

# Enabling Interactive Augmented Prototyping by a Portable Hardware and a Plug-In-Based Software Architecture

Jouke Verlinden\* - Imre Horváth

Delft University of Technology, Faculty of Industrial Design Engineering, The Netherlands

*Interactive Augmented Prototyping (IAP) combines digital and physical modeling means to support design processes. Although pilot implementations indicate a possible added value, practical use is hindered by the fact that no off-the-shelf solution exists. Based on empirical studies and assessment of emerging technologies, this article introduces a projector-based IAP hardware platform called I/O Pad. Furthermore, a flexible software architecture is presented that supports a multitude of input devices and usage scenarios. In this architecture, an existing 3D CAD system is extended by a collection of plug-ins. The plug-ins are responsible for managing specific elements of the interactive augmented prototyping process. A first implementation has been developed, using a small projector and a handheld PC, which proves the wireless versatility of the hardware platform. The proposed software architecture allows the designer to work in a familiar modeling environment yet includes powerful concepts from tangible user interfaces to support several types of interaction with physical components.*

© 2008 Journal of Mechanical Engineering. All rights reserved.

**Keywords:** augmented reality, interactive prototyping, software architecture, CAD

## 1 INTRODUCTION

In supporting design and prototyping activities, Augmented Reality technology provides an appealing solution that combines physical and virtual reality. This combination might eliminate some of the problems associated to an entirely virtual or physical application. Several researchers have explored this concept of Interactive Augmented Prototyping, e.g. [2], [7], [14] and [23]. In an earlier publication [22] existing applications and enabling technologies were surveyed. The presented augmented prototyping systems showcased the power of tangible computing as natural, embodied interaction. Several augmentation techniques could be found in Milgram and Koshino's reality-virtuality continuum [13] to mix physical prototypes with digital imagery.

At present, the required technologies constitute a wide palette of input, processing, and output principles; devices and algorithms that are unknown for traditional CAD developers or users. Furthermore, these solutions typically rely on custom-coded applications that are incompatible with existing CAD systems (both user interface and modeling capabilities differ). With some

exceptions, typical Augmented Reality systems are bulky and prone to noise.

This article presents a system to support IAP, based on three extensive case studies in which design processes were empirically followed. The proposed solution comprises both hardware and software and is targeted to enrich the design process.

### 1.1 Empirical Findings

In order to obtain insight in the possibilities and limitations of current prototyping practice in industrial design, three design projects in different sub domains have been monitored: the design of a tractor, a handheld oscilloscope, and the interior for a museum [21]. These represent a range of industrial design engineering domains, which were considered to be susceptible to support by IAP (automotive, information appliances, and furniture design). Our objective was to produce a deep and accurate account of prototyping and modeling activities, with a primary focus on product representation and design reviews.

The field studies were used as a starting point to compile characteristics and specific events

\*Corr. Author's Address: Delft University of Technology, Faculty of Industrial Design Engineering, Landbergstraat 15, 2628 CE Delft The Netherlands, j.c.verlinden@tudelft.nl

that influenced the design processes, which we grouped in four different perspectives: functionalist, interpretive, emancipatory, and postmodern. The four perspectives originate from organizational sciences and relate to different objectives that designers might have for a particular representation, respectively: i) efficiency, ii) increasing shared understanding, iii) influencing decision making, and iv) creativity [24]. This body of findings was then used to identify specific IAP functions, grouped per scenario as discussed in the following section. Furthermore, the present physical prototypes have been analyzed to identify non-functional requirements. Some of the requirements pertaining to hard- and software will be reiterated and expanded in sections 2 and 3.

## 1.2 Scenarios

Based on the three case studies and the four perspectives, a number of IAP functions were identified. Main categorization aspect is a particular usage scenario, which differs among the domains and intended prototyping goals:

- User studies – evaluating intermediate designs by using the prototype as a stimulus. The prototype also acts as an excuse to study the user in its natural habitat and to provoke comments on product specifications.
- Exploration – probing various aspects of the design to diverge or understand underlying relationships; effectively creating design proposals, sometimes in combination with extensive simulation means.
- Design review – making design decisions, discussing design alternatives and considering the strengths and weaknesses as perceived by different stakeholders.
- Presentations to customers, higher management – inspire and possibly overwhelm distant stakeholders or public with (intermediate) results and possibly show user studies.

At present, the collection includes over 29 functions, summarized in the appendix of this article. However, in developing IAP support, we do not strive to deliver a Swiss army knife that covers all of these functions in a single module. Instead, smaller subsets of functions can be linked to the situation at hand and this will determine what IAP hard- and software configuration most

appropriate; this customization will be discussed in section 4.3.

## 1.3 Related Work

In earlier publications, we have selected projector-based augmented reality systems as the most likely candidate to support design [21]. The principle of projector-based augmented reality is treated in Bimber and Raskar [3]; it provides computational solutions to the challenges of projector-based Augmented Reality. Issues like registration of virtual and physical coordinate systems, calibration of colors and the simultaneous use of multiple projectors. However, the solutions are fragmented and not implemented in a single platform.

Conversely, a number of Augmented Reality software architectures have emerged, for example Studierstube [17], Avalon [1], and Avocado [18]. These deal with position tracking, virtual camera updating and provide hooks to script interactivity. All existing software platforms focus on video-mixing or see-through systems, and require adaptation to support projectors. Furthermore, the integration with modeling and simulation has not been addressed to a generic level. The main focus is to in support OpenGL based rendering libraries or X3D/VRML-based scene graph management. Story-based AR systems like DART [12], Geist [10], AMIRE [28] and the APRIL language [11] focus on playing narrative experiences in AR systems. All these solutions do not directly fit high-level CAD operations and model conversions, while engineering simulations have to be hard-coded which makes employment of for example injection flow-molding, finite elements analysis or fluid dynamics simulations difficult to implement and adapt. Similarly, middleware to run shared AR like Muddleware [26] focus on multi-user game playing and level of detail management and do not support interactive visualization and adaptation of objects.

Several systems and architectures have been devised to support design activities by advanced 3D graphics interfaces, based on completely virtual paradigms. Although some of these are readily available like Open Cascade<sup>1</sup>, they will not be easily adopted by existing design Studios, who are committed to use a specific commercial CAD package that offers distinct features and a familiar user interface.

<sup>1</sup> [www.opencascade.org](http://www.opencascade.org)

## 2 IAP HARDWARE

To establish augmented reality for design, a growing selection of output, input, and physical prototyping has to be considered. A treatise of these enabling hardware technologies was published in [21]; as output means, our first preference is the projector-based display. On input and physical model making, a wide variety of options is available, none of which provided a complete solution. Based on the situation at hand, a final selection will have to be made.

### 2.1 Hardware Requirements

In considering current design practice and future support scenarios, we identified the following requirements regarding IAP hardware:

- **Mobility:** design reviews often take place at the location of client or at other stakeholders; the IAP apparatus should fit in two reasonably sized suitcases and should withstand unsupervised transport (by air, in trunk of a car).
- **Installment:** during the execution of the scenarios presented at Section 1.2, installment of time and effort should be kept at a minimum (max 15 minutes, with self-starting facilities). If calibration is required, the system should provide step-by-step guidance. We acknowledge the constraint uttered by [26] that such devices should be self-contained units with no loose parts, which should auto-start if a failure occurs.
- **Fixation:** position and orientation of the projector systems should be fixed without creating hazardous or erroneous situations. In the case that a projector is a hand-held system, it should have a facility to stay in a particular posture while the user can release it.
- **Portability:** during use, the projector systems could be moved and this should be doable by one person. To enable this, the amount of cables should be kept at a minimum.
- **Time performance:** as the IAP concerns both model inspection as model adaptation and simulation, the update frequency is important to keep an interactive 3D experience. The complete system must run at least at 10 Hz, with little lag time as possible
- **Accuracy performance:** constrains on tracking accuracy and projector resolution are situation

dependent. This also depends on the scale of the physical model and the level of detail of the projected information. This issue requires revisiting during evaluation.

- **Environment interaction:** IAP systems should not adversely influence general environmental conditions, in particular regarding noise and lighting condition. As projectors typically contain fans, noise should be minimized in order to support regular conversations (max 30 decibels, the level of whispering). Furthermore, IAP systems might require a dimmed room, but it should still be possible to see and interact with other persons and objects in the environment. A minimum level of 200 LUX is allowed (a dimmed training room).
- **Projector performance:** a lot of variation exists in projector specifications, like resolution, zoom range, field of depth and light intensity. However, little can be constrained regarding these characteristics, as the application in depends strongly on for example distance to the object.

### 2.2 System Framework: the I/O Pad

As a fundament for the hardware platform, we would like to adopt the paradigm of the I/O bulb as presented by Underkoffler and Ishii [19]. The I/O bulb (Input-Output bulb) views the input (camera or other sensors) and output (projector) as one single unit. This bulb can be switched on and off at will, can be configured in groups and so on. For example, each I/O bulb could perform a particular task: 3D modeling, simulation analysis or annotation management. In this fashion, dedicated projector modules can be viewed as a

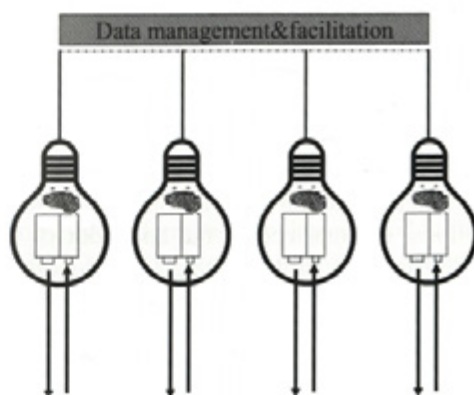


Fig. 1. Networked I/O pads

physically addressable (i.e. tangible) component. As demonstrated by the so-called procams community<sup>2</sup> (projector-camera systems), many algorithms and applications have evolved that can be employed in this setup including calibration of colors temperature, 3D scanning, and visual echo-canceling.

We extend the I/O bulb concept, by including processing power and a touch screen interface, the result of which we would like to label as I/O Pad. As its name suggests, it is supposed to fit within the series of tangible user interfaces devised to blend physical and virtual realms like I/O bulb and I/O brush [16]. The I/O Pad is a self-sufficient, untethered device (if battery operated). Collaboration of multiple pads is facilitated through wired or wireless network connections. In essence, our concept overlaps with iLamps [15]; both add handheld, geometrically aware projection and allow ad-hoc clustered projections. However, the I/O Pad differs in three ways: i) each I/O Pad contains a touch screen to interactively capture and display sketches and gestures from designers, ii) each pad is equipped with recording devices (webcam) to pick up discussions and usability assessment sessions, and iii) the I/O Pad network architecture encompasses a distributed structure to facilitate data sharing, dialogue management and in particular session recording. As shown in Figure 1, different

instantiations of the I/O Pads might be used concurrently. To support particular activities, some pads might be switched on or off or moved according to the user's wishes.

I/O Pads can be small and portable, or they carry increased projection and computing power. Two extreme versions can be conceptualized, as specified in Table 1. For the handheld system, a small, LED-based projector seems the most appropriate; these can run on batteries and are almost silent. As a processing unit, an Ultra-Mobile PC (UMPC) is a good candidate; it contains a touch screen and in fact is a miniaturized PC that runs standard windows or Linux software. Due to the lack of computing power, a lightweight 3D tracking system should be selected, for example ARToolkit. This is an open source library for optical 3D tracking and tag identification that employs flat rectangular markers [9]; our field tests suggest it will perform well on the UMPC platform (approximately 20 Hz, 640x480 camera resolution).

The larger I/O Pad is equipped with more powerful constituents, to offer improved projection, processing, and 3D tracking. Recent video projectors offer XGA or higher resolutions and produce over 3000 Lumen. As processing and interaction unit, we propose the employment of a high-end Tablet-PC with a touch screen option. Such Tablets harbor both active and passive touch technologies and can be operated by fingers and special pens. In the latter case, the tablet is pressure sensitive, which supports the natural expressiveness of designer's sketching abilities. To support 3D tracking and user events, this system can be equipped with an infrared camera and infrared lamp, as being found in typical motion capture systems like Motion Analysis and Vicon. By deploying retro-reflective passive markers in combination with active, LED-based tags, both fine-grained 3D component tracking and user interaction with physical components (by for example phidgets) will be supported. This I/O pad is meant to offer global lighting of a design/environment, from a larger distance. Due to its weight, proper fixture like a professional tripod is essential.

### 2.3 I/O Pad Implementation

Based on the hardware specifications described above, two I/O Pads have been

Table 1. Characteristics of two Smart I/O Pads

	Handheld I/O Pad	Large I/O Pad
Projector	LED-based, battery operated	Silent standard video projector.
Projector power	30 Lumen	3000 Lumen
Projected resolution	800x600 pixels	1280x768 pixels
Working distance from object	10-50 cm	100-300 cm
Processing unit	UMPC	Tablet PC
Touchscreen diameter	5-7 Inches	12-15 Inches
3D tracking	Marker-based (ARToolkit)	Active/passive infrared tracking (motion capture system, camera includes infrared canon)
Estimate total weight	1 kg	2.5-3 kg

<sup>2</sup> procams workshop: [www.procams.org](http://www.procams.org)



Fig. 2. Pilot versions of I/O Pads: handheld (top) and large (bottom) I/O Pads

developed, as depicted in Figure 2. The smallest version contains a LED-Based projector from Toshiba (FF-1), which weighs 750 grams including battery. This projector is connected to an Asus R2H UMPC (900 MHz ULV Celeron processor, 500 MB RAM), which has a 7-inch passive touch screen. A Microsoft NX-6000 web camera delivers up to 2 Mega pixel resolution video images. This package weighs approximately 1.5 kilograms.

The larger I/O Pad is based on a standard video projector (Epson EMP-811) that has 2000 lumen and is capable to project at XGA resolution (1024x768 pixels). A Tablet PC delivers processing and a passive touch screen (HP TX1000, AMD dual-core TL50 processor, 1GB RAM, 12.1-inch screen). For Infrared motion capturing the system currently employs a Wii remote controller (also known as WiiMote), which is able to track 4 Infrared light sources simultaneously at a resolution of 1024x768 pixels at 100 Hz. This WiiMote game controller is connected wirelessly to the I/O pad by Bluetooth. A converted pen containing an infrared LED at its tip, plus a battery, and a switch serves as lightpen which can be tracked by the WiiMote in two dimensions; contact of the pen tip

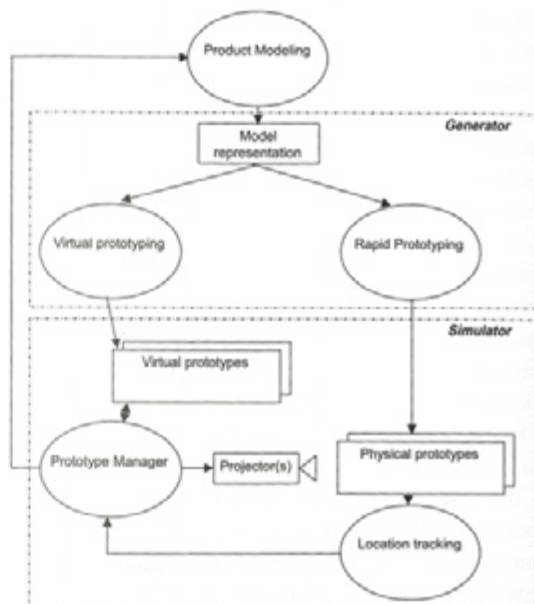


Fig. 3. Workflow of the WARP system

with the object surface can be reconstructed to a 3D point as the physical surfaces are known, given the exact location and orientation of the controller. This complete bundle weighs approximately 2.5 kg and requires a professional tripod to aim toward the area of interest.

The user interaction is performed by operating the touch screens of the I/O pad, drawing on physical objects surface with the light pen, or moving the model and I/O Pad.

### 3 IAP SOFTWARE

As stated in section 1.2, a growing collection of Augmented Reality software exists; yet the solutions are not focused towards design support and require extensive adaptation and configuration. In order to establish IAP, algorithms are necessary to connect input and output, and to allow a variety of modeling and simulation applications. In [20], we have devised the WARP framework to support the complete prototyping pipeline, shown in Figure 3. It encompasses both manufacturing (Generator) and usage (Simulator) of IAP. In this publication we elaborate on the second module. Based on the case studies, a number of functional requirements have been formulated and we will then propose elements of the resulting software architecture.

### 3.1 IAP Functional Requirements

In devising a software architecture for IAP, the following requirements were identified:

- Operation of IAP should be compatible with the user interface and conceptual modeling/simulation the designer is familiar with.
- A wide range of options to calibrate the I/O pads will be offered (coordinate systems, color, optical distortions).
- The architecture should be open for (future) 3D tracking and event sensing methods.
- The system should auto-start and should offer a number of preset configurations that fit the scenarios.
- The IAP can relate various physical components to virtual counterparts. The user should be able to attach and detach these in an easy fashion.
- IAP will also recognize certain physical behavior as actions (gestures, button presses etc.), which can be connected to various functions.
- The architecture will support recording all input events and the corresponding 3D models. Different levels of granularity might be selected to optimize recording performance (time, level of detail, channels).

### 3.2 WARP 2.0 Architecture

The resulting WARP 2.0 architecture is shown in Figure 4. In the center, the IAP Session Manager is shown. It is responsible for setting up sessions at one or more I/O Pad. This includes model sharing, session recording, and configuration management. As stated in the final requirement, the recording function will combine the modeling history with discussion by recording video and audio as well.

On the right, the set of input and output devices are shown. Processing of input signals and 3D tracking is performed by the a Tracker subsystem, which supports an arbitrary number of commercial and research position sensing devices. It is based on (networked) data streams. The data flow based paradigm also enables easy recording of movement and configurations by storing the streams to persistent memory.

Key ingredient in the IAP architecture is a third-party 3D modeler or simulation package, depicted on the left. Instead of creating our

proprietary visualization solution, we want to exploit the fact that most designers already have some type of 3D modeling package like Catia, Solidworks or Rhinoceros. Most of these are capable render the virtual components in real-time, adjusted for the projector by means of configuring and maintaining a virtual camera. Furthermore, most modeling packages can be extended by scripting, macros or other automation mechanisms (like ActiveX). For supporting IAP, we have defined four plug-ins that need to be implemented for a particular package: i) Configurator, ii) 3D Viewer, iii) TUI Management, and iv) Watcher. The responsibilities of these are discussed below.

The Configurator plug-in can be viewed as the local liaison of the IAP session manager– it is responsible for the local setup and execution of other plug-ins, loading/saving models and sharing this with other IAP instances. Furthermore, it offers an auto-start function and a GUI to arrange the IAP in line with the defined application scenarios and the related functions.

The 3D Viewer is responsible to define and update a virtual camera that copies the internal and external parameters of the attached projector. Internal parameters include field of view, aspect ratio and projection center; external parameters correspond to translation/rotation of the projector and the scale of the virtual and real-world coordinate systems. In terms of 3D computer graphics concepts, these are being specified in two transformation matrices: a projection and model matrix [5]. In some cases, these transforms need to be mapped to different units for the CAD package (e.g. CATIA requires focal point instead of field of view). When the projector is moved, the virtual camera will update the model transform accordingly, based on the input from the IAP session manager. Ideally, the 3D viewer plug in should sense alterations in projector zoom (focal point) and adjust the projection matrix accordingly. Furthermore, the 3D Viewer module is responsible for determining the appropriate field-of-depth and should be capable to adjust the focus of the projector when required (based on distance between the projector and objects in virtual space).

The motion capture system will track individual physical elements including identification, position and possibly state (e.g. button press). The TUI-management plug-in is in charge of mapping these actions to the

corresponding virtual components in the modeling or simulation package. This might effect in showing/hiding and translating/rotating objects but also steering additional virtual simulation modules (like physics behavior or screen navigation).

The Watcher plug-in is responsible to support the recording functions, which can be either saved to file or streamed to a centralized session recorder through a network connection. This plug-in offers a number of services, including capturing either screenshots, full 3D models per stage, or hybrid version of both based the modeling events the hybrid option could for example encompass capturing full 3D models after alterations of the model, and screenshots during model viewing. Furthermore, the update frequency can be set in time or event-based triggers.

Specific interfaces are being determined at this moment. Storage of data at the IAP Session Manager will based on tuplespaces (also known as a blackboard). This type of associative memory is highly flexible and supports various mechanisms to share data with clients. In particular, the IAP Session Manager uses a publish-and-subscribe protocol in order to propagate events and data streams to those modules that have shown interest. Furthermore, the IAP Session Manager will offer a calibration routine to determine the projection matrix and to determine the (fixed) transformation between 3D tracking and projector positions.

### 3.3 Plug-in Factory Concept

Although the plug-in architecture offers a lot of flexibility, it yields challenges towards implementation and maintenance of these plug-ins. First, all modeling applications have their own automation solution which requires different (dialects of) scripting languages; for example Visual Basic for Applications versus C++. Second, each application offers a different set of operations and data structures, which evolve at each – typically annual – update. Configuration and version management needs to be addressed in the WARP architecture.

A solution to this problem can be found at the Abstract Factory design pattern [6], as depicted in Figure 5 for the 3Dviewer and Configurator classes. Abstract classes for each plug-in type define its public interface and contain the basic functions which can be shared among instantiations; for each dialect, the plug-ins are subclassed to adapt for the particular version for example 3Dviewer\_Solidworks. Second, a Factory class for each of the configurations is defined, based on the AbstractPluginFactory class. These factory classes are responsible to instantiate the actual plug-ins by means of Create() function calls. These instantiation calls will use the publicly available GNU make application<sup>3</sup> to support file

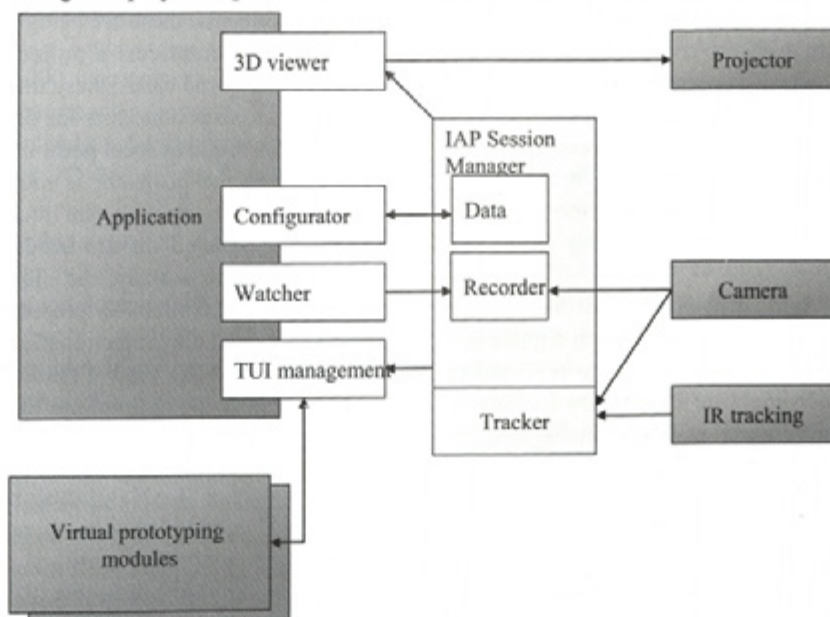


Fig. 4. WARP 2.0 Software Architecture for Interactive Augmented Prototyping

<sup>3</sup> <http://www.gnu.org/software/make>

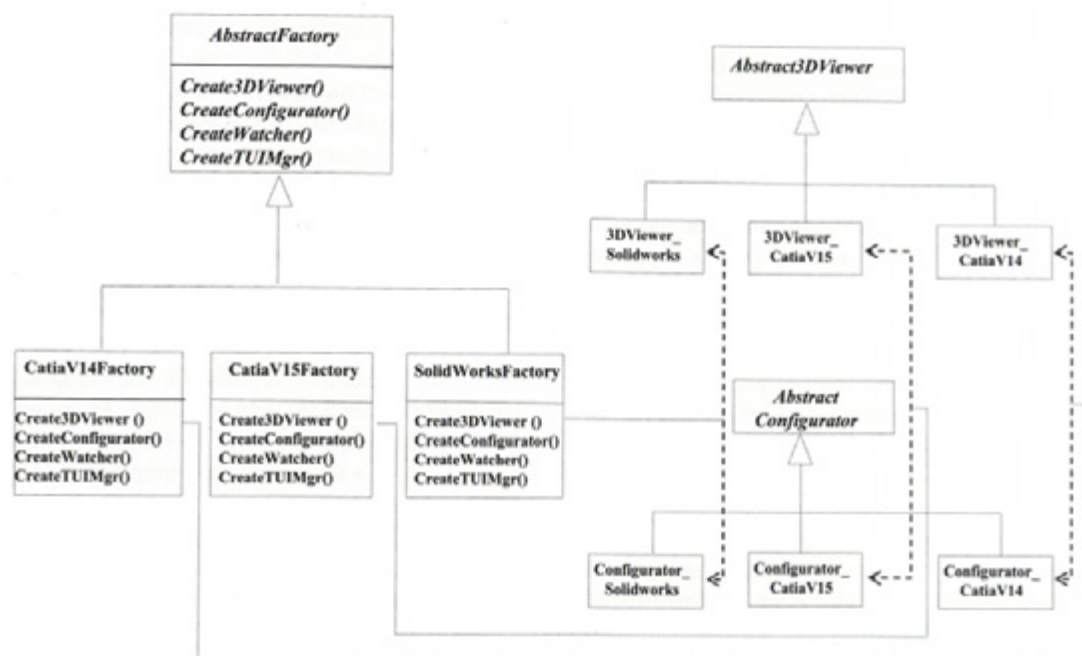


Fig. 5. UML diagram of Plug-in factories (illustrated are only 3Dviewer and Configurator plugins)

deployment, shell scripting and compilation/linking in different languages.

From personal experience, we found several ways to hide and show components in Catia; the most straightforward implementation worked, but resulted in poor performance (consuming 200 milliseconds for one cycle and extensive flashing of component wireframes). After putting some efforts in optimizing this operation, we switched to a less elegant but better working strategy by simply translating objects outside/inside camera reach. Selecting and implementing strategies requires tuning and such principal solutions should be encapsulated in the Factory classes, in order to share these basic functions among all plug-ins for that particular dialect.

### 3.4 Deployment and Calibration

In the case of using a single I/O Pad, all processes outlined before can run on the same machine. In order to enable concurrent use of multiple I/O Pads, a networked system layout is necessary. An overview of a typical setup is shown in Table 2. As a main communications solution, we have selected OpenSound Control protocol [27], which is currently supported by emerging Tangible User Interface APIs. For each projector unit, a

single application instance should run with its corresponding plug-ins. The same holds true for the Tracker stack, although these can be combined

Table 2. Software deployment if using Smart I/O Pads

	Handheld I/O Pad	Large I/O Pad
Application	Same on both, e.g. SolidWorks or Catia	
OpenTracker input	Webcam (AR patterns)	Motion capture system
Data streams to recorder	Webcam, projector position, user interaction with Pad	3D positions of components
TUI Management	Limited capability to run simulations	Optional
IAP session manager	Preferred at Handheld.	Only when computationally too heavy at handheld.
Calibration required		
Transformation camera->projector coordinates	Yes (fixed)	Yes (fixed)
Projector parameters	Yes (fixed)	Yes (variable)
Camera internal parameters	Yes (fixed)	No
Camera position/orientation	No	Yes (variable)



through the data flow definitions. It seems logical to employ a networked file system on all I/O Pad systems, which is hosted by the same machine that runs the IAP Session Manager (of which only one instance should persist). This can be one of the I/O Pads or a separate server. Another dependency at this point is the fact that multiple physical component and event tracking is only supported by the larger I/O Pad, through the motion capture equipment. Tracking such cues needs at least 2 IR cameras of which one is integrated in the large pad.

Calibration for the I/O pads deals with a number of elements, most notably, the transformation between camera and projector world coordinate systems, defined by the projector internal parameters (field of view, center position), the camera internal parameters, and the vector (translation/angle) between the two components. As both types of I/O Pads are equipped with different hardware, both require separate sorts of calibration, as summarized in the lower part of Table 2. The handheld unit typically requires one single calibration step, which can happen after assembly of the unit. The large I/O pad, however, requires calibration before each installation due to the registration of the IR cameras of the motion capture system and fluctuations in both projector focal distance (zoom) and color temperature.

## 4 FIRST APPLICATION AND DISCUSSION

### 4.1 Application and Findings

The I/O Pad configurations specified in Section 2.3 were constructed, with a specific focus on the handheld system. An impression of this system in use is given in Figure 6. Four standard ARToolkit markers were placed around a simple object (a cup resembling a pyramid with its top cut off). The geometry was modeled in Catia and decorated by texture maps. The only interaction

supported application was to combine the digital and physical models. At present, the fundamentals of the WARP 2.0 software architecture have been developed, much effort was necessary to devise and implement calibration routines. The calibration procedure was developed and tuned for the simple object – the projection (u,v coordinates) of each of the 8 vertices (x,y,z coordinates) has to be indicated at the touch screen, the camera calibration algorithm specified in Faugeras [4, Chapter 1] is used to compute the projection matrix.

In this simple application, the computing power of the small PC was sufficient. Although it is a bit bulky and heavy to lift for longer periods of time, the system was easy to transport and to move while running the system.

In the empirical case studies, different applications were used: CATIA (automotive), Vectorworks (interior), and Unigraphics (information appliances). For reasons of availability, the minimal collection of target applications for WARP 2.0 has been set to CATIA and Solidworks. Although this is less compatible with the interior design domain, these still offer similar modeling functions while they are based on completely different representation and automation mechanisms.

### 4.2 Discussion

Software-related issues

When multiple I/O Pads are used simultaneously, synchronization between the running CAD or simulation packages is necessary. This can be solved in several ways, for example by application sharing (i.e. running the same synchronized instance on all pads), by model sharing among different applications (for example one pad dedicated to modeling and one to simulation) or by hosting diverse models of the same product. More investigations are necessary



Fig. 6. Handheld I/O Pad in use

to determine which option is the most appropriate. In order to record sessions, the required bandwidth to capture all video and application states can be large. As with traditional capturing systems, a tradeoff has to be made between quality and size/bandwidth; this is even more in the case of using multiple I/O Pads concurrently. One probable option is to store data locally in real-time and to upload/share this after use.

#### *Projector-related issues*

In terms of luminosity, the projector is useful when it produces 50 to 100% more light at the surface than its environment<sup>4</sup>; this would equal 300 Lux at the given surface light of 200 Lux as presented in Section 2.1. Here, we have to take into account that the reflectance of the (white) object is approximately 80%.

The small projector (Toshiba FF-1) has a measured power of 16 Lumen<sup>5</sup>; as Lux equals Lumen per square meter, the conversion has to take into account the maximum envelope of projection at a certain distance from the object. For the furthest distance of 40 cm, the projected area is 23.1 by 17.3 centimeters or 0.04 m<sup>2</sup>; this results in approximately 400 Lux light energy at the object's surface, or 320 Lux when corrected for reflectance; this is sufficient. In the case of the minimal distance of 20 cm., the area was 12 x 9 cm (0.01 m<sup>2</sup>), which results in 1185 Lux adjusted for 80% reflectance. In the case of the larger projector (Epson EMP 811, 2000 Lumen), in the wide-angle setting on a 1-meter distance, the projected area is 67 x 50 centimeters (0.33 m<sup>2</sup>); this yields approximately 4800 Lux corrected for reflection. In the furthest specified distance of 3 meters, the area is 1.99 x 1.47 meters (2.93 m<sup>2</sup>), resulting in 547 Lux. This means that effectively in all cases the projector brightness is sufficient.

Field of depth – in experimental studies, we found little issues in using a single, fixed focus on the large projector in the range specified (1-3 meters). For the small projector, the focus remained acceptable in the following ranges: i) between 20 and 30 cm, ii) between 30-50 cm. Its focus ring is not easy to operate, at least not when the projector is assembled in the I/O pad. Instead, we could imagine a manual of automatic switch between these two ranges. The automatic option seems viable, as the distance between projector and object is constantly measured.

Noise – for cooling purposes, projectors are equipped with a fan that generates noise. For the small projector this is negligible, but for the larger this is an issue. New “whisper” video projectors are currently being marketed which have better characteristics, but are still not completely silent (yielding approx 28 Decibels).

#### **4.3 Customization of Functions**

As mentioned in section 1.1, the basis interpreting the case studies was the framework of Critical Systems Thinking [8]. Its philosophy of Total Systems Intervention (TSI) will be used in customizing the IAP system towards a particular design situation. TSI has three phases: creativity, choice, and implementation. The creativity phase focuses on selecting a number of perspectives to assess the situation and to identify concerns and problems. During the second phase, a suitable intervention methodology is selected to deal with the problem at hand, originally based on the collection of methods of organizational sciences (covering functionalist, interpretive, emancipatory, and postmodern theories); the selected intervention is then implemented in the subsequent phase.

In a similar fashion, the use of IAP should be preceded by a similar assessment and selection. The scenarios of IAP have been characterized to cover one or more of the four theories mentioned above, an overview is given in the Appendix. This ensures coverage of a number of concerns and issues during design when either product shape or product behavior plays a role. The application of TSI in selecting IAP functions can be shaped in several ways, for example as a wizard dialogue, as a flowchart or map or encapsulated in templates.

#### **5 CONCLUSION**

Although the concept of interactive augmented prototyping allows possible benefits to design processes, there is a large threshold in employing this technique. In this paper, we have proposed a combination of hardware and software solutions. Based on empirical findings from three different design processes, functional requirements and usage scenarios were specified.

This paper introduced a hardware solution called I/O Pad. Equipped with a video projector, optical 3D tracking, and a tablet PC, the system

<sup>4</sup> <http://www.dvimg.com.au/iti-f1.html>

<sup>5</sup> <http://www.pcmag.com/article2/0,2704,2099318,00.asp>

represents a fully equipped IAP system. Two versions were presented, a larger, more powerful and a smaller, more mobile I/O Pad. Multiple I/O Pads can be used concurrently. An initial implementation has been developed, using a LED-based projector and a UMPC. The first application of the I/O Pad proves the wireless versatility of the hardware platform.

The software architecture called WARP 2.0 was proposed based on a plug-in architecture to empower existing 3D modeling and simulation applications and thus be compatible with existing design practice. The architecture was designed to connect several 3D tracking and sensor devices through a centralized IAP session manager to an arbitrary number of I/O Pads. The proposed software architecture allows the designer to work in a familiar modeling environment yet includes powerful concepts from tangible user interfaces to support several types of interaction with physical components. Secondly, the application of the Abstract Factory design pattern solves the configuration and version management of the plug-ins.

Technical issues involve application sharing and projector characteristics. As development is still early, it has to be determined to what degree the usability of the system and the application IAP influences the design process at hand. The resulting I/O Pads will be tested in a series of field experiments, in varying domains of industrial design.

## 6 REFERENCES

- [1] Becker, M., Bleser, G., Pagani, A., Stricker, D., Wuest H. An architecture for prototyping and application development of visual tracking systems. *Int. Conf. on 3DTV*, 2007.
- [2] Bimber, O., Stork, A., Branco, P. Projection-based augmented engineering. *Proceedings of International Conference on Human-Computer Interaction (HCI'2001)*, vol. 1, p. 787-791.
- [3] Bimber, O., Raskar, R. Spatial augmented reality: merging real and virtual worlds. A. K. Peters, Ltd., 2005.
- [4] Faugeras, O. Three-dimensional computer vision: a geometric viewpoint. MIT press, 1993.
- [5] Foley, J., van Dam, A., Feiner, S., Hughes, J. *Computer graphics: principles and practice*. 2nd Ed. in C. Reading: Addison-Wesley, 1995.
- [6] Gamma, E., Helm, R. Johnson, R., Vlissides, J. Design patterns, elements of reusable object-oriented software. Reading: Addison-Wesley, 1995.
- [7] Grasset, R., Boissieux, L., Gascuel, J.-D., Schmalstieg, D. Interactive mediated reality. *Proceedings of AUIIC2005*, 2005, p. 21-29.
- [8] Jackson, M. Systems approaches to management. New York: Kluwer/Plenum, 2000, ISBN 0306465000.
- [9] Kato, H., Billinghurst, M. Marker tracking and HMD calibration for a video-based augmented reality conferencing system. *Proceedings of International Workshop on Augmented Reality (IWAR 99)*, 1999, p. 85-94.
- [10] Kretschmer, U., Coors, V., Spierling, U., Grasbon, D., Schneider, K., Rojas, I., Malaka, R. Meeting the spirit of history. *Proceedings Conference on Virtual Reality, Archeology, and Cultural Heritage VAST '01*, 2001, p. 141-152.
- [11] Ledermann, F., Barakonyi, I., Schmalstieg, D. *Abstraction and implementation strategies for augmented reality authoring*. Emerging Technologies of Augmented Reality: Interfaces and Design, Haller, Billinghurst&Thomas (eds), 2006, p. 138-159.
- [12] MacIntyre, B., Gandy, M., Dow, S., Bolter, J. D. DART: a toolkit for rapid design exploration of augmented reality experiences. *Proceedings of UIST '04*, 2004, p. 197-206.
- [13] Milgram, P., Kishino, F. A Taxonomy of mixed reality visual displays. *IECE Trans. on Information and Systems (Special Issue on Networked Reality)*, vol. E77-D, 1994, no. 12, p.1321-1329.
- [14] Nam, T-J, Lee, W. Integrating hardware and software: augmented reality based prototyping method for digital products. *Proceedings of CHI'03*, 2003, p. 956-957.
- [15] Raskar, R., van Baar, J., Beardsley, P., Willwacher, T., Rao, S., Forlines, C. iLamps: geometrically aware and selfconfiguring projectors. *ACM Trans. Graph. (SIGGRAPH)* 22(3), 2003, p. 809- 818.
- [16] Ryokai, K., Marti, S., Ishii, H. I/O brush: drawing with everyday objects as ink. *Proceedings of Conference on Human Factors in Computing Systems (CHI '04)*, 2004, p. 303-310.
- [17] Schmalstieg, D., Fuhrmann, A., Hesina, G., Szalavari, Zs., Encarnacao, L.M., Gervautz,

- M. Purgathofer, W. The studierstube augmented reality project. *Presence – Teleoperators and Virtual Environments*, vol. 11(1), 2002, p.33-54.
- [18] Tamberend, H. Avocado: a distributed virtual reality framework. *Proceedings of IEEE Virtual Reality 1999*, p. 14-21.
- [19] Underkoffler, J., Ishii, H. Urp: a luminous-tangible workbench for urban planning and design. *Proceedings of the CHI'99 conference*, 1999, p. 386-393.
- [20] Verlinden, J.C., de Smit, A., Peeters, A.W.J., van Gelderen, M.H. Development of a flexible augmented prototyping system. *Journal of WSCG*, vol. 11(3), 2003, p. 496-503.
- [21] Verlinden, J., Horvath, I., Framework for testing and validating interactive augmented prototyping as a design means in industrial practice. *Proceedings of Virtual Concept 2006*.
- [22] Verlinden, J., Horvath, I., Edelenbos, E. Treatise of technologies for interactive augmented prototyping. *Proc. of Tools and Methods of Competitive Engineering*, 2006, p. 523-536.
- [23] Verlinden, J., Nam, T-J, Aoyama, H., Kanai, S. Possibility of applying virtual reality and mixed reality to the human centered design and prototyping for information appliances. *Research in Interactive Design*, vol. 2, 2006.
- [24] Verlinden, J., Horvath, I. A critical systems position on augmented prototyping systems for industrial design. *Proceedings of ASME-CIE 2007*, 2007, DETC2007-35642.
- [25] Wagner, D., Schmalstieg, D. Handheld augmented reality displays. *Proceedings of Virtual Reality Conference 2006*, p. 321- 322.
- [26] Wagner, D., Schmalstieg, D. Muddleware for prototyping mixed reality multiuser games. *Proceedings of Virtual Reality Conference, 2007*, p. 235-238
- [27] Wright, M., Freed, A., Momemi, A. OpenSound control: state of the art 2003. *Proceedings of 2003 Conf. on New Interfaces for Musical Expression (NIME '03)*, p. 153-160.
- [28] Zauner, J., Haller, M., Brandl, A. Authoring of mixed reality assembly instructor for hierarchical structures. *Proceedings of ISMAR '03*, 2003, p. 237-246.

## APPENDIX: FUNCTIONS DERIVED FROM CASE STUDIES

Scenario	Function	Originating Perspective Domain*		
User studies	To simulate usage (augmenting interaction on physical mockup)	IA	F(Functionalist)	
	To record use and user reactions (keystrokes and performance, (non)verbal communication of users)	IA	E(Emancipatory)	
	As a conversational piece – projecting/capturing contexts and challenge the user	IA	E,P(Postmodern)	
Exploration	Combining (manual) modeling physical shape with interaction design	IA	F,P	
	Inspire by projecting alternative component layouts (also of older and competing products)	IA, AD, ID	F,P	
	Combining existing physical model (chassis, engineering package) with virtual surface modeling	AD	F	
	Browsing through a selection of physical components and include some of these a virtual global concept	AD	F,P	
	Freehand sketching on physical surface	AD	I(Interpretive),P	
	Browsing through a collection of physical models and explore their placement in a global concept, ability to record/bookmark alternatives	ID	P	
	Project/adjust pedestrian flows interactively	ID	F,I,P	
	Facilities to add references to style elements in several information carriers, including verbal, textual, symbolic, properties, and so on.	ID	I,E	
	Freehand sketching on physical surface, integration with modeling	ID	F,E	
	Combining existing physical models with textures/materials exploration	ID	F	
	Design review	Internal discussion of design alternatives, capturing interaction and reflections (annotation)	IA	F,E
		Freehand sketching on surface (captured with author +timestamp for later use)	IA	E
		Present user studies: usage feedback, co-located events and subjective evaluations	IA	I
Presentation of design alternatives, capturing interaction and reflections and possibly design decisions (annotation)		AD, ID	I,E	
Presentations of design exploration scenarios to support reasoning and try to convince client		AD	I,E	
Archiving and retrieving reviews (replay, overviews etc), allowing shared access		AD	I,E	
Ability to prepare the model for discussions, by fixing/filtering items and by setting a small number of configurations		ID	I,E,P	
Interactive display of colors/materials in focused areas only (similar to colored doll in existing model)		ID	I,P	
Combine physical model as an indexing tool for design details		ID	F,I	
To present usage scenarios (pedestrian flows)		ID	I	
Archival and retrieval of design reviews (replay, overviews etc), to be shared through network		ID	I,E	
Abilities to add coarse budgeting and design requirements tools with interior design		ID	F,I	
Presentations to customers or higher management		Present project status: design (alternatives), disciplines (design: industrial, interaction, engineering: electrical, mechanical, manufacturing)	IA	F,I
	Present a summary of most interesting user feedback	IA	I,E	
	Present a variety of designs as a portfolio overview, either interactive or self running	AD	E	
	Present one particular product in its context and its specific (animated) features, kiosk mode	AD	F,E,P	

\*Domains: IA= Information Appliances, AD=Automotive Design, ID=Interior Design.