be a valuable idea. We have made two important observations: first, there seems to be a trade-off between ease of use for a tool and the complexity of results that can be accomplished with it. WIMP tools can be used to model more complex interactions, whereas ease of use is greater with tools that have a tangible user interface or an augmented reality user interface. Second, the combination of tools with different paradigms opens new possibilities of interaction design, that would not be possible with tools employing a single paradigm. Interaction designers are typically not fluent in complex programming tasks, so their involvement with easy to use tools yields important benefits.

Ideally, it would be enough to create one generic tool that novices can use to explore the design space of UAR user interfaces. Our first prototype towards this goal has been published in (Sandor et al., 2005).

This tool seems to be very easy to use, as it employs only direct manipulation of real world objects – no conventional programming is required at all. However, the ceiling of the tool (i.e., what can be achieved with it) is quite low, since our system supports only a fixed, and very limited number of operations. We are exploring how we can extend it to allow users to specify new operations at runtime. While we anticipate using programming-by-demonstration to address a carefully planned universe of possibilities, supporting arbitrary operations through demonstration and generalization is an open problem.

The CAR project also showed us that for design space explorations, rapid prototyping is more important than realism for finding new interaction techniques. Though, for the thorough evaluation of these new concepts, formal usability studies within a realistic environment are still necessary. We have conducted a first study in this respect (Tönnis et al., 2005).

## References

- Azuma, R., Baillot, Y., Behringer, R., Feiner, S., Julier, S., & MacIntyre, B. (2001). Recent advances in augmented reality. *IEEE Computer Graphics and Applications*, 21(6), 34–47.
- Bauer, M., Bruegge, B., Klinker, G., MacWilliams, A., Reicher, T., Riss, S., et al. (2001). Design of a component-based augmented reality framework. In *ISAR '01: Proceedings of the International Symposium on Augmented Reality* (pp. 45–54). New York, New York.
- Bell, B., Höllerer, T., & Feiner, S. (2002). An annotated situation-awareness aid for augmented reality. In *UIST '02: Proceedings of the 15th Annual ACM Symposium on User interface Software and Technology* (pp. 213–216). Paris, France: ACM Press.
- Blasko, G., & Feiner, S. (2002). A menu interface for wearable computing. In *ISWC '02: Proceedings of the 6th IEEE International Symposium on Wearable Computers* (pp. 164–165).
- Bowman, D. A., Kruijff, E., LaViola, J. J., & Poupyrev, I. (2004). *3D user interfaces: Theory and practice*. Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc.
- Bubb, H. (1993). Systemergonomische gestaltung. In H. Schmidtke (Ed.), *Ergonomie* (1st ed., pp. 390–420). München, Germany: Carl Hanser.

- Echtler, F., Sturm, F., Kindermann, K., Klinker, G., Still, J., Trilk, J., et al. (2003). The intelligent welding gun: Augmented reality for experimental vehicle construction. In S. Ong & A. Nee (Eds.), *Virtual and Augmented Reality Applications in Manufacturing*. London, UK: Springer Verlag.
- Güven, S., & Feiner, S. (2003). A hypermedia authoring tool for augmented and virtual reality. *New Review of Hypermedia*, 9(1), 89–116.
- Haringer, M., & Regenbrecht, H. (2002). A pragmatic approach to augmented reality authoring. In *ISMAR '02: Proceedings of the IEEE and ACM International Symposium on Mixed and Augmented Reality* (p. 237-245). Darmstadt, Germany.
- Hilliges, O., Sandor, C., & Klinker, G. (2004). A lightweight approach for experimenting with tangible interaction metaphors. In *MU3I '04: Proceedings of the International Workshop on Multi-user and Ubiquitous User Interfaces*. Funchal, Madeira, Spain.
- Ishii, H., & Ullmer, B. (1997). Tangible bits: Towards seamless interfaces between people, bits and atoms. In *CHI '97: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. Atlanta, USA: ACM.
- Kato, H., Billinghurst, M., Poupyrev, I., Tetsutani, N., & Tachibana, K. (2001). Tangible augmented reality for human computer interaction. In *Proceedings of Nicograph 2001*. Nagoya, Japan.
- Klinker, G., Dutoit, A., Bauer, M., Bayer, J., Novak, V., & Matzke, D. (2002). Fata morgana – a presentation system for product design. In *ISMAR '02: Proceedings of the IEEE and ACM International Symposium on Mixed and Augmented Reality*. Darmstadt, Germany.
- Klinker, G., Stricker, D., & Reiners, D. (1999). Augmented reality: A balancing act between high quality and real-time constraints. In *ISMAR '99: Proceedings of the 1st International Symposium on Mixed Reality* (pp. 325–346). Yokohama, Japan.
- Kulas, C., Sandor, C., & Klinker, G. (2004). Towards a development methodology for augmented reality user interfaces. In *MIXER '04: Proceedings of the International Workshop Exploring the Design and Engineering of Mixed Reality Systems*. Funchal, Madeira, Spain.
- Lee, G. A., Nelles, C., Billinghurst, M., & Kim, G. J. (2004). Immersive authoring of tangible augmented

- reality applications. In *ISMAR '04: Proceedings of the IEEE and ACM International Symposium on Mixed and Augmented Reality* (p. 172-181). Arlington, VA: IEEE Computer Society.
- MacIntyre, B., Gandy, M., Dow, S., & Bolter, J. D. (2004). Dart: A toolkit for rapid design exploration of augmented reality experiences. In *UIST '04: Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology* (p. 197-206). Santa Fe, New Mexico, USA.
- MacWilliams, A. (2005). *A decentralized adaptive architecture for ubiquitous augmented reality systems*. Phd thesis, Technische Universität München, München, Germany.
- MacWilliams, A., Sandor, C., Wagner, M., Bauer, M., Klinker, G., & Brügge, B. (2003). Herding sheep: Live system development for distributed augmented reality. In *ISMAR '03: Proceedings of the IEEE and ACM International Symposium on Mixed and Augmented Reality* (pp. 123–132). Tokyo, Japan.
- Novak, V., Sandor, C., & Klinker, G. (2004). An AR workbench for experimenting with attentive user interfaces. In *Proceedings of IEEE and ACM International Symposium on Mixed and Augmented Reality* (pp. 284–285). Arlington, VA, USA.
- Olwal, A., Benko, H., & Feiner, S. (2003). Senseshapes: Using statistical geometry for object selection in a multimodal augmented reality system. In *ISMAR '03: Proceedings of the the 2nd IEEE and ACM International Symposium on Mixed and Augmented Reality* (pp. 300–301). Washington, DC, USA: IEEE Computer Society.
- Olwal, A., & Feiner, S. (2003). The flexible pointer—an interaction technique for selection in augmented and virtual reality. In *UIST '03: Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology* (pp. 81–82). Vancouver, BC.
- Olwal, A., & Feiner, S. (2004). Unit: Modular development of distributed interaction techniques for highly interactive user interfaces. In *GRAPHITE '04: International Conference on Computer Graphics and Interactive Techniques* (p. 131-138). Singapore: ACM Press.
- Ong, S., & Nee, A. (2004). *Virtual and augmented reality applications in manufacturing*. London, UK: Springer Verlag.
- Piekarski, W. (2002). *Interactive 3D modelling in outdoor augmented reality worlds*. Phd thesis,

University of South Australia.

- Poupyrev, I., Tan, D. S., Billinghurst, M., Kato, H., Regenbrecht, H., & Tetsutani, N. (2001). Tiles: A mixed reality authoring interface. In *INTERACT '01: 7th Conference on Human-Computer Interaction* (pp. 334–341). Tokyo, Japan.
- Pustka, D. (2003). *Visualizing Distributed Systems of Dynamically Cooperating Services*. Unpublished master's thesis, Technische Universität München.
- Sandor, C. (2005). *A software toolkit and authoring tools for user interfaces in ubiquitous augmented reality*. Phd thesis, Technische Universität München, München, Germany.
- Sandor, C., & Klinker, G. (2005). A rapid prototyping software infrastructure for user interfaces in ubiquitous augmented reality. *Personal Ubiquitous Comput.*, 9(3), 169–185.
- Sandor, C., Olwal, A., Bell, B., & Feiner, S. (2005). Immersive mixed-reality configuration of hybrid user interfaces. In *Proceedings of IEEE and ACM International Symposium on Mixed and Augmented Reality*. Vienna, Austria.
- Tönnis, M., Sandor, C., Klinker, G., Lange, C., & Bubb, H. (2005). Experimental evaluation of an augmented reality visualization for directing a car driver's attention. In *Proceedings of IEEE and ACM International Symposium on Mixed and Augmented Reality*. Vienna, Austria.
- Vertegaal, R. (2003). Attentive user interfaces. *Communications of ACM, Special Issue on Attentive User Interfaces*, 46(3).
- Zauner, J., Haller, M., Brandl, A., & Hartmann, W. (2003). Authoring of a mixed reality assembly instructor for hierarchical structures. In *ISMAR '03: Proceedings of the IEEE and ACM International Symposium on Mixed and Augmented Reality* (p. 237-246). Tokyo, Japan.