Universität Bielefeld

Master's Thesis

Design and Implementation of Multi-modal AR-based Interaction for Cooperative Planning Tasks

Author:
Alexander Neumann,
alneumann@gmx.net

Supervisors:
Dr. Thomas Hermann,
Dr. Karola Pitsch

May 16, 2011

Master's Thesis

Design and Implementation of Multi-modal AR-based Interaction for Cooperative Planning Tasks

Alexander Neumann

Submitted to the Faculty of Technology
of Bielefeld University

Abstract

Cooperative task solving requires complex coordination and interaction patterns which are generated and perceived by humans intuitively and unobtrusively. To investigate these patterns and to research opportunities to support cooperative task solving itself we designed and implemented an augmented reality framework. Seven augmentation prototypes were designed based on elicited requirements from which four were implemented to test the expandability of the framework. Insights gained during the development process were used to propose opportunities to improve the user experience of augmented reality based applications.

May 16, 2011

Contents

1. Introduction

Cooperative task solving requires complex coordination and interaction patterns which are generated and perceived by humans intuitively and unobtrusively. Gestures, spoken language, gaze direction and other signals are used to communicate one's internal state as well as direct the interlocutor's activities. These and other – not yet completely understood – factors emerge from very efficient processes of communication known as alignment. While aligned, interaction partners can communicate more effectively with a lower risks of misunderstanding. It is speculated that the underlying processes of alignment are vital factors for the success of human-human interaction

Today's environments for cooperation are more and more influenced by technology, which gives rise to three research questions:

- How does interaction technology interfere with alignment processes in human-human communication,
- how can technology be used to better understand alignment, and
- how can technology be used to support and facilitate alignment processes?

This thesis addresses the possibilities offered by augmented reality (AR) for alignment research but before going into details let us have a look at AR itself.

During the last decade augmented reality found its way into everyday life devices. Thanks to progress in mobile phone development, so-called smart phones offer powerful hardware and features capable of supporting AR related applications. Spatial Augmented Reality (SAR) gains more and more attention in art and is used in a variety of light art installations. Head-Mounted Display (HMD) related AR, one of the most traditional branches of AR research has largely not found its way into public cognizance though. Work assistance in assembly or surgery for example, or information assistance in tourism or orientation contexts are just a few promises connected to that branch. However, AR and other modern human-computer interaction (HCI) methods not just offer improved user experiences, but also the possibility to investigate human-human interaction itself.

Considering Head-Mounted Displays (HMDs) as a gateway for visual input, the AR-based Interception Interface (ARbInI) was developed and is used for alignment research in the C5 – Alignment in Augmented Reality project today [DBH+09] within the DFG Collaborative Research Centre (CRC) 673. Communication channels like head movement, gaze direction, fields of view and gestures are monitored and logged for analysis with respect to alignment. ARbInI also offers the possibility to *intercept* and alter inputs for subjects independently so that misalignment can be induced to observe alignment strategies and repair attempts. Some basic studies to investigate the possibilities of supporting alignment with the help of AR were also conducted.

The problems which have become apparent with the previously used AR-system are

- **Impact of hardware on human-human interaction**
  Heavy devices and issues caused by HMD related loss of eye contact were identified as drawbacks to alignment recording.

- **Impact of software/system stability on human-human interaction**
  Previous mentioned interferences were increased by tracking stability and latency issues.

Furthermore it seems necessary to extend the AR functionality for easier development of augmentation strategies.

The Lightweight Augmented-Reality Facility with Open Real-world-based Graphical Enhancement (LAFORGE) is a part of ARbInI and it is used for capturing video signals recorded by the cameras of the HMDs, adding generated content and feeding the result to the displays of the HMDs. To tackle software related interaction interferences as well as to provide new possibilities for augmentation prototyping, LAFORGE was analyzed during this work. To fulfill research-related requirements such as study authoring and post-study data handling as well as integration of new AR features to increase user acceptance, a variety of perspectives were taken in account. Interaction design analysis methods and software development procedures were used to redesign LAFORGE from scratch and create a flexible tool set.

The result of this work is LAFORGE 3, a freshly written third version of LAFORGE, designed and implemented in this thesis which supports prototyping of visualization methods, interchangeability of tracking methods, scene and rendering handling and distributed problem solving. The introduced Object-oriented Graphics Rendering Engine (Ogre) extends the set of available visual augmentation tools. Additionally, methods for the combination of multiple tracking devices were designed and implemented in this work to increase the system's stability and giving the opportunity to augment according to tracking data even though local data are not available. These features will provide the basis for further development in the second funding phase of C5 to improve user experience and ease analysis to gather new insights in alignment research.

This thesis follows and builds upon Christian Mertes' diploma thesis *Multimodal Augmented Reality to Enhance Human Communication* [Mer08]. Readers familiar with his work, may quickly feel familiar with this thesis since context and used technology have not changed. However, it is not required to have read that thesis in advance though. But it is recommended reading for a more detailed introduction to communication channels and multimodality.

In the next chapter related work and basic concepts are introduced. This includes a review of the current state of key technology used in this thesis. Chapter 3 describes the requirement elicitation process and design prototypes created based on analysis results. System related requirements are discussed afterward in Chapter 4 where we introduce the system's architecture and its implementation. Results of test runs conducted including comparisons between the new software LAFORGE 3 and the old LAFORGE software can be found in Chapter 5. In the outlook the findings from the evaluation are reviewed to derive future development possibilities as a roadmap for software development. A summary is given in Chapter 7.

2. Related Work

To get in touch with problems faced when developing HMD related AR systems we will introduce these technologies first. In the following, Section 2.3 summarizes the concept of alignment and its relation to A project within the CRC 673 Alignment in Communication at Bielefeld University. Its goal is to investigate benefits of augmented reality for alignment research and to develop AR-based methods to support alignment. The basic concepts of Interaction Design (IxD) will be introduced afterwards. This includes the IxD focus regarding this work as well as the generic necessity of IxD in the development process. The introduction of *Phaenomeum* – the usage context of the system developed – closes the chapter.

2.1. Head Mounted Displays

Figure 2.1.: Sensics zSight HMD[1]. The zSight provides 70 degree field of view, an integrated head tracker, audio and microphone.

As the name implies, a Head-Mounted Display is a device attached to the head with either one (monocular) or two (binocular) display optics in front of the eye. Besides

[1]Photograph by Sensics, Inc.

the number of displays, HMDs are separated into closed-view and optical see-through devices. Optical see-through devices use semitransparent displays which allow a direct view of the real world. Visual augmentations are projected onto these displays into the view field. In contrast closed-view devices use opaque displays. The user's view is provided by one or more video cameras. The video signal of these cameras is combined with augmentations and then and displayed to the user.

In 2000, Arthur compared a variation of HMDs [Art00]. He concluded that both, weight and limited field of view (FOV) reduces task solving performance. Unfortunately, the wider the FOV provided, the larger and the more expensive the devices were. Acquisition costs of more than $100,000 and dimensions as seen in Figure 2.1 limit the field of application drastically.

Today, AR devices are smaller and less expensive than the devices Arthur used. Zeiss offers its *Cinemizer Plus*[2] for about $300, a closed-view HMD for home entertainment. A comparable $500 see-through solution designed for winter sports is the *Transcend GPS Goggle* by Recon and Zeal[3]. However, consumer HMDs like the previous mentioned usually offer a FOV of about 30 to 40 degrees. With respect to the human eye with a FOV of 75 to 90 degrees these devices are not supposed to achieve an immersive effect. Additionally, closed-view HMDs with integrated high resolution cameras and an acceptable frame rate are – if available – still very pricy. The possibility of interception like switching real world objects or hiding them – which is a prerequisite of ARbInI – requires such devices though. Thus, hardware limits for this work are defined by *Trivisio ARVision* prototypes which were already used in previous experiments conducted in the C5 project [DMH+09]. A more detailed description of the devices used will follow in Section 4.3.

2.2. Augmented Reality

First mentioned in 1992 by Caudell, the most common definition of AR was introduced by Azuma [Azu97] in 1997. Azuma defines ...

> [...] AR as systems that have the following three characteristics:
> - Combines real and virtual [environment]
> - Interactive in real time
> - Registered in 3-D

He points out that AR is not necessarily coupled with HMDs but keeps his definition technical. 3D movies (not interactive) as well as video signals with a virtual overlay (not registered in 3D) are not considered as AR. No more details about quality

[2]http://www.zeiss.de/cinemizer
[3]http://reconinstruments.com

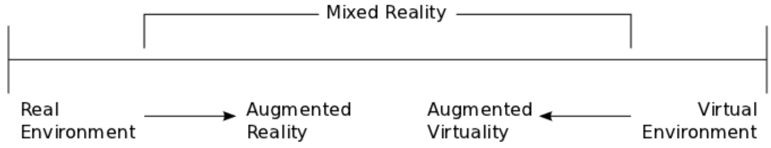

Figure 2.2.: Reality-Virtuality Continuum. Mixed Reality is a continual space where ratio of real and virtual elements can differ.

and purpose of provided virtual information were stated. Some researchers postulate the provision of extra information to a user's perception that otherwise would not be perceivable [KMS07]. To distinguish AR from Virtual Reality (VR) and Augmented Virtuality, Milgram et al. introduced the Reality-Virtuality Continuum [MTU+94](see Figure 2.2).

Compared to Virtual Reality, AR offers some simplifications related to simulations. In a traffic scenario as described by Behzadan and Kamat [BK08] many elements do not have to be rendered and thus creating a more realistic environment. Real world objects offer a more immersive experience and do not require massive computer graphic processing power. If see-through HMDs are used, a higher resolution and better frame rates can be achieved as well.

The capability of AR adding otherwise not perceivable information to one's perception offers interesting possibilities for a variety of applications. Often mentioned fields of interest are architecture [BLO+04], surgery [BFO92], assembly [CM92] or rescue and military settings [LRJ02]. These scenarios usually include the use of HMDs. However, other hardware platforms for AR are emerging. Modern mobile phones offer enough computation power and a variety of sensors to be used for location based services, games or aids to orientation. With the help of various projection devices, Spatial Augmented Reality became popular for design and art performances. The scenarios range from playing with ambience [Kir08] over indoor exhibits [Sup10] to public performances [Bar10].

Gartner Technology Hype Cycle shown in Figure 2.3 depicts that AR is about to pass the *peak of inflated expectations* and heading towards the *trough of disillusionment*. In other words: High expectations present today towards AR cannot be achieved. Economically this leads to a drop in research and development funding and former business opportunities will be revealed as not adequate.

In 1997 Azuma stated a list of obstacles AR has to pass to be used productively. The most problematic issue he called the *registration problem* which addresses the lack of accurate positioning of real and virtual objects. Thus, he concludes that the biggest single obstacle to overcome is the lack of accurate, long-range sensors and trackers

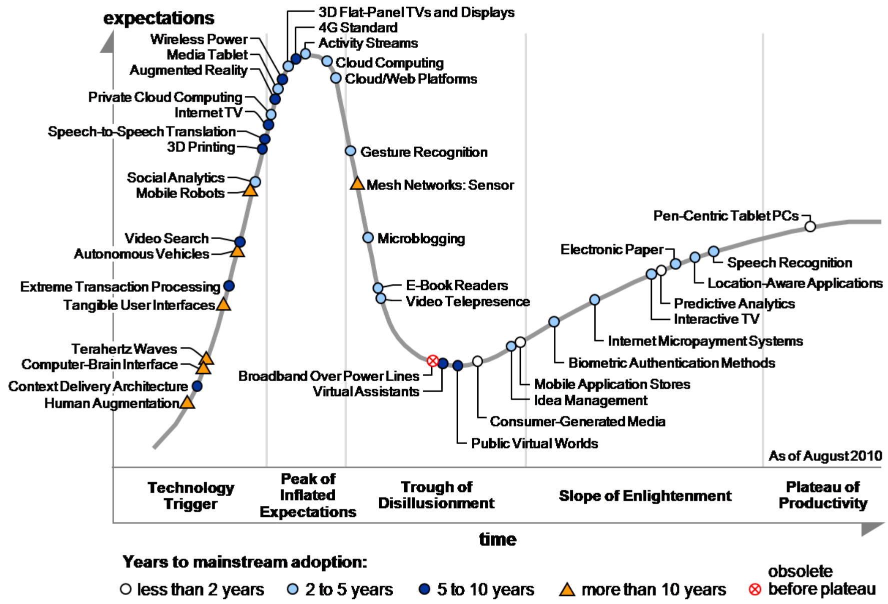

Figure 2.3.: Gartner Technology Hype Cycle. According to Gartner, Augmented Reality will shortly pass the Peak of Inflated Expectation. The maturing process will take additional five to ten years until mainstream adaption will be achieved. Figure reproduced from [Gar10]

to retrieve correct positions of users and objects in real world. In contrast to VR, a greater input variety and higher bandwidth needs will have to be faced as well.

Other obstacles like display performance and quality of virtual objects were mentioned but classified as minor issues since their importance for VR is significantly higher than for AR.

While other basic performance issues such as lack of computation power has been solved or will be solved in the near future, tracking and sensing systems with an accuracy and speed required for AR applications are not available yet.

Bimber and Raskar compare AR's maturing process with closely related and more mature VR with the help of a building blocks model (see Figure 2.4) which addresses the same low-level issues Azuma did earlier. Bimber assumes that when AR matures, the research focus will shift from low level features like tracking and registration to higher abstraction levels like authoring or presentation. If we take a look at a 10 year review of the International Symposium on Mixed and Augmented Reality, we see that tracking, calibration and display technologies are still in the 5 most issued research areas [DB08]. This allows the conclusion that these issues are far from solved.

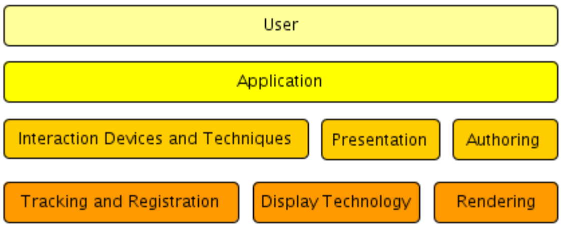

Figure 2.4.: Bimber's Building Blocks. Bimber describes AR application architecture to be very similar to the architecture of VR [BRB05].

Gartner's hypothesis is backed up by these findings. However, this is a very typical development process of new technologies. Even though initial desires cannot be satisfied, chances are good that AR will pass the *trough of disillusionment* and enter the *plateau of productivity*. According to Duh et al., bringing AR from laboratories to industry and widespread use is still a huge challenge, but there is a strong interest of academia and industry to push AR to the next majority level [DB08].

2.3. Alignment in Augmented Reality

As mentioned earlier, this work shares context and follows Christian Mertes' work conducted within his diploma thesis. His introduction to the DFG Collaborative Research Centre (CRC) and related concepts is still valid.

2.3.1. Definition

Mertes writes in his thesis:

> Alignment is a comparatively recent concept in human communication research, first presented by Pickering and Garrod [PG04]. It describes the implicit mutual convergence of the internal states of dialog partners through very basic mechanisms. On the level of the representation of a certain situation, this means that interlocutors can focus their exchange on the chunks of information that are most responsible for this particular mental representation. They do this by relying on the assumption that their vis-à-vis functionally shares the same underlying structure to represent situation models and thus can be shifted into the same condition by some key stimuli. Obviously, this greatly reduces the amount of information that has to be transmitted, compared to a complete description of all relevant

> parts of their internal state. Because the assumption of equal underlying structures usually does not perfectly hold true though, alignment is an interactive process during which the right amount of information that is necessary to reach a common ground is progressively established. It is also an automatic process which means that it goes easy on cognitive resources.
>
> The Collaborative Research Centre (CRC) Alignment in Communication (Sonderforschungsbereich 673) – from which the project described in this work originates – was founded for the sole purpose of studying this phenomenon. While Pickering and Garrod in their original paper focused on a more linguistic view on alignment, highlighting alignment at the different structural levels of language, the CRC takes a more general approach to alignment as I did with the above characterization. So in the CRC's proposal Rickheit et al. [RJR+05] define alignment very similarly to be
>
>> [...] an ensemble of verbal and non-verbal means that serve to increase the similarity in structure of two interacting dynamic systems in a largely automatic and non-reflexive fashion, without an explicit exchange of information on system states.
>
> This definition is not even restricted to human interaction, so in the CRC there are many different research projects, ranging from purely linguistic ones to those situated in the field of cognitive robotics.

The alignment state offers a very efficient way of communication. With insights about how alignment works, we hope for a better understanding about the underlying mechanisms of human-human interaction. Additionally, findings might help to develop machines which are able to simulate alignment with their users for a more natural and intuitive way of human-computer interaction.

2.3.2. Relation to this work

One project of this CRC is the C5 Alignment in AR–based cooperation. According to Mertes, the C5 project uses AR for the benefit of alignment research in three ways:

> (1) To record data from two interlocutors and to analyze this data with respect to alignment that might take place, (2) to interfere with the subjects' interaction by augmenting their perception differently, leading to controlled misalignment, (3) to create artificial means by which alignment might be facilitated.

Within this work we want to continue the work on a platform which is suitable to support all of the mentioned three goals. Research, design and development of this platform will be introduced in Chapter 4.

2.4. Designing Interaction

Regarding Section 2.3 a research system for monitoring user interaction is required. That means, user satisfaction is not the primary goal in the collaboration analysis setup. Stakeholders of the system are the researchers working with the collected data rather than the people supposed to solve tasks. So why is it important to design for usability even though the stated task is more or less a distraction? As mentioned by Mertes one goal of C5 is to *create artificial means by which alignment might be facilitated* which seems to be the most obvious answer to that question.

But it is vital for observation and analysis of alignment as well. People should forget about the system and interact as naturally as possible for valid observation results. However, interacting in a natural way attached to the available hardware is not that easy and it is almost impossible to forget about it. Since no unobtrusive hardware for a *closed-view* HMD set up is available, we have to strive for user acceptance through usability and feature-rich interaction.

Concerning acceptance we already have a big advantage compared to a similar system in productive usage since using the system is a part of the subject's task. However, lacking usability and robustness led users to develop *workaround* behavior in the past. Instead of the task, the system became the big issue to overcome. We have to prevent this if we actually want to be able to monitor interaction *through* our artificial communication channels instead of *with* them.

Preece et. al [PRS02] outline Interaction Design as the process of

> designing interactive products to support people in their everyday and working lives.[...] In particular, it is about creating user experiences that enhance and extend the way people work, communicate and interact.

Even though our AR system is not supposed to be used in everyday and working life, the definition still covers our goals.

The IxD process consists of four basic activities which – according to Preece – are:

1. Identifying needs and establishing requirements.
2. Developing alternative designs that meet those requirements.
3. Building interactive versions of the designs so that they can be communicated and assessed.
4. Evaluating what is being built throughout the process.

Furthermore there are three basic principles to consider during the design process:

1. Users should be involved through the development of the project.
2. Specific usability and user experience goals should be identified, clearly documented, and agreed upon at the beginning of the project.
3. Iteration through the four activities is inevitable.

Consider the people you are designing for, communicate what you plan to do and adapt your plan since the chance to do everything right at the first time is fairly low. This sounds intuitive and easy to follow but practically it is often disregarded.

2.5. Interaction Analysis Methods

Getting to know the *use system* (see Figure 2.5) before designing solutions is a major task in interaction analysis [Eng04]. In the beginning of a product design process just a few requirements are known. A very common elicitation method is the conduction of a form-based survey. One can reach any number of people in a short time period and thus collect a fair amount of data. However, the questionnaire's success regarding their return rate and the value of data depends on many factors like choice of survey target group, distribution strategy (eg. paper-based or digital) and of course the survey design.

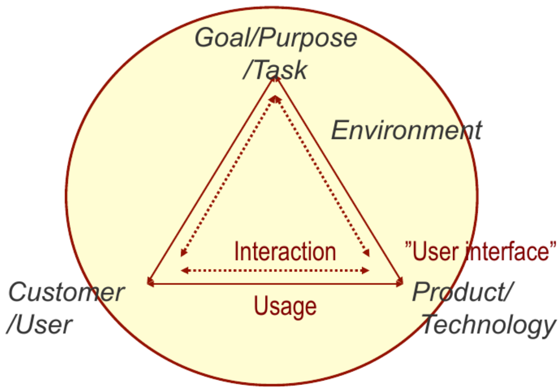

Figure 2.5.: Use System[4]. Four basic elements make up the use system. User, Goal and Product are embedded into the environment. Changes conducted to one element are usually followed by adaptions of the other elements.

[4]Scheme created by MariAnne Karlsson with inspiration from Yrjö Engeström's work on Activity theory.

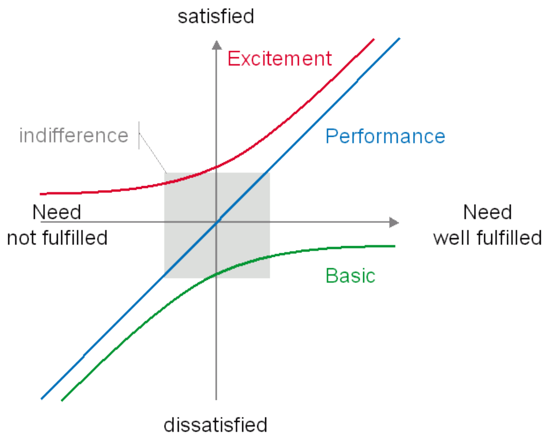

Figure 2.6.: Kano Model. Mandatory features have to be fully implemented for user acceptance. Additional features' impact is hard to predict and usually start to boost product performance not until a certain complexity level was achieved [Coh88].

Being a powerful method, questionnaires alone are not enough to elicit all types of requirements. Many of them are *below the surface* which is described by the so-called *Iceberg Phenomenon* [Ber04]. Even though they are unknown, those requirements have to be fulfilled to gain user acceptance and satisfaction (see Figure 2.6).

Christel and Kang [CK92] define three major requirement elicitation problem categories:

1. **Problems of Scope**,
   in which the requirements may address too little or too much information. Limit product design possibilities too early can cause the *real* issues shifting out of scope. On the other hand, providing too few details about the targeted product will hinder accurate issue identification.

2. **Problems of Understanding**,
   within groups as well as between groups such as users and developers. This includes unawareness of problems as mentioned earlier but also the lack of understanding the capabilities of a system. On the other side, developers unaware of the characteristics, needs and demands of users will lack elicitation capabilities.

3. **Problems of Volatility**,
   i.e., the changing nature of requirements. Things people favor today might be obsolete in the near future.

Suitable methods to *lift* hidden requirements are observations and interviews. As for surveys, the success of these methods depend on various factors. In order to produce useful observation results, one has to know *what* should be observed, *where* it takes place and *how* it should be observed. Albeit these questions seem to be easy to answer, uncertainties often reduce observations' efficiency.

For example, imprecise observation instruction like "what do customers do in a shopping mall" will not return satisfying results since the pool of possible actions and interactions to observe is too big. The findings of "what do people do while they are calling friends" will differ according to the place there are in and if they are aware of being observed.

However, it is impossible to identify all issues. A precise understanding of the *use system* and the related requirements has a big advantage though: Design can be conducted with an accurate goal and the risk of critical changes during the development process – which are usually way more expensive and/or time consuming – is lowered.

The development process should not be confused with the second of the basic IxD activities. As mentioned in the previous section, iterating through the design activities is encouraged. Design decisions used to produce a final product or an increment of it however, will cause additional costs if they get declined in an advanced stage of the development process.

2.6. Phaenomeum

Dierker et al. developed a cooperative planning task scenario for her interaction research called Phaenomeum [DPH11] which can be seen in Figure 2.7. The name is related to interactive exhibitions located in Bremerhaven, Flensburg, Lüdenscheid, Peenemünde and a traveling exhibition called Phänomenta. The goal of this series of exhibitions is to familiarize visitors with the laws of physics through self-conducted experiments [FK99]. The exhibits themselves are interactive and some of them emit sound and/or light and some of them need silence and/or darkness to work correctly.

In this setup two subjects are given the task to create a layout plan for an exhibition with a variety of exhibits. There are two identical layout plans, one in front of each subject. Small paper signs on a wooden cube or augmented images of the real world exhibits are used as placeholders. The images show the required environment of the exhibits as well to imply constraints. These constraints influence placement since space is limited and divided into several rooms are of fixed size.

In contrast to experiments conducted in [DPH11], we aim for a single shared area which is manipulated by both subjects at the same time (see Figure 2.8). We have observed that if both subjects create their own solution, changes hardly occur when

Figure 2.7.: Phaenomeum setup. The images shows the discussion phase of a previously conducted *Phaenomeum* experiment. Both participants created an individual solution first and are about to compare it.

they are confronted with the interlocutors solution. Common behavior consists of deciding which solution to take and only conducting minor adaptions. By limiting the resources which have to be shared we expect more discussion during the work on a solution. Subjects will be located either opposite or to next to each other on table to observe whether a similar or an opposite point of view (POV) ease the task solving process. Instead of images we use 3D-models as augmentations so that the exhibits can be spatially organized appropriately. Images of the exhibits as well as the related 3D-models can be found in Section 4.4.4.

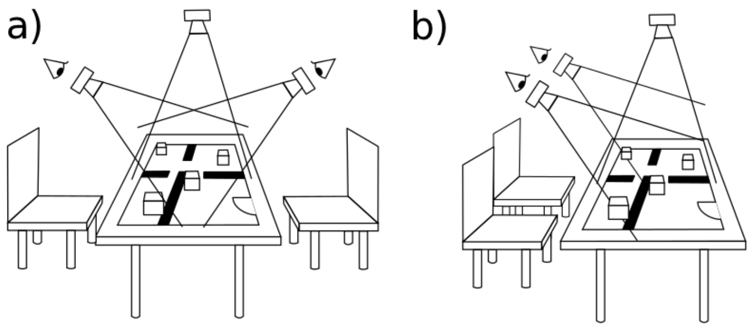

Figure 2.8.: Alternative Phaenomeum Setup. Two subjects sit on a table opposite (a) or next to each other (b). On the table there is a layout plan and cubes as placeholders for exhibits. In the AR scenario a camera mounted over the table will provide real-time video streaming in addition to the streams from the subjects' HMDs.

2.7. Homogeneous Transformations and Coordinate System

In Chapter 4 and 5 we will talk about spatial reference systems and transformations between them. Albeit profound knowledge is not required, we want to introduce a transformation notation and standard which we will use later on.

Many scientific and non-scientific disciplines have to deal with the description of position and rotation in three dimensional space. Unfortunately, there is no standard defining how world coordinate systems and objects located in them have to be oriented. We will use so-called right-handed coordinate systems since those are widely accepted in computer graphics. Additionally it is assumed that objects are facing down the Z-axis in their local reference system with the Y-axis facing up. Derived from that assumption cameras and associated viewports however, face into the *negative* direction of the Z-axis. That might sound confusing but if one imagine the simplest possible scenario including one object and one camera in a world space, the object facing down the Z-axis, the camera has to face into the *opposite* direction to be able to see it. In Chapter 4 one will see, that other approaches exist as well.

Points or better vectors will be notated as ${}^{system}p_{fromTo}$ and homogeneous transformations as ${}^{to}T_{from}$. To show the benefit of that notation let us have a look at Figure 2.9.

The position P is known in the system $S2$. It is notated as ${}^{S2}p_{0P}$ which means it is known relative to its own origin. All systems know their positions and orientations in the world coordinate system and how to transform local data into global data and therefore know ${}^{G}T_{S}$ which describes that information mathematically.

To get the world position ${}^{G}p_{0P}$ we need a transformation ${}^{G}T_{S2}$ because

$$ {}^{G}p_{0P} = {}^{G}T_{S2} \cdot {}^{S2}p_{0P} \tag{2.1} $$

If we want to know the position of P in $S1$ because no information are available in that system since the object is out of sight we do

$$ {}^{S1}p_{0P} = {}^{S1}T_{S2} \cdot {}^{S2}p_{0P} = {}^{S1}T_{G} \cdot {}^{G}T_{S2} \cdot {}^{S2}p_{0P} = ({}^{G}T_{S1})^{-1} \cdot {}^{G}p_{0P} \tag{2.2} $$

${}^{G}p_{0P}$ will be published by $S2$ and can be used with the inverted ${}^{G}T_{S1}$ to calculate the position in $S1$ space.

The indices connect equation elements neatly what makes homogeneous transformation a bit less confusing for people without profound knowledge in projective geometry.

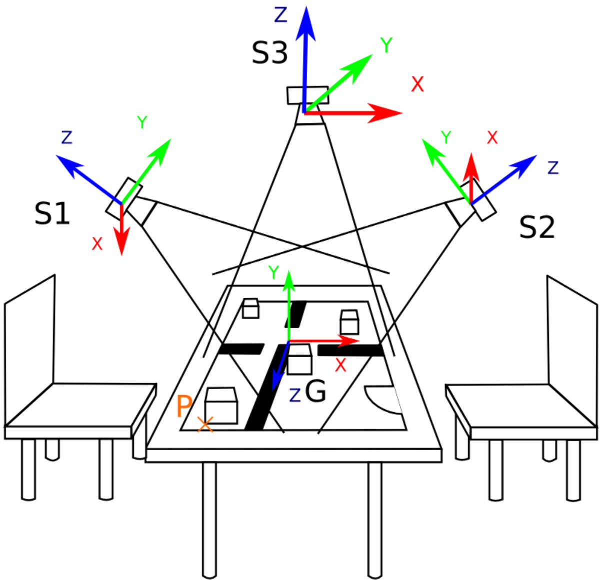

Figure 2.9.: Reference Systems. Position and orientation data have to be available in local reference systems for correct augmentation. The table reference system (G) will be defined as the global system or world space. Tracking results of an object (P) gathered in viewer's reference systems (S1-S3) have to be converted into that system to be usable for other local reference systems which do not know the publishing system's position in world space.

3. AR-Methods for Collaborative Planning Tasks

To identify the needs of the C5 project, requirement elicitation was done in the initial phase of the design process. Therefore, several methods were reviewed. The findings were summarized and will be presented in 3.2. Alternative design prototypes will be presented afterward. The chapter ends with a little summary of the design process.

3.1. Requirement Elicitation

We decided to choose a qualitative approach rather than quantitative like surveys. Observation based approaches – obtrusive or unobtrusive – could not be conducted since the specific usage scenario is rare and not common. Question based approaches seemed to be most appropriate.

As mentioned in Section 2.4, surveys for determining needs of potential customers and user tests for increasing usability and bug reduction are very common. However, most quantitative methods for requirement elicitation focus on the majority of users. Von Hippel et al. [HV92] claim that this approach is more time consuming and expensive while also being less productive than focusing on a few users – so-called lead users. As an example of successful innovation processes driven only by users, von Hippel mentions the open source community in general and projects such as Apache [Von07].

According to von Hippel lead users share two essential characteristics [Hip86]:

- Lead users face needs that will be general in a market place - but face them months or years before the bulk of that marketplace encounters them.
- Lead users are positioned to benefit significantly by obtaining a solution to those needs.

We decided to go with this basic idea of rather interviewing a few motivated and skilled stakeholders instead of a representative majority because this means a more efficient use of limited time and resources. Lead user identification was reduced to the context of the C5 project since basic knowledge of AR in general and the specific setup in detail were required.

If we subtract von Hippel's market focus, lead user characteristics are fulfilled by researchers working in this context. They benefit directly from improvements and usually are motivated to support progress because of interest[1].

With this in mind, three stakeholder groups were defined:

- **Conversation Analysts**
  A non-user stakeholder who works with the collected qualitative data. Since he is interested in direct human-human-interaction taking place, he would like to minimize system interference.

- **AR Alignment Research Experts**
  A non-user stakeholder who directs system development and analyzed quantitative data. She represents the developer and operator point of view. Familiar with AR technology and obstacles related to it, she does feasibility checks for planned features and improvements.

- **Subjects**
  A user who is willing to support science and interested in trying new things. She is open minded and aware of the fact that devices and methods are in prototype state. Additionally, she is more forgiving than *common* customers.

One candidate of each group was asked for an interview. All interviews lasted 45 minutes and were conducted with one interviewee and differing so-called mediating tools. Mediating tools were supposed to represent the final product and to support elicitation of *hidden* requirements [Kar96]. We used sketches of an HMD like point of view (see Figure 3.1) for quick drafts. During the subjects' interview the Trivisio HMDs were used as mediating tools as well. To limit distraction and breaks in conversation flow all interviews were conducted in calm areas and were recorded to free the interviewer of the need to take notes. The user interview was conducted in an AR test laboratory to provide another stimuli. probing was used to focus on topics promising more detailed insights.

[1] We rely on common ground here. Since extraordinary wealth cannot be the reason for skilled academics to prefer a PhD spot, it has to be interest in science.

3.1.1. Interviewee Profiles

Conversation Analysis

Interviewee Christian Schnier is currently involved in the C5 project and analyzes the multimodal interaction corpus data recorded with the recent system. More precisely he is interested in video recordings from previously conducted experiments since these are used for conversation analysis.

AR Alignment Research

Angelika Dierker served as an interviewee for the AR Alignment research group. During the last 3 years she conducted several experiments using the current LAFORGE system. She supervised maintenance and improvement of the software system. Thanks to this expertise, she knows what to expect of AR systems concerning recording and detection of alignment.

Subject

Sebastian Stief volunteered for a previous C5 study conducted two years ago and served as an interviewee for the elicitation process. The time passed since his participation might sound like a disadvantage due to fading memories but it offered another information dimension. Interestingly, it revealed varying significance of specific features and properties. The task he had to solve back then and features he considered to be useful he could remembered. However, features considered to be less helpful he had forgotten. For example he could remember a visual gazing augmentation which was evaluated to be useful by most participants whereas the less well evaluated sonification feature he had forgotten.

3.2. Elicitation Results

All interviews were reviewed and summarized[2]. Issues were rated according to the frequency they were mentioned and the experienced severity. The results from the interviews were combined with insights gained from footage studies of previously conducted experiments in [DMH+09]. In the following we will present the identified key issues and discuss their reasons and impacts.

[2] Interviews were handled internally and will not be published

3.2.1. Performance

Hardware Dilemma

Size and handling of the AR gear used during previous studies were criticized by all stakeholders. Issues experienced were similar to problems reported by subjects taking part in Arthur's studies [Art00] which were oculomotor discomfort, disorientation and nausea. Another common symptom was a hurting nose bridge caused by the HMD's weight resting on it. As mentioned in Section 2.1 hardware is still an obstacle which has not been overcome in a satisfactory manner yet. Monitoring of further progress in this field is encouraged but cannot be supported by software related design decisions.

Tracking Stability

A major issue brought up by the interviewees was the insufficient tracking stability. As mentioned earlier this issue is a very common one but when it comes to interaction analysis it gets even more important. While it is "just" annoying for users, it was considered critical for the validity of the collected data from a scientific point of view.

Tracking instability occurs in different shapes. A slight jitter causing shaking objects and a minor offset were issues subjects could get used to. Resulting compensation behavior by subjects did not influence task performance and experiment progress. Even wrong orientations or upside down objects – known as *marker flipping* – did not seem to influence subjects' behavior if it occurred temporary. The footage study supports these assumptions. More troublesome was a rapid change of position if markers were detected in wrong locations. Subjects stopped working instantly if augmented objects started to *fly* through the room even if this occurred just for a fraction of a second. The biggest issue however was a permanent *false positive* marker tracking. Markers were confused by the tracking engine which lead to wrongly placed virtual objects. Since both HMDs were connected to their own tracking and augmentation systems this led to contradictions through differing world models.

For example, subject A was referring to an object with pointing gestures and an oral description of what she saw which confused subject B since the pointing gesture and the described virtual object did not match. The resulting loss of trust in the AR system made subjects always double check virtual objects or in the worst case ignore augmentations and exclusively rely on gestures and verbal information.

3.2.2. Processing Features

Video Data Augmentation

For data analysis it is necessary to keep track of a subject's ongoing discussion and to understand the model state both interaction partners share. In the recent implementation, augmentation is limited to the users' input streams. Video signals from observation cameras which were not connected to the system stayed unaugmented. Post-processing these signals was proposed by Schnier to ease conversation analysis. One issue he experienced was the lack of augmented information in the previous mentioned footage. Subjects often referenced virtual objects which required a permanent shift of one's attention to different video data.

3.2.3. Usability

User Engagement

Dierker pointed out, that from a scientific point of view the task and the system is something necessary to observe conversations. More precisely, the task itself is some kind of distraction to get the subjects to interact. It does not mean that task and system are secondary. The contrary is the case. System and task are supposed to encourage the subjects to interact and create *engagement*.

Generally speaking, *engagement* or *commitment* defines the level of how involved someone gets while solving a task. The highest form of engagement is called *flow*. The term was proposed by Mihály Csíkszentmihályi and has become very common in many scientific fields [Csi08]. Encouraging flow is a major goal in product and game design today since flow and user satisfaction correlate. Achieving flow seems to be a very ambitious goal but every improvement that makes subjects forget all the equipment which they are attached to – or even start to feel comfortable with it – is desirable.

During video analysis some effects could be observed. A fair amount of users have never handled HMDs and AR systems before and thus felt uncertain. If primary handling issues were solved, people usually got excited about seeing virtual objects embedded in the real world. Unfortunately this excitement decreased quite quickly when exploration of the system reveals a limited set of features.

A wider explorable space might extend this period of excitement. However, additional features – if not deactivatable – might overburden some user groups. Even technology-friendly users might feel limited if the system draws too much attention. Designing and evaluating a variety of augmentation prototypes could help to get closer to an acceptable balance.

Attention Focus

In real world human-human communication eye contact is essential. HMDs block this communication channel entirely and hinder communication flow massively [Mer08]. Mertes implemented a gaze augmentation to compensate the negative effects which was reviewed positively by subjects[DMH+09]. As proposed, more advanced augmentations to indicate the interaction partner's focus of attention seem promising to compensate for eye contact limitations.

Constraints

Footage observations of previous *Phaenomeum* studies revealed that the given tasks might have been too simple for the subjects. The highlighting feature mentioned in Section 3.2.3 might compensate the lack of intervisibility to a certain level but all in all a simple planning task does not benefit from AR usage. In addition the lack of guidance seem to let people confused because there is no clear *winning* moment indicating that the given task was solved appropriately. It is up to the subjects to define this moment according to implied constraints through images of exhibits that indicate excessive light emission or the need of darkness. However, the importance of these constraints were defined by the subjects and dynamically adapted if a constraint was considered to be annoying.

In *Rules of Play* Salen quotes Bernard Suits to introduce the danger of an unbalanced rule set [SZ04]. If winning is too easy the game becomes dull. Extending the rule set with defined constraints is supposed to clarify the task and indicates whether the task is solved in a satisfying way. The principle of *solved by agreement* stays intact if the rule set just allows a minimization of constraint violation rather than a perfect solution. Even though we have to relativize here because creating an enjoyable game is not a goal in our scientific context, encouraging flow and engagement is desirable.

3.3. Design Prototypes

The first step towards the creation of working prototype implementation was a low-level prototyping via *sketching*. Therefore images of augmentation objects were manually augmented with the help of an image editing software (see Figure 3.1). Thereby the visual concepts could be studied without the burden of a full implementation in software. Each sketch was then evaluated concerning several categories. First of all we evaluated the chance of augmentations to cover other vital information. The task of the prototype is to add information, not replacing real world information with virtual – and probably less useful – information. This counts for visual augmentations which might cover their related real world object as well as sonification approaches which hinder linguistic interaction. Secondly, we evaluated the necessity of background knowledge needed to understand the augmentation. Features should be usable without much learning time since the targeted user group will just use it for less than an hour. Thirdly, we thought about the impacts of the metaphor used. Assisting features which will be perceived as controlling and demanding do not support the goal they were designed for.

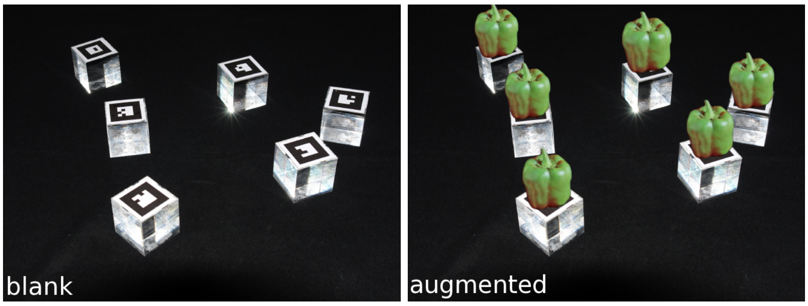

Figure 3.1.: Sketch templates. Design prototypes were created with the help of templates[3]. The images show the original picture with ART markers attached to cubes and a sketch with model mock-ups used as placeholders for augmented 3D models.

3.3.1. Spatial Constraints

As mentioned in 3.2.3 enriching the rule set for the exhibit design task with additional spatial constraints is supposed to make the task more challenging. For example, a spatial constraint could be that an exhibit which requires silence needs to be positioned at least 5 meters away from a sound emitting exhibits. Additionally, we now have a scenario in which subjects can benefit from the technology used if we consider getting

[3] Photograph by Christian Mertes

a value/feedback of a computed distance to be less hassle than measuring the distance manually with a ruler or tape measure. This might be questionable for a few objects but if the amount of constraints to be considered exceeds natural working memory capacity the hypothesis seems to be valid. Miller introduced the *magical number* of 7 elements/chunks which could be processed simultaneously [Mil56]. Initially chunks were independent of their *real* type (e.g. digits, letters, numbers of words), but later studies revealed an inversely proportional relationship between chunk complexity and number of elements [Cow01].

Violation Visualization

Radius disk around objects could be used to indicate constraint violations (see Figure 3.2). This involves the violated and the violating object. Transparency indicates the level of violation. An opaque disk implies highest level.

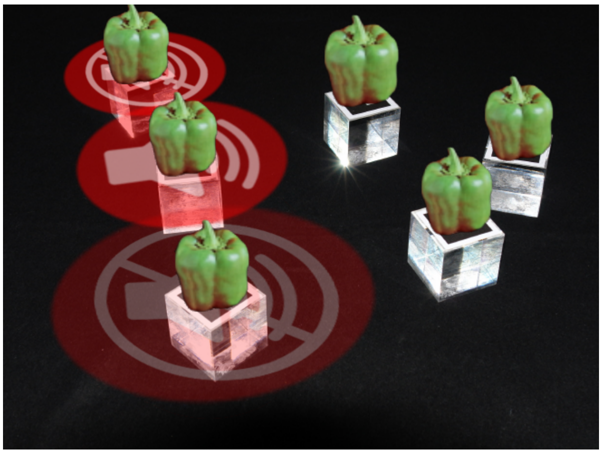

Figure 3.2.: Violation Visualization. In this example a violated distance constraint is visualized by a red disk. Colored intensity is supposed to indicate level of violation.

The visual impact most probably will be acceptable since no vital information is covered. However, the more violations take place at the same time, the more likely cross interferences with other visualizations occur. Besides the continuous information from transparency, the radius of the disk itself tells us about the maximum impact area of the related object. These kinds of information visualization is quite common and most likely to be understood by most users. Its gradient transparency make minor violations acceptable if the user gets used to it. The chosen signal color red[4] is supposed to be

[4]If no red-green color blindness exists

visually salient. The visual salience increases proportionally to the viewing angle on the XZ-plane.

Violation Sonification

Instead of visual augmentations, sound can be used to indicate constraint violations (see Figure 3.3). This will allow an instant evaluation of the chosen arrangement. Severity of violations could be sonified by changing volume levels.

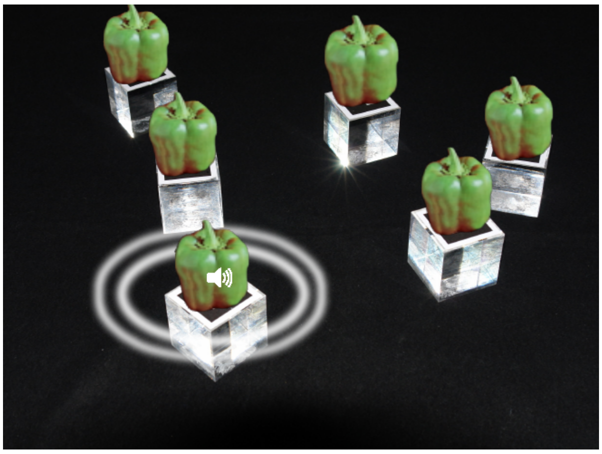

Figure 3.3.: Violation Sonification. The idea of the *violation sonification* sketch is to indicate distance constraint violation by a sound sample played continuously (or once when the violation occurs). A visual wave emitted by the related objects can be used to assist object identification.

Impact depends heavily on volume parameters and numbers of occurring violations. However, while a small amount of violations are likely to be detected easily, a large number could cause an uninformative noise carpet. Usability is significantly influenced by the samples chosen.

However, it is necessary to be capable of connecting these samples to the related objects without much background knowledge. A minor visual indicator may help to locate ongoing constraint violations.

3.3.2. Attention Focus

Even though we added features to make AR support more desirable, we still have to compensate for the lack of face intervisibility. Seeing the interlocutor's eye's helps to estimate gaze direction and focused areas of objects and gives evidences about recent attention focus. With HMDs this information channel is not available and it might be useful to provide this missing information in another way.

Shadow Indicator

In the *Light of Attention* concept, a virtual spotlight, right at the intersection of table surface and one's interaction partner's view ray, causes virtual objects to drop shadows which indicate the position of the virtual light to the user (see Figure 3.4). The idea is to make use of the participant's intuitive capability to locate light source with the help of shadows. With multiple objects in sight, accuracy is likely to be increased without much danger of covering crucial information. In addition this feature is peripherally monitorable like the flickering light of a candle. Long shadows and slow movement imply the absence of a partner's gaze while short and steady varying shadows indicate partner's attention.

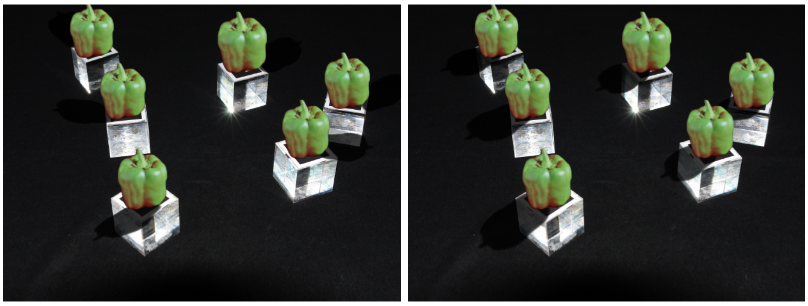

Figure 3.4.: Light of Attention. In this example a light source is placed at the intersection point of the user's interaction partner and the table plane. If the other user looks right into the center of cube collection shadows will point in multiple directions (see left image). Shadow length and direction indicate where the user's attention can be expected even though it is out of sight(see right image).

Glowing Frame

A concept very common in modern video games, especially in first person shooters is the *glowing frame*, either to direct a player to a certain spot or to indicate from which direction a threat is approaching. The user is supposed to turn his (virtual head) into the glow's direction.

In our scenario a steady augmentation is targeted as shown in Figure 3.5. The frame is supposed to vanish, if both users look at the same spot. If the interlocutor looks directly at the user, the whole frame glows. The usage of the whole border region theoretically is a drawback but concerning the HMDs used (see Section 4.3) this region could hardly be used to display information.

Since this prototype is inspired by a video game feature some original intentions are

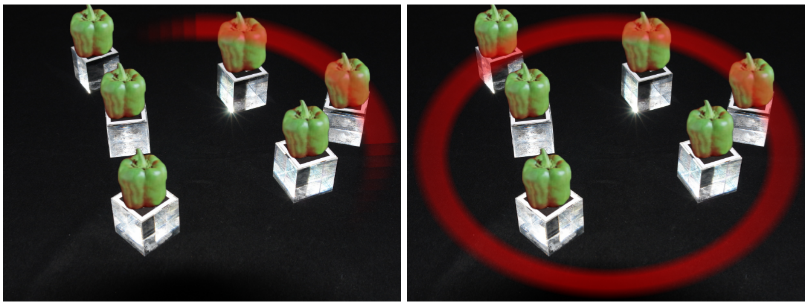

Figure 3.5.: Glowing Frame. These sketches show the glowing frame prototype. The relative position of the interlocutor's field of view is indicated by an arc (left image). If the interaction partner is looking right at the user, the whole boarder glows (right picture).

inherited and have to be taken in account. In video games the occurrence of the glowing frame usually implies the necessity of immediate action as seen in Figure 3.6. If people are familiar with the metaphor, the assistance aspect might be perceived as an instruction rather than a support.

Figure 3.6.: Being attacked in Call of Duty 4[5]. The screen is pulsing red and an arc indicates the direction from which the attack originates. The message is: Act immediately!

[5]Registered trademark of Activision Publishing, Inc.

Hiding Objects

In Figure 3.7, a quite drastic way to enforce shared field of view is shown. The basic idea is to create circumstances which require participants to focus on the same area and hide all objects which are not in both subjects' field of view. This can be even strengthened if the rest of field of view is blackened out. The visual impact is massive but offers much space for supplying additional information since real world information is reduced to a minimum. This approach is not focused on improving usability, but to encourage participants to use other communication channels since it cannot be ignored. It is anticipated that workflow will slow down significantly.

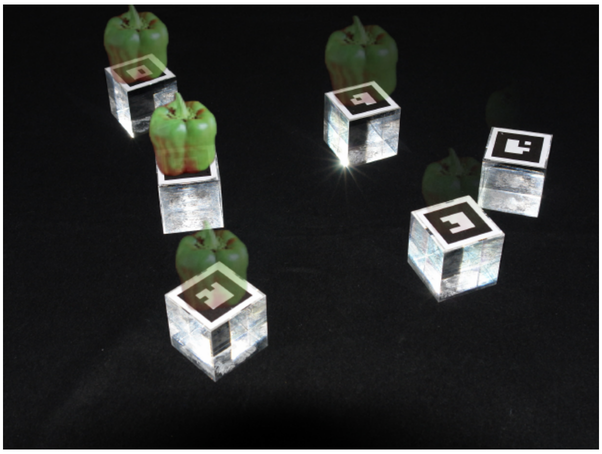

Figure 3.7.: Fading Objects. Hiding objects from the scene could be used by purpose to direct subject's interaction behavior. For example, if users are supposed to spread their attention, one could hide objects which are in both user's field of view.

Alternatively, this technique can be *inverted* to hide all objects the interaction partner is looking at. This can be used as a *conversation task* but also might be useful in situations where the subjects jointly need to monitor a set of objects and the wish is to prevent redundant observations.

3.3.3. User Engagement

Another opportunity for AR technology is to improve immersion. Planning tasks usually require users to be capable of converting an abstract model into a imaginative real world representation. This includes replacing planning placeholders with real objects as well as switching the bird's eye view of planning and pretending to be part of the model. Thanks to AR, the scene could be made virtually accessible which might ease the planning task.

Virtual Visitor

A virtual visitor – an object representing a person within the model – and its point of view ease this transposition and support instant change of perspectives. Besides the possibility of a cognitive walkthrough, varying perspectives can be taken into account.

While planning an exhibition a virtual visitor could be used to evaluate settings from a *grown up* perspective as well as from a child perspective which differs significantly. Buttons or displays placed out of reach of children might influence disabled visitors as well.

A small window placed in the corner of a subject's view appears when the connected marker enters the setup as shown in Figure 3.8. The picture is *pure virtual* and contains only exhibits and an orientation grid representing the floor. In the main view a symbol is shown which indicates view direction of the virtual visitor. If the marker is removed the view is disabled as well to free visual space.

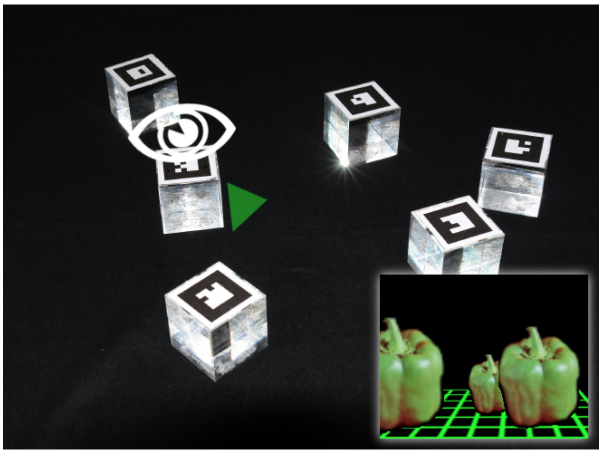

Figure 3.8.: Virtual Visitor. A virtual *exhibition visitor* can be attached to an additional marker object which can be used to inspect the exhibition from that perspective. The visitor's view can be embedded into the user's HMD by a frame-in-frame visualization shown in the lower right corner.

Attention Request

The video footage shows that people often lift object they want to discuss. A rotating circle around the object could be used to communicate this intention (see Figure 3.9). Depending on the distance to the announced user's point of view the radius of the circle would vary so that the arc would stay in sight. This not only inform the other user about the intention of his interaction partner, it also offers him direction and distance information. Related research shows that this information can be extracted easily by many users. For example, Baudisch and Rosenholtz showed that orientation with arcs can be done faster than with the use of arrows [BR03].

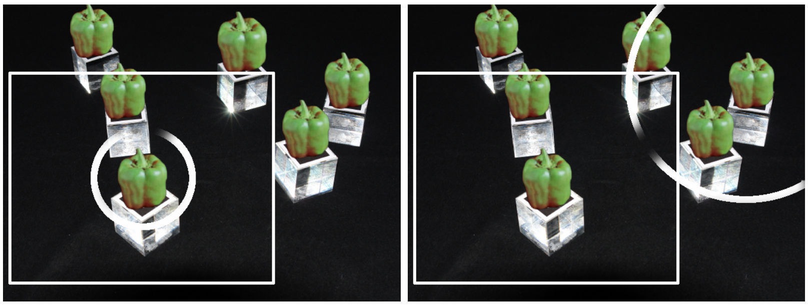

Figure 3.9.: Attention Request. If a user wants to talk about a specific object she can indicate that by shaking an object for example. Depending on where the other user is looking at this moment a circle is drawn around the targeted object (left picture). If the object is out of sight the radius of the circle extends until it is in the view frame of the announced user. The arc not only indicates where the action is taking place but also gives information about how far the object is away from the recent field of view.

The visual impact is supposed to be low and most certainly no other vital information will be covered. However, the rotation of the arc might interrupt the accounted users actions. This is the desired effect of course but holds some potential for significant behavioral changes in regards to interaction with the objects. If the attention request is toggled too easily, users might tend to push the objects instead of taking them to avoid this behavior. On the other hand, if toggling the request is too time consuming people might rather tend to hold the object into the other user field of view instead of waiting for her to end her current actions.

3.4. Design Process Summary

Earlier we talked about the four basic activities of IxD (see Section 2.4). In the design phase we conducted the first two steps. Various augmentation types that make task-specific, setting-specific and user-specific and information mutually accessible were prototyped in form of sketches. The guiding principle for these sketches was the *added value* that the augmentations could give to support cooperation under the constraints of the AR setting. However, while each of the sketches seems to offer helpful information, the augmentations naturally also interferes with the limited resources the users have. More sophisticated prototypes are required to prove and evaluate the benefits which have been qualitatively speculated here. To build these *interactive versions of the designs so that they can be communicated and assessed* LAFORGE 3 was developed and will be introduced in the next chapter. Testing and evaluation of the prototypes is not a part of this thesis though.

Testing however, should not only investigate the benefit of each single prototype, but also evaluate emerging effects which might appear when they are combined. The question to answer is at what point the user becomes overloaded by permanent visual flux that distracts more from the interaction with the interlocutor than being helpful.

4. Laforge 3 - A Platform for AR-Prototyping

This chapter is about the reimplementation of the Lightweight Augmented-Reality Facility with Open Real-world-based Graphical Enhancement (LAFORGE). We will have a look at the recent LAFORGE system, followed by several other available AR frameworks. This review also mentions why we decided to reimplement LAFORGE instead of refactoring the recent version or dropping it for another framework. Afterward unique features[1] of LAFORGE 3 will be stated and its system architecture introduced. Implementation details and information about used third party software will close this chapter.

4.1. Exisiting Frameworks

Which AR framework suits your needs depends on several factors. A major factor is the tracking method – whether marker or marker less tracking for example – since most frameworks focus on a limited set. Considering our experience with AR Toolkit (ART) and the target setup in C5 we chose a marker based tracking approach as the point of departure. However, expendable and exchangeable tracking mechanisms were considered as an essential benefit.

Laforge

LAFORGE is the visual subsystem of the recent C5 setup and implemented in C++. For image processing the Image Component Library (ICL) is used (see Section 4.3.1). Tracking is realized with ART by Kato and Billinghurst [BK02]. To enable video processing through ICL, ART's API was wrapped. Augmentation is done via OpenGL. Interprocess communication and controlling is done with Extensible Markup Language (XML) packages (called Taibaks) which are exchanged via the XML enabled Communication Framework (XCF). Besides the ability to import 3D-meshes in OBJ-format,

[1] compared to the previous mentioned frameworks

LAFORGE is capable of drawing several primitives. Created objects can be highlighted and continuously transformed, moved, rotated and scaled.

OgreAR

While researching possible 3D engines to outsource rendering and scene management to, we stumbled over OgreAR by EDM Studio [Edm]. They developed an AR game and released their core integration of AR Toolkit Plus (ARTKP) into Ogre under that name. It is limited to usage with Point Grey Research FireWire cameras. Additionally, it depends on Microsoft Windows thread libraries which make it platform dependent.

osgART

ARTOOLWORKS' osgART is a cross-platform C++ library for AR application development and rendering. As the name implies it uses ART for tracking which ARTOOLWORKS maintains as well [ART]. Rendering, event handling, scene management and object loading is done with OpenSceneGraph (OSG). A plug-in architecture offers the ability to exchange video capturing and tracking methods if necessary. Animation and other high level features like scaling, moving or particle effects are offered for authoring purposes. It is also available with an improved tracking architecture but only under a commercial license.

Instant Reality

Instant Reality was supposed to replace LAFORGE a while ago. It's a platform independent Mixed Reality (MR) framework developed by Fraunhofer IGD. They aim for a single and consistent interface which covers all components used in a variety of MR scenarios. It extends the Extensible 3D (X3D) standard and makes use of several open source libraries like OpenSG[2], Avalon or VisionLib [Fra]. The vision module supports several visual tracking methods and is capable of combining them for better performance.

[2]OpenScenegraph (OSG) and OpenSG are two different open source scene graph projects

4.1.1. Evaluation

OgreAR does not aim for a generic usage and can rather be seen as some *prove of concept* sample code that shows how to connect Ogre and ART. As mentioned in the Ogre forum this was the author's only goal [Fut07].

osgART aims for fast prototyping and encourage scientific usage. In that manner it was the most promising framework reviewed. However, without intensive testing it was hard to tell how flexible the open source version really is. Unfortunately, ART's lack of recent Video4Linux (V4L)[3] support made testing impossible. These issues were already known back when LAFORGE was developed. An ART wrapper for ICL called ICLART was written but lacked compatibility with osgART (osgART). When a GStreamer[4] workaround with V4L2 in compatibility mode was achieved, evaluation phase was already finished and work on the reimplementation of LAFORGE initiated.

In autumn 2009 Instant Reality was reviewed by student assistant David Fleer for usage in the C5 project. Unfortunately he experienced several issues:

- broken linkage (could be fixed manually)
- missing camera configuration tools in stable Linux releases
- incomplete Java API (e. g. Transformer node)
- licensing related bug in tracker module causes slowdown after 3000 frames
- daily build incompatible to used operating system (Ubuntu 8.10) because of conflicting libavcodec versions
- tracking sensitivity causes
- detection and operation issues with IEEE1394 (FireWire) cameras
- tracker configuration difficult; causes high rate of false negatives or false positives

Concerning these results Instant Reality was considered as not suitable for C5's setup. However in January 2011, Instant Reality released a new stable version 2.0 which unfortunately was too late to be evaluated in this work.

During the first phase of this work, LAFORGE was reviewed. To clarify the domain and context of LAFORGE, an environment modeling session was done with the recent C5 developer team. Hardware and software used was documented in a Unified Markup Language (UML) diagram to get an overview of requirements connected to LAFORGE

[3] http://linuxtv.org/wiki/index.php/Main_Page
[4] http://gstreamer.freedesktop.org

and its successor. Afterward, the most promising steps necessary to improve LAFORGE's functionality were collected in a focus group session. The insights were technical in nature and evaluated regarding the concept work of Chapter 3.

Three steps were proposed:

- Replace ART with ARTKP
  ART was intended to be a all-in-one solution for prototyping of AR applications. This includes in- and output of video streams. ARTKP focuses on tracking which is more suitable for an application that does not depend on image processing features of ART.

- Including a top/bottom camera for better tracking stability
  As for the reasons mentioned in 3.2 a third camera seems to be a feasible extension to the Phaenomeum setup.

- Replace native OpenGL support with a full features 3D engine
  Using LAFORGE, one experience a drop of the frames per seconds (fps) if many objects are shown. Additionally object support is limited. A powerful render engine might solve these issues.

In the next stage, code review was conducted and missing documentation like UML class diagrams were created. Findings were compared to the recent API and refactoring scenarios evaluated resulting in the following refactoring draft.

LAFORGE in its recent stage lags several features necessary to extend its functionally according to design phase results:

- **animation support**
  LAFORGE parses OBJ models and manage them on vertex level. Unfortunately those models do not prevent arbitrariness or incomplete model files. An animated model support would require extending the low level model management or outsourcing it.

- **scene management and control**
  Object management is redundant and spread over three different lists. Keeping these lists state sane results in much code overhead. On the other hand list access was not transparent to other functions which makes interfacing time consuming.

- **maintainable data model parsing**
  XML parsing is managed in a so-called *god object* [BMM98]. God objects do not have a limited field of responsibility and are used for many differing tasks. This makes them often quite complex and hard to maintain. The used hash functions require double-checking of given string commands which extends the used if-then-else structure even further. The parsing method itself lacks vital documentation. Data types and commands are documented in the API but some

features stated there are not implemented. Extending LAFORGE's parsing process in its recent state will take a lot effort. We propose a reimplementation of this process.

- **code structure**
  File names do not represent classes included. Separation between definition and declaration was not divided into header and source files. Usage of global variables should be removed and access managed by controller classes. To ease access and maintainability a structural refactoring should be conducted to solve these issues.

- **dependencies**
  Dependencies on external libraries are not limited to discrete parts of the system. Thus, removing one library breaks many classes and causes extensive adaption. Replacing the tracking and rendering infrastructure would require major parts of the system to be rewritten.

Considering all these issues we decided to rewrite the application from scratch. However, many design decisions and concepts were transferred since they were proven useful during years of work with LAFORGE.

4.2. Laforge 3

LAFORGE 3 is written in C++ and makes use of several third party open source projects. Thanks to ICL it supports a variety of video capture devices like USB web cams and IEEE-1394-Standard devices. Ogre game engine is used for rendering as well as scene management. Network communication and interfaces are realized with XCF. XML data type representations and parsing are handled with Bielefeld Type Library (BTL) and XML Template I/O (XMLTIO).

Software design goals were a robust, low latency tracking system for comfortable user experience as well as concerning post recording issues to support data evaluation. In addition all components were designed and evaluated to be extensible and replaceable to achieve flexibility and encourage authoring with just moderate c++ coding skills. It is supposed to be a basic framework which will be maintained and improved during the second funding phase of C5.

It was developed specially for the needs of the already mentioned C5 project and the involved task planning scenario Phaenomeum (Phaenomeum). Unique features are 1 to n cooperative tracking support, service interfaces and network based communication.

4.3. Hard- and Software Environment

As mentioned in 2.1 we use two HMDs manufactured by Triviso. Their ARvision-3D model8 was customized to work with Point Grey Firefly CMOS FireWire (IEEE 1394) cameras. With a resolution of 640x480 pixels they provide 60 fps of uncompressed video stream. We use one camera from each HMD but theoretically stereo vision is supported. A third Point Grey Firefly camera was mounted to a frame on the desk to provide top view video data. The HMDs contains a display for each eye with a resolution of 800x600 but are operated with a 640x480 resolution to prevent video stream interpolation and to achieve optimal frame rates.

Data is processed by three identical DELL T3500 Westmere with Intel Xeon E5640 and 6GB(6x1GB) 1333MHz DDR3 ECC-UDIMM. Passive Nvidia 9500GT with dual DVI graphic cards from Zotac are used to reduce noise level[5]. The cameras are connected via Point Grey dual bus FireWire cards (FWB-PCIE-02)[6].

All three machines operate with 32 bit version of Ubuntu version 10.04 (Lucid Lynx). Productive system and development environment are separated to limit interference. Changes are conducted on one machine and applied to the other two via Unison and package lists.

4.3.1. 3rd Party Software

Bielefeld Type Library

The department of applied computer science of Bielefeld University developed a library with common exchanged data types. The goal was to reduce redundancy and sources of error for application using XML for communication purposes. Parsing and parameter access are defined within the types to ease usage and clarify source code. Test suite best practices are provided and examples show how to use and adapt basic types for custom usage. The library is available in Java and C++.

Image Component Library

Written in C++ the ICL is a novel cross-platform computer-vision library developed and maintained in the neuroinformatics group of Bielefeld University and CITEC. Its development goals are to provide high performance and user friendly easy to use

[5] Usage of passive GPUs is planned but have not been available during this work. We used active 9500GTs with VGA and DVI interfaces

[6] Planned as well. DELOCK PCI Express FireWire cards were used within the thesis' tests

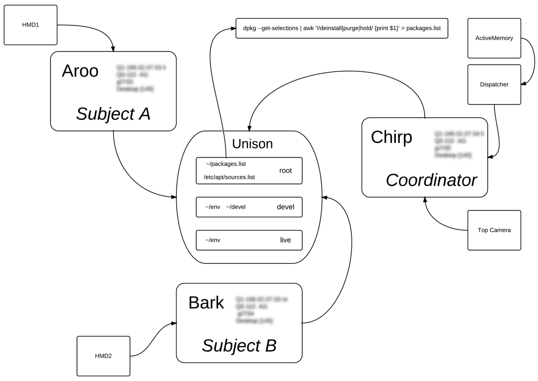

Figure 4.1.: Software environment. The environment is spread over three machines (called Aroo, Bark and Chirp). System maintenance, developing and release environments are separated and synchronized via Unison. If system wide software has to be installed, it is done on one machine. Afterward, a package list is generated and published to the other machines for updating.

classes. For example if the Intel IPP-Library is available processing speed can be increased for all CPUs supporting it. Users familiar with OpenCV can build ICL with corresponding support. XCF and Qt can be used as well for operation purposes. We use image grabbing and conversion features of ICL with OpenCV support for a wide range of supported hardware.

Ogre

The Object-oriented Graphics Rendering Engine is a cross-platform open source engine written in C++ for developing applications utilizing hardware-accelerated 3D graphics. Direct3D and OpenGL features are abstracted to offer high level access and operate interfaces based on world objects. It is purely designed for rendering issues and does not come with any physic engine or mathematical optimizations. However, the plug-in architecture offers additional features like particle support or collision detection.

XML enabled Communication Framework

XCF is a toolkit to operate distributed systems [WFB+04]. It follows the concept of information-driven integration to decouple parts of complex software systems and reduces dependency. It provides publish/subscribe and remote procedure call features. Blackboard features are provided by the Active Memory module. Active Memory not just only offers blackboard features, it records changes of elements over time. These logs can be replayed for scientific analysis purposes as well as application debugging or time sensitive applications.

Messages can be sent in XML or binary formats. Dispatching and database management is done by XCF and do not need to be considered during application development. XCF and Active Memory are cross-platform and available for C++ and Java. Older version are available via Sourceforge[7] but most recent versions are developed withing the CITEC environment and not (yet) publicly available.

Boost

Boost[8] is a collection of open source libraries written in C++ build for portability and to increase coding efficiency. There are boost libraries for a variety of tasks ranging from memory management over efficient implementations of algorithms to thread managements or networking. We use boost for thread and pointer management and program option parsing. Additionally, we use its circular buffer implementation.

4.4. Concept and Software Design

After the initial requirement elicitation and design phase, software design and implementation were done in iteration cycles. Features were reviewed and ordered according to their importance in the system. This list was used to create subsystems with limited feature sets. During a cycle software architecture was designed and implemented. At the end of every cycle a working system was deployed. Deployment could not be done on the target system though since hardware and software environment were just available at a late state of this thesis.

[7] http://xcf.sourceforge.net
[8] http://www.boost.org

The following increments were created:

1. *04.02.2011*: video stream augmented via local tracking data
2. *17.02.2011*: marker motion detection and basic communication framework
3. *15.03.2011*: custom models and calibration controlling
4. *23.03.2011*: distance constraint prototype implemented
5. *06.04.2011*: unified BTL parsing system
6. *19.04.2011*: local and global system states
7. *25.04.2011*: attention focus prototypes (glowing border, shadows)

Since focus was on features of the current cycle, refactoring had to be done every cycle to prevent architecture inconsistency. However, short cycles were intend to prevent *overcomplification* and *analysis paralysis*. Some of the most relevant parts of the final architecture will be introduced in the following.

LAFORGE 3 was designed and implemented object oriented. Software design concepts and classes will be written in `typewriter style` and can be considered as parts of the system with certain competencies and a proper noun.

4.4.1. Control Communication

The system is controlled via Remote Procedure Calls (RPCs) which are basically just application functions triggered by messages received from other applications. Arguments and responses are passed as XML documents. Everybody familiar with XML messaging has experienced parsing issues sooner or later. Easy and generic patterns are not available. The ambiguous nature of string messages and the lack of high-performance string comparison methods usually cause growing *if-then-else-if*-constructions which are difficult to maintain when they have reached a certain size. We tackled this issue with the help of the Bielefeld Type Library which we extended with custom data types. The bigger idea is to keep parsing mechanisms transparent and just define data structure conversation from and to XML representation of a type. BTL classes encapsulate the representation so that classes which want to use data types can use it easily. The workflow is reduced to initializing a template factory with the required data type, passing XML-strings to it and receiving the desired data object. Thanks to its object oriented design reuse of common data types is encouraged and sources of error are reduced.

To add new remotely operated functionality to LAFORGE 3, methods are registered at the RPC server. The server will forward the incoming message to the registered

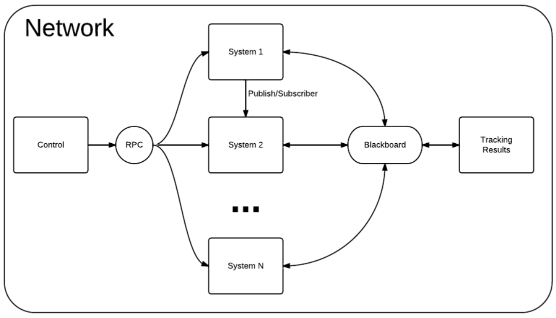

Figure 4.2.: Communication Architecture. An instance of LAFORGE 3 can be controlled by any number of remote applications. To avoid ID collisions Mertes' concept of `UniqIDs` used in the original version of LAFORGE was adapted [Mer08].

method. However, this architecture requires methods to use just a few data types. If methods are supposed to process more kinds of XML messages, a pre-parsing has to clarify the data type before the message is passed to the factory. If such a case appears, one should review the design decision and merge different data types by introducing optional attributes and elements for example.

Implementation

RPC is provided by XCF. XML parsing is done with the help of XMLTIO. Both pieces of software were developed and are maintained at Bielefeld University. To connect controller classes directly to their RPCs, *boost bindings* were used.

4.4.2. Cooperative Tracking

Albeit tracking mechanisms have not been improved, they can be combined to compensate for temporal untrackable markers due to perspective or obstacles reasons. Every instance of LAFORGE is publishing its tracking information to a blackboard. By default all instances monitor this blackboard and initialize so-called `Remote Tracking Services`[9] for each tracker which publish its information to the board. If a tracker updates its tracking information on the blackboard all related remote trackers make these information available in their local systems. A `Tracking Processor` chooses, merges

[9]In contrast to `Local Tracking Services`

or decline information. Local and remote tracking service use the same interface to prevent tracking states' impact on other parts of the system.

Instances need to share a global world reference system to use these information. Static LAFORGE 3 instances – instances connected to an always immobile and calibrated tracker – can provide information about present markers in world coordinates. If such a tracker detects no movement of a marker, it announces the marker as static. Other LAFORGE 3 instances use these information to detect their own calibration status and estimate their position in world space. If decalibration occurs, a system can recalibrate through comparing local tracking information with global position data.

To be able to detect motion, tracking information has to be stored and evaluated. Tracking memories work as a gateway between tracking and requesting services. For every tracked marker tracking time, position, velocity and acceleration data are stored for a configurable amount of time. The size of the memory is configurable but it is supposed to just store tracking information of fractures of a second since more information is not needed. Besides storing position and orientation data, the memory also calculates current speed and acceleration. Motion detection is done via comparing all velocity vectors in the memory. The vectors are examined in every dimension separately. A simple signum function reviews all velocity values' signs recorded in the specified time frame. If signs do not change within this period, we assume movement in this dimension. Or the other way around: If velocity vectors change direction in all dimensions in the examined time period, we assume tracking jitter and a static marker.

$$
\begin{aligned}
x, y, z, n &\in \mathbb{Z} & v_i &= \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} \\
x &= \sum_{n=0}^{N-1} \text{sgn}(\vec{v}_1^n) & y &= \sum_{n=0}^{N-1} \text{sgn}(\vec{v}_2^n) \\
z &= \sum_{n=0}^{N-1} \text{sgn}(\vec{v}_3^n) & H(i) &= \begin{cases} true, & \text{if } |i| < N-1 \\ false, & \text{else} \end{cases}
\end{aligned}
\tag{4.1}
$$

`Tracking Service Provider` return position and orientation of a marker in world space. These information can be merged into a transformation ${}^{G}T_{M}$ which transforms the marker origin

However, to deliver global data they have to know their position and orientation in world space. Calibration is done by defining one marker as global space origin.

To process of retrieving the marker position ${}^{G}p_{0M}$ in world space can be described as

$$ {}^{G}p_{0M} = {}^{G}T_{M} \cdot {}^{M}p_{0M} \tag{4.2} $$

with

$$ {}^{G}T_{M} = {}^{G}T_{T} \cdot {}^{T}T_{M} \tag{4.3} $$

where ${}^{T}T_{M}$ is the actual tracking result and ${}^{G}T_{T}$ is used to convert local results from tracker space into world space. If we assume ${}^{G}p_{0M}$ to be the origin ${}^{G}p_{0}$ of world space we can use ${}^{M}T_{T}$ which is $({}^{T}T_{M})^{-1}$ to find out the position ${}^{G}p_{0T}$ of the tracker since G and M will be identical.

$$ \begin{aligned} {}^{G}p_{0T} &= {}^{G}T_{T} \cdot {}^{T}p_{0T} \\ {}^{G}p_{0T} &= {}^{M}T_{T} \cdot {}^{T}p_{0T} \end{aligned} \tag{4.4} $$

where ${}^{G}T_{T}$ equals ${}^{M}T_{T}$.

We cannot get ${}^{T}T_{M}$ directly since it is just used inside the `Tracking Service Provider`. However we can retrieve current position and orientation of the tracker and reconstruct a ${}^{G}T_{T}^{*}$ to convert the delivered results back into tracker space.

The generalized form of this equation can be used when global tracking information of reference markers are available and consequential ${}^{G}T_{M}$. In combination with the currently used local ${}^{G}T_{M}^{*}$ we can adapt the position of the local tracker.

$$ \begin{aligned} {}^{G}T_{T} &= {}^{G}T_{M} \cdot {}^{M}T_{T} \\ &= {}^{G}T_{M} \cdot ({}^{T}T_{G}^{*} \cdot {}^{G}T_{M}^{*})^{-1} \\ &= {}^{G}T_{M} \cdot (({}^{G}T_{T}^{*})^{-1} \cdot {}^{G}T_{M}^{*})^{-1} \end{aligned} \tag{4.5} $$

Implementation

We use Active Memory as a central data storage engine. Tracking memories are implemented via boost circular buffer for improved performance and memory efficiency. If new data is supposed to be stored and the the memory is already full, the oldest data will be overwritten and do not have to be removed at first.

4.4.3. Service Interfaces

In order to achieve flexibility for further development, features are implemented through interfaces. Interfacing is a common method in object-oriented design to decouple parts of a software system. LAFORGE 3 offers such interfaces for its image processing, communication, tracking, scene management and rendering support. If necessary a service provider can be exchanged or extended without major changes in other parts of the software system. For example it is possible to replace a marker based tracking with a marker less tracking as long as position and orientation can be provided. Another possible scenario is the exchange of the scene management to optimize performance.

Implementation

Image services are implemented with the ICL. Scene management and rendering services use Ogre. The implemented tracking service uses ARTKP. As mentioned earlier, the communication infrastructure is provided by XCF.

ARTKP uses a right handed coordinate system but returns values which assume a different camera position in world space. Additionally, the marker coordinate system is rotated differently as well. To receive ${}^{G}T_{M}$ the ARTKP's matrix has to be rotated 180 degrees around the X-axis.

$$\begin{aligned} {}^{G}T_{M} &= {}^{G}T_{T} \cdot {}^{T}T_{M} \\ {}^{G}T_{M} &= {}^{G}T_{T} \cdot {}^{T}T_{ART} \cdot {}^{ART}T_{M} \end{aligned} \tag{4.6}$$

where

$${}^{T}T_{ART} = R_x(180°) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(180°) & -\sin(180°) & 0 \\ 0 & \sin(180°) & \cos(180°) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{4.7}$$

The matrix is used to extract position data as a 3 dimensional vector and orientation data as quaternions. To deal with the differing marker system rotation quaternions have to be rotated 90 degrees another time around the X-axis.

$$\begin{aligned} q &= (w, x, y, z) \\ {}^{G}q_{M} &= q_x(90°) \cdot {}^{ART}q_{M} \\ &= (\sqrt{0.5}, \sqrt{0.5}, 0, 0) \cdot {}^{ART}q_{M} \end{aligned} \tag{4.8}$$

4.4.4. Exhibition Models

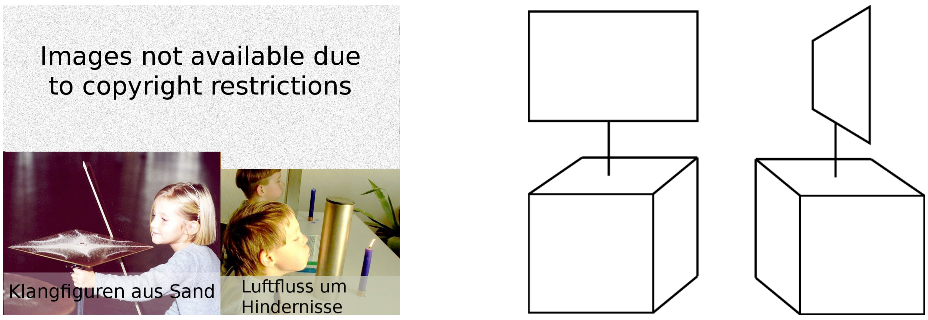

Figure 4.3.: Pictures of Phänomenta exhibits[10]. In previous studies these pictures were placed on both sides of signs (right sketch). However, visibility decreased when these objects were not positioned ideally.

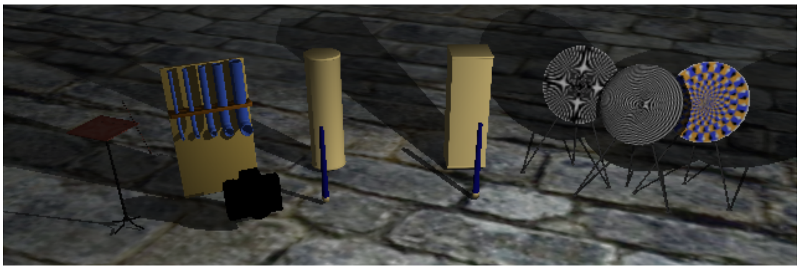

Figure 4.4.: 3D-models created from the Phänomenta exhibit illustrations seen in Figure 4.3. The image is a screenshot of the scene viewer, a test application written to analyze ogre meshes and dotScene files generated by Blender.

In former Phaenomeum experiments pictures of the Phänomenta exhibits were put on paper signs or equivalent augmented placeholders. Footage study showed that these pictures were hard to distinguish because orientation and distance influenced visibility significantly (see Figure 4.3). Thus, we decided to introduce 3D-models which were created by Richard Potter[11] according to the images used previously.

[10]Photographs by Phänomenia Erfahrungsfeld Essen and Phänomenta Peenemünde
[11]http://www.archaeodesign.com

Model integration workflow

Model design was done in Lightwave (version 9.6). To make 3D-models available in Ogre they have to be converted into Ogre's binary mesh format. Converting Lightwave models into Ogre meshes required Blender (2.5x series was used)[12] to read these models and convert them into an XML representation of meshes which can be converted into binary meshes with Ogre's XMLConverter. To be capable of doing so, Blender needs a community plug-in written in Python[13]. Unfortunately Blender had some issues with Lightwave's file format. Because of that Lightwave's OBJ export feature was used.

To convert Lightwave files into Ogre meshes, the following steps were necessary:

1. export Lightwave models as OBJ files
2. load files into Blender
3. check Textures and UV Mapping; adapt mapping settings
4. export Ogre Scene
5. convert XML meshes into binary models with OgreXMLConverter
6. add export path to Ogre's resource.cfg

Objects and dotScene files generated by Blender can be tested with the test application scene viewer which is also a part of the LAFORGE 3 framework.

4.4.5. Software Architecture

In the following we will give an introduction to key features of LAFORGE 3. We indent to keep the description simple to allow understanding without profound software engineering knowledge. Class diagrams for more detailed information can be found in appendix C.

Tracking Workflow

We decided to detach image grabbing, tracking, model updating and rendering to prevent latency if one of these core features requires some extra time. `Tracking`

[12] http://www.blender.org
[13] https://code.google.com/p/blendogre

Service Provider uses the *Observer Pattern* to indicate new tracking results. Observers interested in tracking results can register themselves at the provider. This event triggered workflow is used by classes called Tracking Publisher and Calibration Controller to operate independently from the system's model update cycle.

The core system does not use tracking triggered updates though. Since LAFORGE 3 is supposed to work with multiple tracker instance, it either has to update at every received signal received, favour the local tracking instance or operate independently. We have chosen the third option to minimize update effort and easier manipulation of update cycles. A simplified activity diagram can be seen in Figure 4.5

Marker Model and Control

Markers are key positions within the scene. Usually their position and orientation is obtained from tracking services. They are managed by a Marker Model which acts as an Information Holder. To reduce *false positive* tracking, LAFORGE 3's tracking controller just requests tracking information for markers present in the model. Markers can be observed if status changes are required.

Entity Model and Control

Entities are objects present in the scene. The most basic object is an abstract Scene Element which just contains a position and orientation in space and a UniqID for element management. Visual Elements extend this concept with information about their visual representation, called Renderable but still stay abstract. Simple World Objects realize Visual Elements and can be considered as static objects in space. As the name implies, Simple Marker Objects add information about related marker to this concept. They are not limited to one exclusive marker even though this is the most common case. Properties of Simple Marker Objects can be used for offsets towards related marker coordinates.

Entities are collected by a Scene Model. The model itself is an *Information Holder* and does not offer any other functionality. Laforge Entity Handler monitors the entity states and conduct changes if their state changes.

A simple scene viewer was written to verify the model export. This viewer uses the dotScene file generated by Blender to recreated the scene as it were exported from Blender. Importing other model formats like the native formats of 3D Studio Max, Maya or Softimage/XSI (all Autodesk[14]) can be done directly with the help of Ogre-Max [Ogr].

[14]http://www.autodesk.com

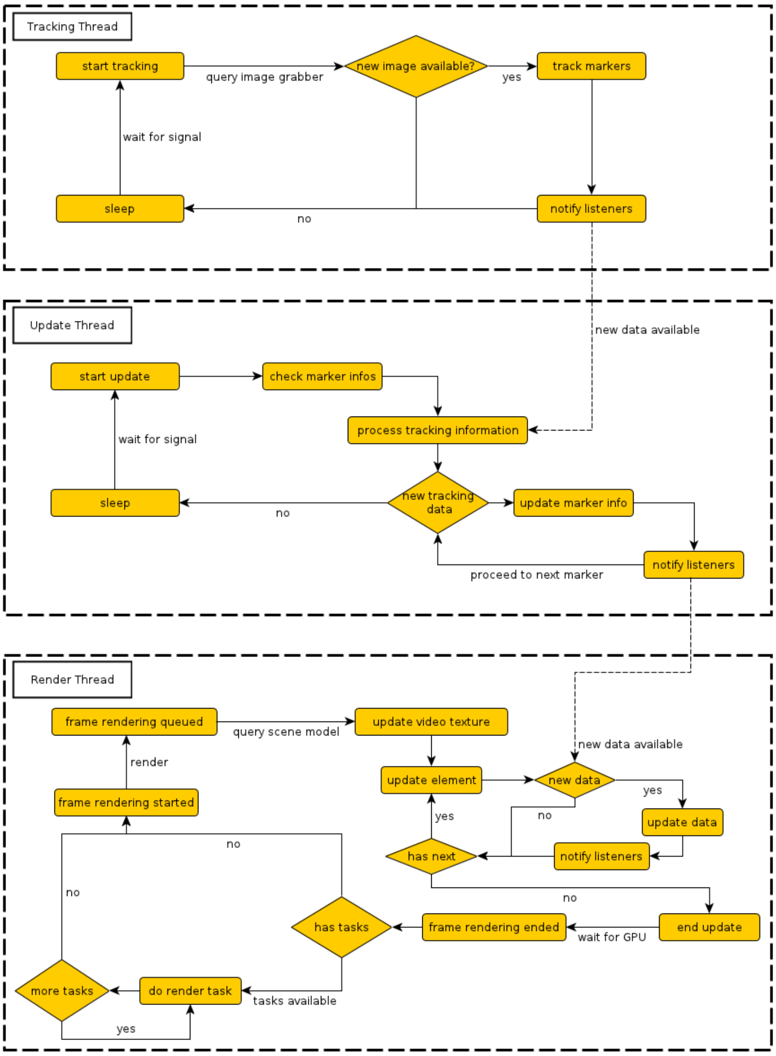

Figure 4.5.: Tracking Workflow. Tracking, updating and rendering is done in independent threads to minimize latency caused my one of the three core services.

4.4.6. Code structure and guidelines

The package structure is shown in Figure 4.6. Classes are organized in packages according to their task within the system. the package `btl` holds all data type definitions with an XML representation for network communication. Communication is realized by classes within the `communication` package. Classes depend on parsing mechanisms and XCF. `Vision` keeps image grabbing related interfaces and ICL using classes. The package `tracking` contains controller and model classes for tracking and calibration. Those four packages contain the core features. `Model` includes the system model for markers and entities whose interface declarations and several definitions can be found in `entity`. The `scene` package holds interfaces for rendering and scene control support. In the sub-package `ogre` one will find the definitions of these interfaces. Packages `constraint`, `participant` and `visitor` keep three sample implementations of how to use the basic features of LAFORGE 3.

Telling a coder how to write her code in every telling is like dictating an author how to write his novel. At least that is how some developers feel about coding guide lines. Code style has no direct influence in application performance but it is an essential factor when it comes to maintainability of large software projects. If code is written by a single person or a small group, developers tend to rely on their *best practices* instead of agreeing on some standards. Defining such standards can take a serious amount of time since there is no accepted worldwide standard and preferences often correlate with personal experience and professional background. Usually that works well enough for short-time projects but if code has to be reviewed by developers who are not directly involved problems occur. Sometimes it is not even necessary that another person than the developer does the review, time alone can convert well understood code into hieroglyphs.

To avoid such phenomena two guidelines were reviewed and considered during development. A well known guide for good code style was written by Steve McConnell and is called *Code Complete* [McC04]. The second guide is the *Google C++ Style Guide* based on the company's experience with open source c++ projects [WSE+10]. Even just comparing these two reveals how much one's definition of good code style correlates with personal preference since they argue contrary to each other in some issues.

We will not mention all the proposals we followed but the ones we experienced to be helpful and made coding and reviewing easier to conduct.

- **Variable names**
  Variables are all written in small letters and words are separated with an underline character. Names represent the intended usage to ease understanding. To avoid collision with local variables, member variables use a prefix 'm'. A prefix 'p' indicates that the given variable is a pointer since pointer and class access are handled differently in C++.

- **Typedefs**
  Following McConnell's suggestion primitive and often used types are redefined and therefore abstract within the program. For example `Position` and `Orientation` are realized with Ogre `Vector3` and `Quaternion` classes. To implement another analysis library (e.g. with better performance) those two types just have to be redefined and no further code adaptions have to be done. However, this a theoretical assumption. If the library differs in operator usage or constants are named differently, a wrapper has to be written or the affected code to be adapted. Thanks to typedefs the use of *boost smart pointers* was encapsulated for exchangeability and type names were shorten for better readability.

- **Other naming policies**
  Methods and constants are written in Camel case. Constants have a prefix *k* without an underscore. Many developers learn to note constants in capital letters. Weinberger et al. argue that this might lead to confusion in C++ since macros are noted like that as well. If all variables are written in small letters only, constants still stand out and – as another advantage we experienced – the leading 'k' makes constant access from IDE's auto completion features easy.

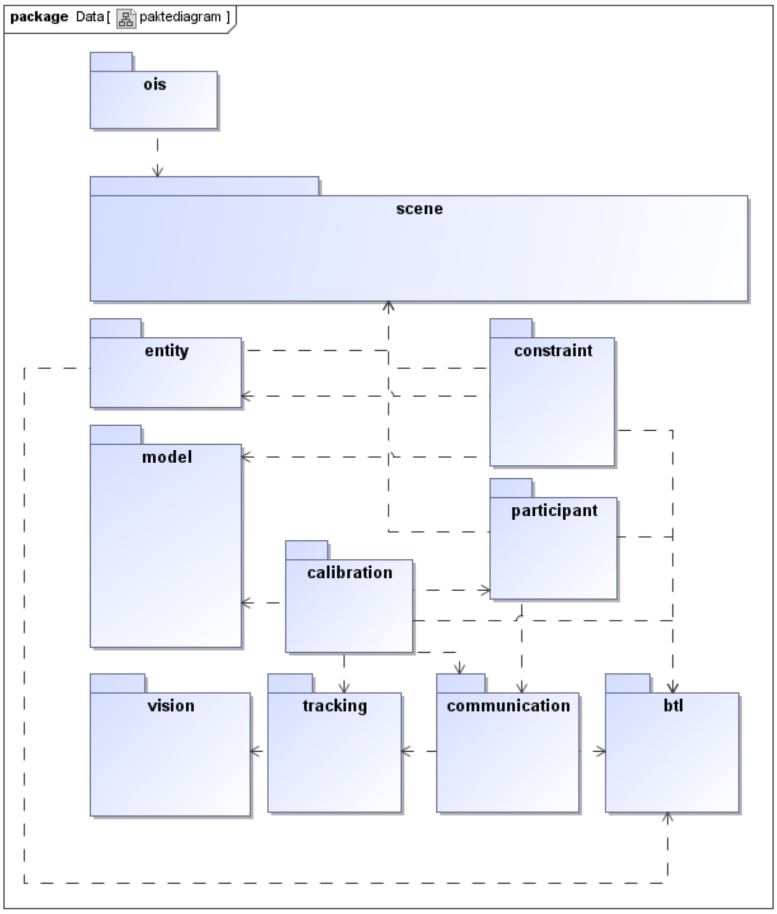

Figure 4.6.: LAFORGE 3 package structure. Dashed lines show usage relations between the packages.

5. Evaluation

LAFORGE 3's core features such as basic tracking capability, communication framework and cooperative tracking were tested on the system described in Section 4.3. Besides the *attention request* feature, all visual augmentation prototypes introduced in Section 3.3 were implemented and could be evaluated. Related changes and extension will be introduced to give insights about how the framework is supposed to be adapted to specific needs.

Prototype evaluation was done in two steps. Firstly, performance was tested with simulated input which means that controlled and optimized values were published to the Active Memory blackboard or directly to the LAFORGE 3 instance. The goal was to gain insights about the best possible performance of the conducted implementation. In cases where input data can differ in regard to accuracy and frequency, we conducted tests with multiple LAFORGE 3 instances on distributed systems. The setup shown in Section 2.8 was used with the camera mounted over the table and one HMD. This was also meant to test the communication infrastructure and its performance under realistic usage circumstances.

This chapter starts with a comparison of LAFORGE 3 and LAFORGE concerning fps and latency. Some findings and impression lack communicability through static images though. Footage recorded during technical tests will be provided on a data carrier with this thesis. All footage is located in the `footage` folder.[1].

5.1. Performance

Both systems share basic features of video streaming and augmentation which will be reviewed in the following. Additionally the cooperative tracking feature of Laforge 3 was tested under simulated and real world circumstances.

[1]Single-window videos are provided in Theora (ogv) format. Merged videos come in MPEG-4 (mp4) format. Videos were tested with the VideoLAN player (http://www.videolan.org/vlc) which is available for all common platforms.

5.1.1. Frame Rate

During review of LAFORGE we could observe dropping frame rates if many objects have to be rendered into the scene at the same time. These observations could be reproduced on the test system. As shown in Figure 5.1, frame rates varied from about 60 fps with no augmented objects present to 20 fps when the screen was filled with such objects. Over time the average frame rate was about 25 fps. LAFORGE 3 was not that heavily influenced by present augmentation objects. The frame rate lowered slightly but stayed at a level above 110 fps during the whole test period. Frame rate changes can be observed in the test videos l1_fps.ogv and l3_fps.ogv.

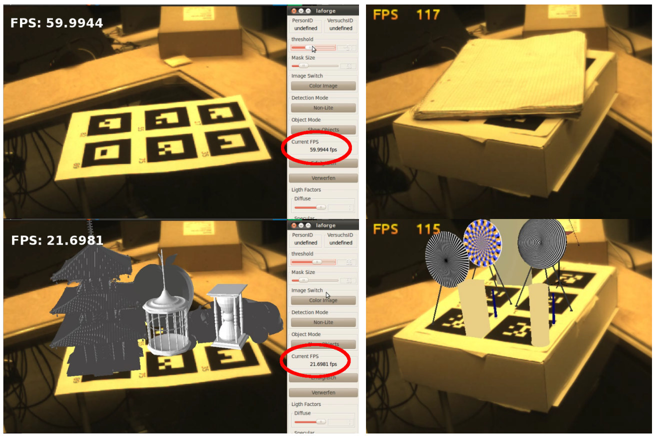

Figure 5.1.: Frame rate evaluation. Performance impact of virtual objects was tested. LAFORGE's fps drops from 60 fps to about 22 fps (left image). The frame rate of LAFORGE 3 is not influenced significantly and stays above 110 fps (right image).

5.1.2. Latency

One issue pointed out the during requirement elicitation phase (see Section 3.2) was latency. Footage of previous studies conducted with LAFORGE showed that most users get used to a small lag though. However, latency is a major issue when it comes to user experience and keeping up the illusion of virtual objects connected to real world objects.

Lagging objects occurred occasionally while testing LAFORGE but stayed within reasonable bounds. In contrast, LAFORGE 3's test run showed perceivable higher latency than the reference system. Figure 5.2 shows static results of these tests. However, for detailed insights the test videos l1_latency.ogv and l3_latency.ogv can be considered.

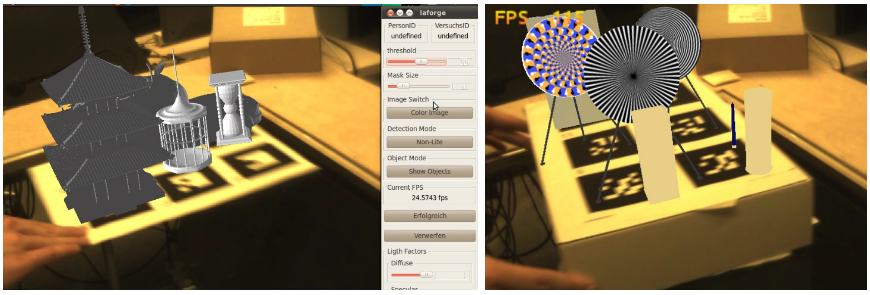

Figure 5.2.: Latency test. Both systems show latency as seen in the pictures. However, LAFORGE 3's latency is perceivable higher than the latency of LAFORGE.

5.1.3. Cooperative Tracking

calibration and cooperative tracking was evaluated in two different setups. First we checked the best possible adaption performance with controlled simulated location data. In the second phase we reviewed performance of the HMD instance when provided with less precise data of the top view instance.

Simulation

Decalibration detection and recalibration could be done without any visual evidence most of the time. Flickering could be observed occasionally but not frequently. Even though no statistical data were collected, one can rate the observed data as satisfying. Figure shows two sequential frames extracted from the footage file `13_calib.ogv`.

Distributed Systems

Under real conditions, decalibration detection and recalibration worked with less performance. While recalibration was done unobtrusively, decalibration and the resulting fallback to local data caused flickering objects. Recalibration success rate dropped significantly which made several attempts necessary depending on variance of data provided by the top view instances. Thus leading to slower adaptions rates and less fluent attention focus updates which can observed in the video file `13_coop.mp4`.

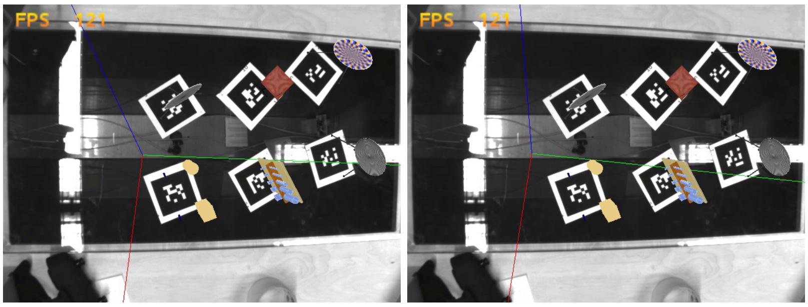

Figure 5.3.: Calibration simulation. To test recalibration performance, controlled information about one marker position was send to the blackboard. Pictures show frame 184 and 185 of the recorded video stream which can be found in the footage file `13_calib.ogv`. The coordinate system has changed and with it the position of all objects in the scene. This transformation stays invisible for the user though.

However, global tracking data usage was possible and could not be distinguished from locally tracked markers as shown in Figure 5.4. Even when no local data were available, object updating stayed smooth and latency did not increase significantly (see Figure 5.5). This required a calibrated, static HMD instance though.

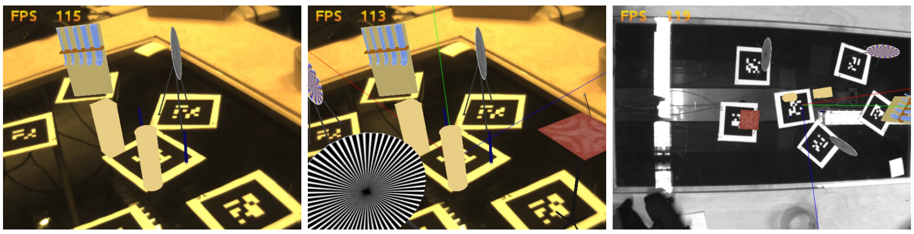

Figure 5.4.: Calibration evaluation 1. In the first picture the HMD instance is in an uncalibrated state. Markers which are not completely visible cannot be tracked. After a calibration of the static LAFORGE 3 instance (right image) information of the visible markers is used to augment former unavailable marker positions.

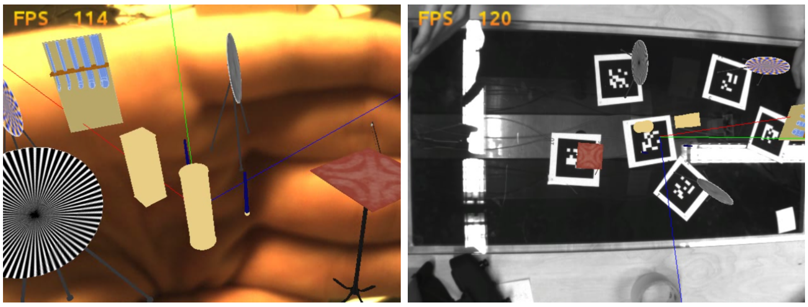

Figure 5.5.: Calibration evaluation 2. Even though no local tracking information is available when the camera is covered, the HMD instance still tracks the marker with only global information available from the blackboard.

5.2. Extension Examples

In most cases the implementation of the implemented prototypes did not differ much from the presented sketches. Only *Glowing Border* was adapted to a more feasible approach. The section will start with a short introduction of how animated objects can be used.

5.2.1. Animated Augmentation Objects

We extended the `Laforge Marker Object` to operate with `Animation States` provided by Ogre. The animation state can be changed according to the underlying entity state. The implemented `Static Change Object` makes use of the *idle* and walking state of Ogre's sample robot object. If a marker is announced as static, the robots stops walking. Since `Static Change Object` inherits the `Visual Element` interface of `Laforge Marker Objects` no further changes were required. Animation works flawlessly which was to be expected since it is one of the core features of the render engine we used. Figure 5.6 shows two sequential frames extracted from the recorded video named `13_animation.ogv`.

5.2.2. Gaze Information

To calculate the attention focus, instances required to be able to publish their current position and receive the position of participants they are interested in. Therefore the position of the local tracker – which is the view of a participant – is published if it changes. This was done by a `Participant Controller` which observe the local

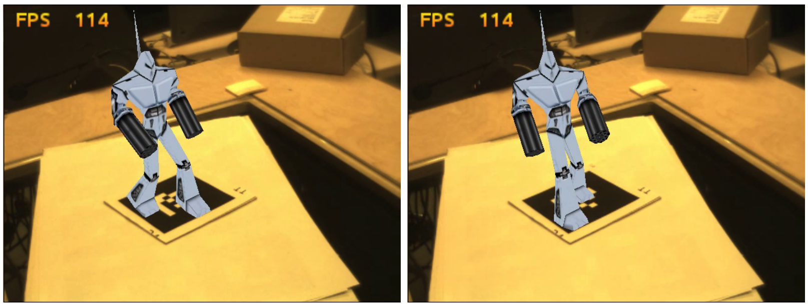

Figure 5.6.: Animated augmentations test. Images show frame 469 and 491 of the Footage 13_animation.ogv showing a different animation state of the robot object, a sample mesh provided by Ogre.

tracker and publish calibration results. To receive these information remote instances have to observe the specific instance what. Participant Controllers connect each other via publish/subscriber provided by XCF.

Model

If the users position or position of the interaction partner changes a raytrace is conducted. The Participant Controller uses interface methods defined by Scene Controller and declared by Ogre Scene Controller to check intersection of the opponents view ray with the extended table surface – the world's XZ-plane – and the XY-plane shifted to the position of the user. The intersection point closer to the remote participant will be passed to the Attention Focus Visualizer interface whose implementation will be introduced later. Due to the fact that visualization methods naturally depend on the techniques provided by the render engine, they are engine specific and have to be reimplemented if another engine should be used. This does not count for the Participant Controller though.

This approach was designed for the *Phaenomeum* setup where both subjects are facing each other (see Section 2.6). We assume that the shifted XY-plane of one user marks the end of the table space which is interesting for the other user. If she raises her gaze even more, she either looks behind her opponent or watches him directly. This is not true for all scenarios though. However, during the footage review of previously conducted studies we rarely observed a situation where one user put his head over the table while his opponent payed attention to an object located behind his shoulders.

Attention Focus Visualizers have to check which of the intersection points is provided. Therefore they have to evaluate if the intersection point is located on the local XY-plane which means it has a Z-coordinate value of zero. If both scenarios are handled differently depends on the implementation of the visualizer though. For testing purposes, the participants view ray is visualized by a yellow line starting at the participant's location and ending at the intersection point with one of the planes (see Figure 5.7).

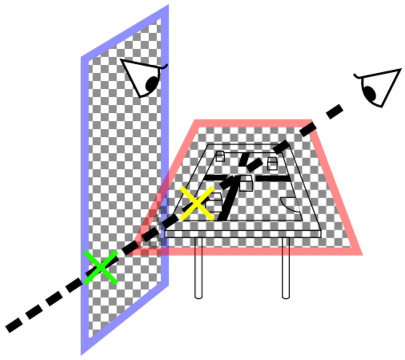

Figure 5.7.: Intersection model. A raytrace is conducted and intersection points of the participant with the table plane (red) and the user plane (blue) calculated. The point closer to the participant – the table intersection point (yellow) in this example – will be passed to the `Attention Focus Visualizer`.

This test case was not evaluated separately. However, this feature can be seen in footage of Section 5.2.3 and Section 5.2.4.

5.2.3. Light Augmentation

The implementation sticks to the *Ligh of Attention* prototype mentioned in Section 3.3. We use additive stencil shadows provided by Ogre and a spotlight to indicate the intersection point with the table plane. The table itself is covered with a transparent texture which does not cast but receives shadows. Without this texture shadows would be visible on other objects just.

We used stencil shadow instead of texture shadows since they are easier to calculate if there are only a few model details. However, the frame rate of LAFORGE 3 did not drop below 110 fps (see Figure 5.8) so that more expensive methods could be tried out in the future.

An observed drawback is caused by slightly jittering objects. The minor shaking becomes more visible with the shadows. This makes the ambience appear a bit restless. However, from a "It is not a bug, it is a feature!" point view, one can argue that the resulting ambience is closer to real candle light which produces "similar" restless light in an open environment. The experience also depends on the update rate of the participants' view rays. The simulation shows how smooth shadows can be updated if very accurate information is available (see Footage `13_light.ogv`). During tests on distributed systems however, shadows often `jump` which might reduces the benefit of this augmentation (see Footage `13_light.mp4`). This has to be investigated in further user studies.

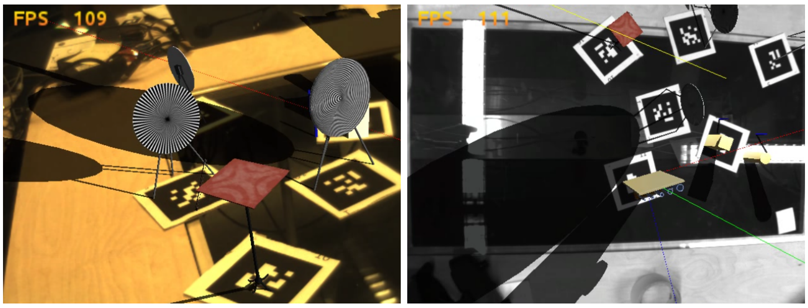

Figure 5.8.: Gaze direction augmentation with light. HMD and top view perceive the intersection point of table plane and opponents view ray as a light source. On the right side the yellow view ray of the HMD is visible. In user tests the will be removed though.

5.2.4. Glowing Border

The second `Attention Focus Visualizer` implementation varies slightly from the related prototype. Instead of an arc pointing into direction of the intersection point, arrows on each border of the screen are used. The intersection vector is projected onto the 2 dimensional screen and divided into X and Y components. Depending of the length of these resulting one dimensional vectors the arrows transparency changes (see Figure 5.9). The sign of the vector decides whether the left or right, or the top or bottom arrow is influenced.

We use an Ogre overlay script with four images located on each side of the view borders. The implemented class receives the image values via Ogre's material management and adapts transparency values if necessary. Images, position and dimensions of images can be modified within the script without the necessity of recompilation.

In contrast to the light augmentation described in the previous section, the glowing border is not that heavily influenced by the lack of available data. Update rates became slower as well but this incident was less obvious compared to the light augmentation (see Video `13_radar.mp4`).

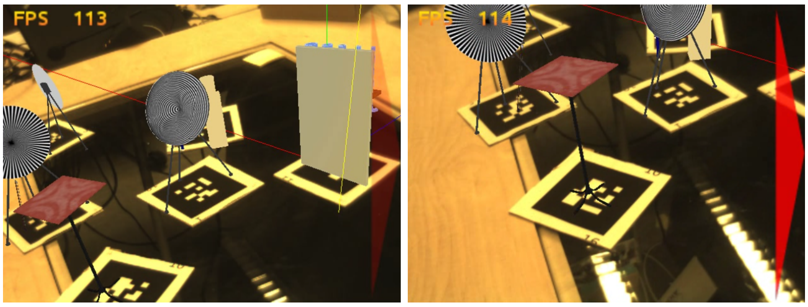

Figure 5.9.: Alternative view ray augmentation. Related to the distance of the view ray's 2d projection from the center of the user's view arrows appear on the view's border. An opaque arrow implies more distance to the view ray intersection point.

5.2.5. Constraint Visualization

A `Constraint Controller` was implemented and connected to the RPC server. LAFORGE 3's `Simple World Entity` was extended to support texture manipulation for transparency support (called `Alpha World Object`). Texture properties were retrieved from Ogre with the helper classes written by Kencho who made them accessible in the Ogre wiki[2]. Additionally primitive support was added to the `Ogre Scene Controller` via ogre-procedural library[3] but is not used due to texture mapping issues. However, if necessary, primitives can be added as entities as well as constraints or other types of properties related to an entity.

Figure 5.10.: Constraint evaluation. Two stages of constraint violations are shown from top view. The left picture shows minor constraints violations caused by multiple objects. On the right picture more severe constraints violations resulting in less transparency of radius disks are shown.

[2]http://www.ogre3d.org/tikiwiki/Per+renderable+transparency
[3]http://code.google.com/p/ogre-procedural

Adding constraints is quite simple: If a message is received, the Constraint Controller parses the properties, request loading of the mesh as a Visual Element and attach the created constraint Renderable to the Visual Entity specified.

The constraint class implemented is called Distance Constraint. To update the value – and the related alpha value – a set of other constraints is passed and evaluated. Depending on the closest passed constraint the violation value is adapted. If all constraints are out of range the alpha is set to 0 which makes the related visual element invisible.

The Calibration Controller monitors all constraints. If an entity constraint is moved or rather the object it is attached to, all counter constraints are updated. That means that if a light emitting entity is moved, all constraint omitting light will be updated.

Tests were conducted with *sound emitter* and *sound avoider* constraints (see Figure 5.10 and Footage l3_constraint.ogv).

5.2.6. Virtual Visitor

To show multiple render windows, multi-viewport support was implemented. Additionally binary render masks were introduced to obtain different render results even though all viewports operate with the same scene. Thus allow to show elements like the orientation grid in the visitor window but not in the main view.

The addVisitor method is called and the message passed is parsed by a Virtual Visitor Controller. This controller is an extension of the Laforge Entity Controller which handles synchronization of Scene Controller internal model and LAFORGE 3's scene model.

The controller passes an entity load request to its base class Laforge Entity Handler and retrieves the result afterward. The visual element is extended with a viewport and from there handled like an ordinary entity. When objects announce updating or removal Virtual Visitor Controller checks their UniqIDs and passes them to its base class. If the object connected to the viewport will be removed from the rendering scene – due to lack of recent tracking data for example – the visitor controller will remove the connected viewport as well. If new tracking data is available and the objects requests rendering again, a new viewport will be added. This makes activation or respectively deactivation of the visitor feature fairly easy. Intended usage is shown in Figure 5.11 and Footage l3_visitor.ogv.

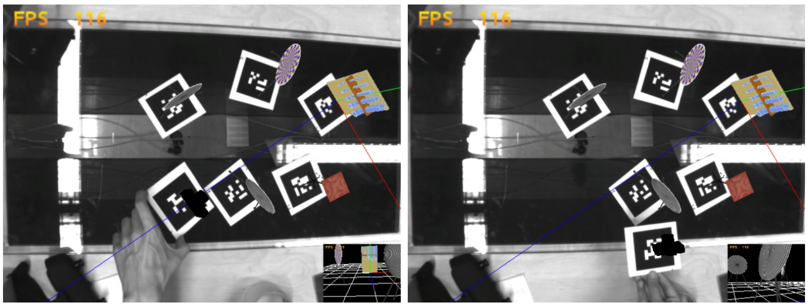

Figure 5.11.: .
Observing the scene from a different angle with a virtual visitor. The visitor's point of view is shown in the bottom right corner.

5.3. Discussion

The frame rate of LAFORGE 3 fulfills all our requirements due to the Ogre engine integrated with LAFORGE 3 being capable of more complex rendering tasks than those we have tested here. This frame rate just gives feedback about the rendering part of LAFORGE 3 though since rendering, processing and image grabbing were detached as mentioned in Section 4.2. Test runs of LAFORGE 3 with enabled logging showed that new tracking data requests triggered by frame rendering cycles occur two to three times for every frame grabbed from the camera. However, the latency observed during tests shows room for optimization of the image processing queue. Since rendering speed is not the problem here, a closer look to the the synchronization process of images and scene graph information is recommended.

Considering the naive approach of simple coordinate transformations for visual calibration, cooperative tracking performance was satisfying. Decalibration detection and recalibration algorithms are simple but can be improved to achieve better data in the future. Calculation speed was not an issue here since the amount of necessary calibration attempts limited its performance, not the duration of one attempt. However, if more complex approaches should be implemented, calculation performance can be improved by using optimized libraries like the Integrated Performance Primitives. This can probably be done fairly easily with redefinitions of the (4D) `Matrix`, `Position` and `Orientation` classes. However, at the moment those classes are defined by Ogre classes. Basic operators wont require changes but functions like `OgreMatrix4D`.*extractQuaternion()* are used directly without interfacing. If the usage of such libraries is planned for LAFORGE 3 we recommend refactoring in close temporal proximity.

Since animation support was one major reason to use Ogre, LAFORGE 3 offers animation features of 3D augmented objects. Compared to the other extensions it required the fewest adaptions to the existing system though. Implementing test extensions were a very helpful task to identify weaknesses and false or incomplete abstraction levels. However, further extensions will probably require additional changes to the framework. But we are quite optimistic that experience gained through continued work with LAFORGE 3 will help to define and distort responsibilities.

The visualization implementations could be done straight forward when the information communication architecture for participant position date was designed and implemented. But again we mostly have to credit Ogre since all features used were provided by the render engine. The abstraction however made it possible to focus on visualization prototyping instead of changing underlying structures. Implementing additional `Attention Focus Visualizer` – if they make use of known techniques – can be done in a short amount of time which satisfies the IxD approach of varying, fast deployed and interactive prototypes. Concerning that we consider this task to be successfully accomplished.

6. Outlook

Development of LAFORGE 3 was never intended to end after this work. We propose to lay a foundation for community driven development of LAFORGE 3 since resources inside C5 cannot focus on software development alone. The open source environment *OpenSource@CITEC* provided by Bielefeld University is suitable to host such a project. However, just publishing LAFORGE 3 wont be enough to encourage people to participate. Although the framework was designed for a specific context, it was also designed to extend this context if necessary. This might make LAFORGE 3 more interesting for related AR research in and outside of Bielefeld University. During this thesis we found several attempts to connect AR technology – especially ART related – to Ogre. We could observe several issues which occur quite regularly and which cannot be solved without serious effort. LAFORGE 3 solves some of these issues and might motivate people from within the Ogre scene or related communities to overcome starting issues. Feedback from such communities could help to strengthen the authoring focus of LAFORGE 3 and let people work on Bimber's second layer (see Section 2.2) without worrying about more basic mechanisms. Features could be evaluated faster since no user tests have to be organized. Advantages are not just limited to software development issues. Wrapping powerful libraries and making them accessible for developers with less coding experience might result in the creation of new and interesting augmentation strategies. OpenFrameworks[1] and Processing[2] are examples for how to make feature rich libraries available for a community from different backgrounds.

However, many obstacles have to be tackled upfront. User's feedback has to be enabled through services like public bug trackers (e.g. MANTIS[3]) or feedback environments (e.g. UserVoice[4]). Such features might be planned for OpenSource@CITEC anyway. Another issue to be solved is the availability of used but not openly accessible 3rd party applications. ICL is already a part of OpenSource@CITEC but current XCF and Active Memory (acme) development is not publicly available. BTL developers announced that they intend to take this step in the near future though. Stakeholder interests and plans as well as licensing issues might complicate the goal of making LAFORGE 3 available in its recent state. If so, some applications might have to be replaced by similar components.

[1] http://www.openframeworks.cc/
[2] http://processing.org
[3] http://www.mantisbt.org
[4] http://uservoice.com

During the design process we created UML class diagrams which were adapted during development (see Appendix C). Keeping them up to date consumed a considerable amount of time and might not be as necessary as it is in large group projects. However, thanks to that, design options could be played through with the big picture kept in mind. If other developers plan to work on LAFORGE 3 it probably will be a good introduction to the system as well. Creating such documents when they are needed the first time usually takes even more time as we experienced during our refactoring phase of the old LAFORGE system. Other kinds of architecture visualizations like sequence or activity diagrams might ease access to LAFORGE 3 even more. If this kind of documentation is considered to be useful to attract and support new developers in the future, documentation should be extended as soon as possible.

There is a variety of interesting and promising features and augmentation methods to be implemented as well. Sonification attempts had to be postponed but still promise to increase benefit of LAFORGE 3 in C5's context. The scene element interface can be extended with non visual features to place sound entities in the scene or directly attached to an object. Movement of objects could be sonified via scratching or grinding sounds. It could be distinguished between pushing or lifting object, shaking them or placing them. In combination with spatial information this acoustic interface indicates action taking place outside of the user's field of view. Integrating sounds into the virtual visitor feature creates the opportunity of indicating constraint violations through sound. A suitable audio interface has been already implemented by Till Bovermann which just has to be connected to LAFORGE 3. Users can place the visitor close to an object needing silence to listen for distracting noises from another exhibit affecting it. Additionally, the immersive effect is increased if the visitor also returns acoustic ambience feedback.

As mentioned in 4.4.2 the tracking of virtual objects is supported by `Tracking Memories` which represent a history of recorded frames over time. This feature could be extended. For instance, more sophisticated mechanisms could be implemented to increase object stability and prevent "unnatural" behavior of augmentations. The tracker engines sometimes detect markers in wrong spots, so called *false positives*. The distance to the *real* marker position is random and causes objects connected to the marker *jumping* around in the scene. With the help of the `Tracking Memory` we could monitor the velocity of a marker and identify some of these *false positives*. During update cycles the memory can decide if a position change indicated by the current tracking results is physically possible. For example, if we assume a tracking frame rate of about 30 fps and the tracked position of a markers differs for about 30 cm caused by *false positive* tracking results, that would imply a speed of about

$$\frac{30}{\frac{1}{30}}\frac{\text{cm}}{\text{s}} = 900\frac{\text{cm}}{\text{s}} = \underline{32.4\frac{\text{km}}{\text{h}}} \tag{6.1}$$

which is a very unlikely velocity for a marker in that context.

Another possibility is to estimate object positions and reduced lag caused by temporal blurry images. This can also be used to detect marker leaving the field of view to reduce false positive of similar markers since markers which had left the scene most probably won't show up one second later right in the middle of it. Even though more calculations require more computing time, the available hardware should be capable of handling such extensions.

Another interesting feature is a shared vision approach. Recent tests have shown that XCF is capable of transmitting video streams fluently. If performance tests succeed, the visual visitor window can be used to transmit the other subject's view to an instance of LAFORGE 3. The specific behavior is to be designed yet but the estimated difficulty is low since all steps necessary to achieve this goal were already done in other parts of the system.

The tracker concept could be extended with other marker types as well. Tracking mechanisms returning position just need to define the tracking service interface and can be used with the implemented models and controllers. Acceleration and orientation sensors need a bit more adaptation but it should be possible to implement within the tracking section of LAFORGE 3. In context with visual based marker tracking procedures such data can be used to predict movement of markers present in the picture. Such a combination of tracking data was already proposed by Azuma [Azu97] and was taken into account during software design.

Besides the addition of more sensing mechanisms the number of participating subjects can be varied as well. Instances of LAFORGE 3 operate independently but make usage of global data. That means any amount of instances can operate at the same time as long as they differ in names. However, a central operating unit is needed for syncing (or knowingly) desyncing the world models, entities and markers to be tracked. It was an intentional decision to not control model states of instances through the blackboard. The most simple way would be a script which gathers the functionality of all existing test applications. If a more sophisticated solution is desired User Interfaces can be developed with the help of any programming language or framework which supports XML message sending.

Extending the scenario to more than two participants creates new augmentation tasks. As mentioned earlier visual augmentation covering huge parts of the view are rather likely to either cover real world information or distract the user. In the Phaenomeum planning task scenario one could get more precise insights about how to let a big number of augmentation systems cooperate. Color as a distinction parameter should be avoided since GUI design best practices already mention the risks that come with parameters which might not be perceivable by all possible users. Sound connected to world coordinates might tackle this issue in a better way. User's spatial knowledge about the location of other participants can be used to tag actions less obtrusive than visual augmentation.

Extending LAFORGE 3 with a more powerful Heads-Up Display (HUD) support offers a wider operational area. It could provide guidance and provide introduction in traditional AR fields like assembly or orientation tasks. A point of departure could be a cooperation with Stefan Rüther who conducts research work related to AR assistance systems at Bielefeld University. The proximity ease communication which is a benefit one should not underestimate. There is a reason why agile project development proposes on-side development and personal meetings are important in business relations still even though modern communication technology would allow steady remote conferences.

An issue mentioned by Fischer et al. is the appearance of virtual objects [FHT08]. Rendered objects often stay out and do not integrate well into the underlying video stream. They used so-called *stylized augmentation* to alter both, video stream and virtual objects for an integrated experience. The resulting image makes it hard to distinguish between real and virtual objects which benefit immersion a lot. Unfortunately, the video stream has to be manipulated quite much and looks rather like a comic movie than a real video stream. However, in many scenarios this should not be an issue though. Additionally, one could think of this approach as in inspiration for other merging techniques.

A major issue with many AR environments today is the lack of depth information. Most applications can just *overlay* a given video stream with generated objects. For systems with vision-based tracking that share tracking input and field of view this has not been an issue. When the marker was covered by an obstacle the augmented object was hidden as well. Our approach allows to keep the augmented object within the scene even though the marker is not completely visible from the user's point of view. However, this leads to the unwanted side-effect of virtual objects covering real world objects even though they are perceived closer to the user. To tackle this issues depth information out of the real word have to transferred into the virtual representation. A promising low-cost approach includes the usage of Kinect[5]. A Kinect mounted with a top view of the scene might provide the necessary information. The use of two Kinects mounted with a user's point of view is harder to achieve since the risk of cross-interference caused by emitted infrared light is high. Another more sophisticated approach includes the usage of body trackers like Vicon. The bone and positioning data provided could be used to adapt a "invisible" virtual representation of the user within the scene. However, the performance and quality correlates with the detail level of the models gaining best results with a full upper body tracking including arms and hands. Additionally it brings in more obtrusive devices into the set up which most probably will affect subject behavior in an unforeseeable way.

[5] http://www.xbox.com/kinect

7. Conclusion

The goal of this thesis was to design and implement a framework to research on cooperative planning tasks. To achieve this goal stakeholder groups were identified and interviews were conducted. Combined with footage study results and proposals gained from focus group meetings, requirements were elicited and augmentation prototypes were designed and implemented later on.

Subsequently, the requirement elicitation insights were used to develop a new, more flexible object-oriented framework called LAFORGE 3 which succeeds the currently used augmentation system LAFORGE. LAFORGE 3 offers, since this was one of the redesign objectives, new features such as cooperative tracking and animation support. Without the necessity of adding additional hardware to the setup, cooperate tracking increases the user experience because tracking results can be combined and used for augmentation purposes even though they are not available locally. The very same data also allow fairly reliable assumptions about subjects' field of view which offer many opportunities for AR assistance. Furthermore, global data were used to create a spatial model of all markers within the scene. This model was used to introduce constraints between scene objects.

In performance tests, LAFORGE 3 achieved a permanently higher frame rate than LAFORGE. Additionally, while LAFORGE's frame rate dropped related to the amount of markers which had to be processed simultaneously, LAFORGE 3's frame rate was not influenced. LAFORGE 3 also showed better tracking performance than LAFORGE due to the usage of global tracking information. No improvements could be achieved concerning tracking latency which is even slightly higher in LAFORGE 3 than in LAFORGE. This need to be judged in light of the circumstance that it was not an agreed goal that LAFORGE 3 entered the productivity stage by the end of the planned developments. Rather, Laforge3 aimed to provide the point of departure for ongoing optimizations and finally for AR-based alignment research.

To investigate the newly offered augmentation possibilities we designed seven augmentation prototypes according to Interaction Design standards. The augmentations were (a) Violation Visualization to indicate if distances between contrary types of objects have to be taken in account, (b) Violation Sonification, which signals to the interaction partner acoustically when specific constraints such as visual interference with other objects are violated, (c) Light of Attention and (d) Glowing Frames, two alternative visualization concepts meant to communicate the position of the interlocutor's field of view, (e) Virtual Visitor, which can be used to *dive* into the scene and observe it from different angles and (f) Attention Request, a method to guide the interaction partner's attention to a specific point in the scene, from which (a),(c),(d) and (e) were chosen to be implemented.

The implemented augmentation prototypes will be tested in further user studies. Further improvement will not just benefit that field of research but allow the framework to ease access to the development of AR applications without profound knowledge of all methods and technologies provided by third party open source libraries combined and used within it.

Bibliography

[Art00] Kevin Arthur. "Effects of field of view on task performance with head-mounted displays". Doctoral Dissertation. New York, New York, USA: University of North Carolina, 2000, p. 137. ISBN: 0897918320. DOI: 10.1145/257089.257116. URL: http://portal.acm.org/citation.cfm?doid=257089.257116.

[Azu97] Ronald T Azuma. "A Survey of Augmented Reality". In: *Presence: Teleoperators and Virtual Environments* 4.August (1997), pp. 355–385. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.35.5387\&rep=rep1\&type=pdf.

[Bar10] Barco. "Projection mapping: A map to spectacle". In: *redefine* 2.2 (2010), pp. 4–6. URL: http://www.barco.com/ebooks/redefine_issue_2/index.html.

[Ber04] Daniel M Berry. *The Requirements Iceberg and Various Icepicks Chipping at It*. 2004.

[BFO92] Michael Bajura, Henry Fuchs, and Ryutarou Ohbuchi. "Merging virtual objects with the real world: Seeing ultrasound imagery within the patient". In: *ACM SIGGRAPH Computer Graphics* 26.2 (1992), pp. 203–210. ISSN: 0097-8930. URL: http://portal.acm.org/citation.cfm?id=134061.

[BK02] M. Billinghurst and H. Kato. "Collaborative augmented reality". In: *Communications of the ACM* 45.7 (2002), pp. 64–70. ISSN: 0001-0782. URL: http://portal.acm.org/citation.cfm?id=514236.514265.

[BK08] A.H. Behzadan and V.R. Kamat. *Simulation and Visualization of Traffic Operations in Augmented Reality for Improved Planning and Design of Road Construction Projects*. 2008. URL: http://portal.acm.org/citation.cfm?id=1517172.

[BLO+04] Wolfgang Broll, Irma Lindt, J. Ohlenburg, et al. "ARTHUR: A Collaborative Augmented Environment for Architectural Design and Urban Planning". In: *Journal of Virtual Reality and Broadcasting* 1.1 (2004), pp. 1–10. ISSN: 1860-2037. URL: http://scholar.google.com/scholar?hl=en\&btnG=Search\&q=intitle:ARTHUR+:+A+Collaborative+Augmented+Environment+for+Architectural+Design+and+Urban+Planning\#0.

[BMM98] William J. Brown, Raphael C. Malveau, and Thomas J. McCormick, Hays W. Mowbray. *AntiPatterns: Refactoring Software, Architectures, and Projects in Crisis*. 1st. Wiley, 1998, p. 336. ISBN: 978-0471197133.

[BR03] Patrick Baudisch and Ruth Rosenholtz. “Halo: a Technique for Visualizing Off-Screen Locations”. In: *Proceedings of the conference on Human factors in computing systems - CHI ’03*. 5. New York, New York, USA: ACM Press, 2003, pp. 481–488. ISBN: 1581136307. DOI: 10.1145/642611.642695. URL: http://portal.acm.org/citation.cfm?doid=642611.642695.

[BRB05] O. Bimber, R. Raskar, and Bimber. “A Brief Introduction to Augmented Reality”. In: *Spatial augmented reality: Merging real and virtual worlds*. AK Peters Ltd, 2005, p. 33. ISBN: 1568812302. URL: http://books.google.com/books?hl=en\&lr=\&id=zD_Q-CYi0SAC\&oi=fnd\&pg=PR11\&dq=Spatial+Augmented+Reality\&ots=1T303BcZRS\&sig=FRZ4qbliUAehFafDsPw1MZSwarg.

[CK92] M.G. Christel and K.C. Kang. *Issues in Requirements Elicitation*. Tech. rep. September. Pittsburgh, Pennsylvania: Software Engineering Institute, Carnegie Mellon University, 1992. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.141.5894\&rep=rep1\&type=pdf.

[CM92] T.P. Caudell and D.W. Mizell. “Augmented reality: an application of heads-up display technology to manual manufacturing processes”. In: *Proceedings of the Twenty-Fifth Hawaii International Conference on System Sciences* (1992), 659–669 vol.2. DOI: 10.1109/HICSS.1992.183317. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=183317.

[Coh88] Louis Cohen. “Quality function deployment: An application perspective from digital equipment corporation”. In: *National Productivity Review* 7.3 (Jan. 1988), pp. 197–208. ISSN: 02778556. DOI: 10.1002/npr.4040070302. URL: http://doi.wiley.com/10.1002/npr.4040070302.

[Cow01] Nelson Cowan. “The magical number 4 in short-term memory: A reconsideration of mental storage capacity”. In: *Behavioral and Brain Sciences* 24.1 (Feb. 2001), pp. 87–114. ISSN: 0140525X. DOI: 10.1017/S0140525X01003922. URL: http://www.journals.cambridge.org/abstract_S0140525X01003922.

[Csi08] Mihaly Csikszentmihalyi. *Flow: The Psychology of Optimal Experience*. 1ST. Harper Perennial Modern Classics, 2008, p. 336. ISBN: 0061339202. URL: http://www.worldcat.org/isbn/0061339202.

[DB08] Henry Been-Lirn Duh and Mark Billinghurst. “Trends in augmented reality tracking, interaction and display: A review of ten years of ISMAR”. In: *2008 7th IEEE/ACM International Symposium on Mixed and Augmented Reality* (Sept. 2008), pp. 193–202. DOI: 10.1109/ISMAR.2008.4637362.

URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4637362.

[DBH+09] Angelika Dierker, Till Bovermann, Marc Hanheide, et al. “A Multimodal Augmented Reality System for Alignment Research”. In: *Proceedings of the 13th International Conference on Human-Computer Interaction*. Vol. 1. 2009, pp. 1–5. URL: http://aiweb.techfak.uni-bielefeld.de/files/hcii09dierker_et_al-preprint.pdf.

[DMH+09] Angelika Dierker, Christian Mertes, Thomas Hermann, et al. “Mediated attention with multimodal augmented reality”. In: *Proceedings of the 2009 international conference on Multimodal interfaces - ICMI-MLMI ’09* (2009), p. 245. DOI: 10.1145/1647314.1647368. URL: http://portal.acm.org/citation.cfm?doid=1647314.1647368.

[DPH11] Angelika Dierker, Karola Pitsch, and Thomas Hermann. *An augmented-reality-based scenario for the collaborative construction of an interactive museum*. Tech. rep. Bielefeld University, 2011.

[Edm] Edmstudio. *OgreAR*. URL: http://software.edmstudio.com/browser/ogre/OgreAR.

[Eng04] Pontus Engelbrektsson. “Enabling the User-Exploring methodological effects on user requirements elicitation”. dissertation. Chalmers University of Technology, 2004, p. 67. ISBN: 9172914912. URL: http://swepub.kb.se/bib/swepub:oai:services.scigloo.org:4308?tab2=abs\&language=en.

[FHT08] Jan Fischer, Michael Haller, and Bruce H Thomas. “Stylized Depiction in Mixed Reality”. In: *International Journal* 7.4 (2008), pp. 71–79.

[FK99] L. Fiesser and M. Kiupel. “Interaktive Exponate–mehr als eine Attraktion für Kids”. In: *Museum aktuell* 42.2 (1999), 1488f. URL: http://www.phaenomenta.com/flensburg/dansk/phaenomenta/publikationen/schriftenreihe_2.pdf.

[Fut07] Futnuh. *OgreAR, an augmented reality interface for Ogre*. 2007. URL: http://www.ogre3d.org/forums/viewtopic.php?t=30376.

[Gar10] Gartner. *Hype Cycle for Emerging Technologies (Press Release)*. 2010. URL: http://www.gartner.com/it/page.jsp?id=1447613 (visited on 02/24/2011).

[Hip86] E. von Hippel. “Lead Users: A Source of Novel Product Concepts”. In: *Management Science* 32.7 (July 1986), pp. 791–805. ISSN: 0025-1909. DOI: 10.1287/mnsc.32.7.791. URL: http://mansci.journal.informs.org/cgi/doi/10.1287/mnsc.32.7.791.

[HV92] Cornelius Herstatt and Eric Von Hippel. "Developing New Product Concepts Via the Lead User Method: A Case Study in a 'Low Tech' Field". In: *Journal of Product Innovation Management* 9.3 (1992), pp. 213–221. URL: http://scholar.google.com/scholar?hl=en\&btnG=Search\&q=intitle:Developing+New+Product+Concepts+Via+the+Lead+User+Method:+A+Case+Study+in+a+'Low+Tech'+Field\#4.

[Kar96] MariAnne Karlsson. "User Requirements Elicitation - A Framework for the Study of the Relation between User and Artefact". Doctoral thesis. Chalmers University of Technology, 1996.

[Kir08] Peter Kirn. *Projection as Augmented Reality: Mapping Magic, Made with vvvv*. 2008. URL: http://createdigitalmotion.com/2008/10/projection-as-augmented-reality-mapping-magic-made-with-vvvv/.

[KMS07] Denis Kalkofen, Erick Mendez, and Dieter Schmalstieg. "Interactive Focus and Context Visualization for Augmented Reality". In: *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on Augmented Reality*. Ieee, Nov. 2007, pp. 191 –201. ISBN: 978-1-4244-1749-0. DOI: 10.1109/ISMAR.2007.4538846. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4538846.

[LRJ02] MA Livingston, LJ Rosenblum, and SJ Julier. "An augmented reality system for military operations in urban terrain". In: *The Interservice/Industry*. 2002, pp. 1–8. URL: http://ntsa.metapress.com/index/qdntl0771u631dam.pdf.

[McC04] Steve McConnell. *Code Complete: A Practical Handbook of Software Construction*. Ed. by 2nd. Microsoft Press, 2004, p. 960. ISBN: 978-0735619678.

[Mer08] Christian Mertes. "Multimodal Augmented Reality to Enhance Human Communication". Diploma Thesis. Bielefeld University, 2008. URL: http://bieson.ub.uni-bielefeld.de/volltexte/2009/1414/pdf/diplomathesis_cmertes.pdf.

[Mil56] George A. Miller. "The magical number seven, plus or minus two: some limits on our capacity for processing information." In: *Psychological Review* 63.2 (1956), pp. 81–97. ISSN: 0033-295X. DOI: 10.1037/h0043158. URL: http://content.apa.org/journals/rev/63/2/81.

[MTU+94] P. Milgram, H. Takemura, A. Utsumi, et al. "Augmented reality: A class of displays on the reality-virtuality continuum". In: *SPIE 2351-34 (Proceedings of telemanipulator and telepresence technologies* 2351 (1994), pp. 282–292. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.83.6861\&rep=rep1\&type=pdf.

[Ogr] OgreMax. *OgreMax Tools Collection*. URL: http://www.ogremax.com/.

[PG04] Martin J Pickering and Simon Garrod. “Toward a mechanistic psychology of dialogue.” In: *The Behavioral and brain sciences* 27.2 (Apr. 2004), 169–90; discussion 190–226. ISSN: 0140-525X. URL: http://www.ncbi.nlm.nih.gov/pubmed/15595235.

[PRS02] J Preece, Y Rogers, and H Sharp. *Interaction Design: Beyond Human-Computer Interaction*. John Wiley, 2002, p. 519. ISBN: 0471492787. URL: http://oro.open.ac.uk/5250/.

[RJR+05] G. Rickheit, G. Jäger, Helge Ritter, et al. *Proposal for the establishment and funding of collaborative research centre 673 "alignment in communication" for 2006/2-2010/1*. 2005.

[Sup10] Superbien. *Envision : step into the sensory box*. 2010. URL: http://vimeo.com/10692284.

[SZ04] K. Salen and E. Zimmerman. *Rules of play: Game design fundamentals*. The MIT Press, 2004, p. 670. ISBN: 0262240459. URL: http://books.google.com/books?hl=en\&lr=\&id=UM-xyczrZuQC\&oi=fnd\&pg=PP13\&dq=Rules+of+Play:+Game+Design+Fundamentals\&ots=2zCDtX8GVx\&sig=zfgB4JIxk9tNVYC6Oqa_5w80uPI.

[WFB+04] S. Wrede, J. Fritsch, C. Bauckhage, et al. “An XML based framework for cognitive vision architectures”. In: *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.* (2004), 757–760 Vol.1. DOI: 10.1109/ICPR.2004.1334304. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1334304.

[WSE+10] Benjy Weinberger, Craig Silverstein, Gregory Eitzmann, et al. *Google C++ Style Guide*. 2010. URL: http://google-styleguide.googlecode.com/svn/trunk/cppguide.xml.

[ART] ARToolworks Inc. *osgART - ARToolworks*. URL: http://www.artoolworks.com/products/stand-alone/osgart/.

[Fra] Frauenhofer IGD. *Instantreality - link today’s standards to tomorrow’s technology*. URL: http://www.instantreality.org/story/what-is-it/.

[Von07] Eric Von Hippel. “Horizontal innovation networks—by and for users”. In: *Industrial and corporate change* 16.2 (2007), p. 293. ISSN: 0960-6491. URL: http://icc.oxfordjournals.org/content/16/2/293.short.

Glossary

additive stencil shadows A description from Ogre's manual: Stencil shadows are a method by which a 'mask' is created for the screen using a feature called the stencil buffer. This mask can be used to exclude areas of the screen from subsequent renders, and thus it can be used to either include or exclude areas in shadow. Calculating the shadow volume for a mesh can be expensive, and it has to be done on the CPU, it is not a hardware accelerated feature. The advantage of stencil shadows is that they can do self-shadowing simply on low-end hardware, provided you keep your poly count under control. The complete manual can be found at http://www.ogre3d.org/docs/manual. 77

alignment The term alignment – as it is used within this work – originates from linguistics an describes a subconscious internal state adaptation processes of agents taking part in a dialog. Thereby information exchange necessary for understanding can be reduced significantly which improves communication speed and performance. Alignment can be observed due to changes in dialog partners' communication concerning words used or (copied) behaviour patterns for example. 1, 5, 77

C5 A project within the CRC 673 Alignment in Communication at Bielefeld University. Its goal is to investigate benefits of augmented reality for alignment research and to develop AR-based methods to support alignment. 1, 2, 5, 6, 10, 11, 19–21, 35, 37, 39, 67, 68, 77, 90, 98, 101

calibration Calibration is the process of checking, adjusting, or determining measuring instruments by comparison results with a standard. If data received previously calibrated devices do differ significantly from these standard, a device is called decalibrated and requires recalibration.. 57, 77

Camel case Camel case or camelCase is a coding notation style where several words are written without spaces. Capital letters indicate a new word. A phrase written in camel case usually consists of several uppercase letters which looks like bumps metaphorically. The phrase 'new color value' converted into camel case becomes 'newColorValue'. 53, 77

CRC Collaborative Research Centers are interdisciplinary research projects of long duration funded by the German Research Foundation (DFG). Their goal is to gain new insights into a specific research field. 77

IEEE-1394-Standard IEEE 1394 or Firewire is a serial bus interface standard for high-speed communications and isochronous real-time data transfer. Currently there are two version in use with data rates up to 400 Mbit/s (FireWire 400) or up to about 800 Mbit/s (FireWire 800). The standards are compatible so that FireWire 800 devices can be operated with or operate older hardware. 39, 77

Integrated Performance Primitives Intel's Integrated Performance Primitives is a multi-threaded software library written for data processing and multimedia. It is available for a variety of platforms and supports Intel processors and compatible making use of Intel specific processor features like MMX or SSE. 65, 77

jitter Jitter is a term describing little displacements occurring with a high frequency. In electronics it describes a phenomena which "blurs" signals. 22, 77

Kinect Kinect is a vision based controller system developed by Microsoft for its Xbox 360. Players can interact with games through gestures and poses instead of using traditional game controller. Kinect uses infrared light to reconstruct 3D depth information from a monochrome picture taken by webcam like CMOS sensor. 70, 77

mediating tool A mediating tool is a product representation meant to support the requirement elicitation process. Sketches are considered as low fidelity mediating tools while full featured prototypes or released products are called high fidelity mediating tools. 20, 77

offset An offset can be seen as a systematic error, caused by imprecise measurement results. In AR a tracking offset occurs if the distortion of the camera used is not known or imprecise. 22, 77

probing Probing is question technique to get more insight during an interview. If the interviewer hopes for more specific information concerning a specific topic she uses probing questions to get more detailed answers from the interviewee. One part of probing is the identification of signals which indicate that more information can be elicited. 20, 77

Qt Qt is a cross-platform application framework which is commonly used for development of applications with a graphical user interface (GUI). There exist many wrapper for other coding languages – so-called bindings – so that Qt development is not limited to C++ applications. It is owned by Nokia an maintained by its Qt Development Frameworks division. 41, 77

Vicon Vicon is a professional vision based motion capturing system by Vicon Motion Systems. It uses infrared light to tracker markers covered with a infrared light reflecting paper. A common Vicon system consists of several high speed cameras with infrared light emitting units and a central server to process these data. Vicon systems are widely used for movie animations, video games and motion analysis. 70, 77

Acronyms

laforge Lightweight Augmented-Reality Facility with Open Real-world-based Graphical Enhancement. 2, 3, 21, 34–39, 43–46, 49, 50, 52, 54–58, 61, 63–71, 77, 85–87, 90–97, 99, 101

acme Active Memory. 55, 67, 77, 85

API application programming interface. 77

AR augmented reality. 1, 2, 5–11, 15, 20–24, 27, 31, 32, 35, 36, 38, 67, 70–72, 77, 80

ARbInI AR-based Interception Interface. 1, 2, 6, 77

ART AR Toolkit. 35–38, 67, 77

ARTKP AR Toolkit Plus. 36, 38, 47, 77

BTL Bielefeld Type Library. 39, 43, 67, 77, 85, 92

CRC Collaborative Research Centre. 2, 9, 77, *Glossary:* CRC

FOV field of view. 6, 77

fps frames per seconds. 38, 40, 55, 56, 61, 68, 77

GUI graphical user interface. 77, 80

HCI human-computer interaction. 1, 77

HMD Head-Mounted Display. 1, 2, 5–7, 11, 15, 20, 22–24, 28, 29, 32, 40, 55, 57–59, 62, 77

HUD Heads-Up Display. 70, 77

ICL Image Component Library. 35, 37, 39–41, 47, 52, 67, 77, 85

IxD Interaction Design. 5, 11, 14, 33, 66, 71, 77

MR Mixed Reality. 36, 77

Ogre Object-oriented Graphics Rendering Engine. 2, 36, 37, 39, 41, 47, 49, 59–63, 65–67, 77, 79, 85

OSC Open Sound Control. 77

OSG OpenSceneGraph. 36, 77

osgART osgART. 37, 77

Phaenomeum Phaenomeum. 39, 48, 77

POV point of view. 15, 77

RPC Remote Procedure Call. 43, 44, 63, 77

SAR Spatial Augmented Reality. 1, 7, 77

UML Unified Markup Language. 37, 38, 68, 77, 89

VR Virtual Reality. 7–9, 77

X3D Extensible 3D. 36, 77

XCF XML enabled Communication Framework. 35, 39, 41, 42, 44, 47, 52, 60, 67, 69, 77, 85, 87, 92

XML Extensible Markup Language. 35, 38–40, 42–44, 49, 52, 69, 77

XMLTIO XML Template I/O. 39, 44, 77

A. Installation

Installing LAFORGE 3 and related third party software requires some knowledge and experience with compiling code from a terminal.

The first step is to install all required third party software. Ogre is available at http://www.ogre3d.org and can be checked out and installed from a mercury repository. Introduction of how this can be done can be found there as well. Ogre-Procedural can be retrieved from http://code.google.com/p/ogre-procedural. ICL and BTL can be checked out from the CITEC open source SVN repository at from http://opensource.cit-ec.de. Boost is available at http://www.boost.org but is also provided for many Linux distribution by apt, yast, portage, pacman or other software management solutions. XCF and acme and their dependencies are available in the custom gar-installer[1] from Bielefeld University. However, using it requires access to repository of CITEC and the Faculty of Applied Computer Science.

LAFORGE 3 can be obtained from the CD delivered with this thesis or from the project management system of CITEC. To compile and install LAFORGE 3 CMake[2] is required. However, like Boost, CMake is available for a variety of platforms. If Ogre has been compiled and installed already, CMake is already present. To find all necessary software, LAFORGE 3 uses *pkg-config*[3]. To ensure that LAFORGE 3 is able to find all required libraries we recommend to install it and obtain PC files of all installed libraries. If the libraries were compiled, these files should be located within the project folders. If binary files were installed instead, most probably there will be *dev* packages containing these file.

1. In a terminal, switch into the LAFORGE 3 folder,
2. create a *build* folder and switch into it,
3. execute `cmake -DLAFORG3_VAR_INSTALL_PREFIX=/laforge3/install/path ..` to run configuration. This location should be identical with the location where ICL and Ogre-Procedural were installed in. If that cannot be done, the CMake configuration file `CMakeLists.txt` in the root folder of the project has to be adapted.

[1] http://eris.liralab.it/wiki/GAR_Installer
[2] http://www.cmake.org
[3] http://www.freedesktop.org/wiki/Software/pkg-config

4. (optional) running `cmake-gui ..` instead let you configure all parameters more comfortably with a gui

5. run `make` to build the code

6. if this was done successfully one can find the the application `laforg3`[4] and the library `liblaforge.so` in the folder `build/src`. The test suite and applications such as the `scene viewer` can be found in `build/test` and `build/test/testapp`

[4]LAFORGE 3 was developed under the name *laforg3* which is an allusion to so-called *leet speech*.

B. Running Instructions

The first step is to start a spread and an XCF dispatcher. Afterwards, an Active Memory called `markermemory` has to be initialized. With these three things set up, LAFORGE 3 can be started from a terminal with the following parameters passed:

- –name=INSTANCE_NAME (required)
- –tracker-conig=/path/to/artkp/config/file (required)
- –static (optional)
- –help (will quit the application after printing a help message)

C. UML

UML diagrams shown here are meant to be references to the high-resolution versions provided on CD in the `uml` folder. The CD also contains MDUML files for MagicDraw[1] (created with version 16.5 Personal Edition). In these files one can find additional documentation and extra activity diagrams.

[1] http://www.magicdraw.com

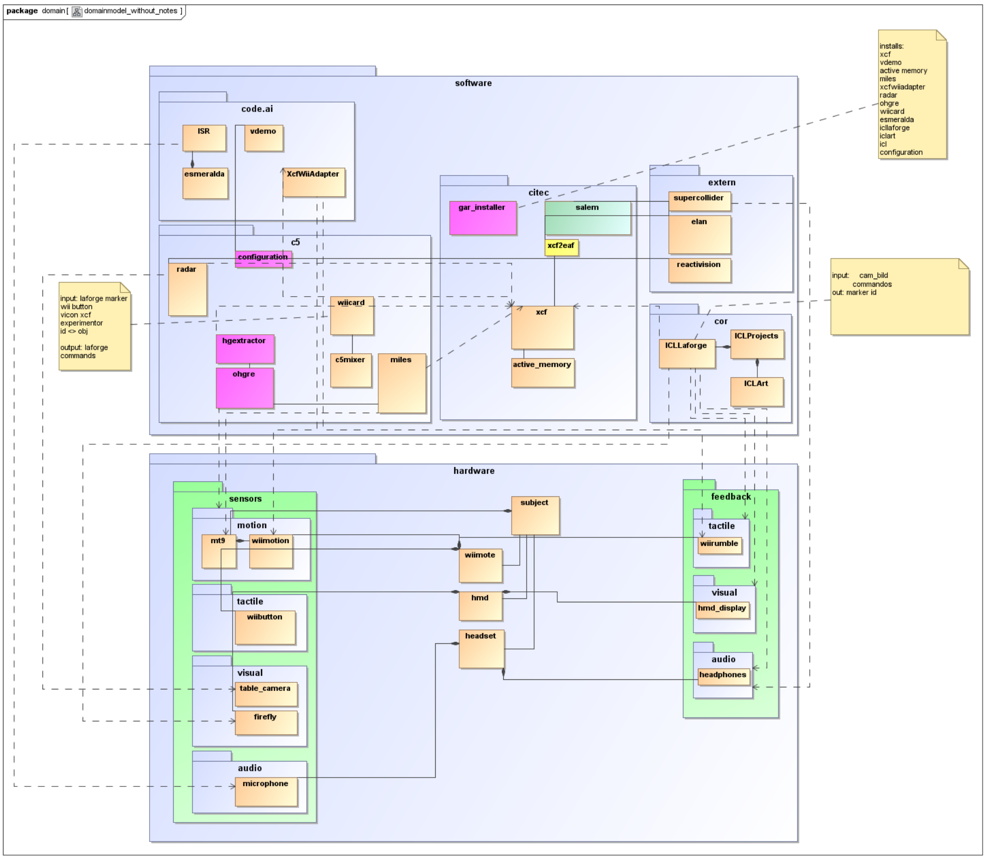

Figure C.1.: Focus group result. During a focus group meeting with the current developer team of the C5 project a domain model was produced. From a bird perspective all hardware and software components which are currently used were put into context. The goal was to get an overview of the environment LAFORGE 3 will operate in. See `uml_domain`.

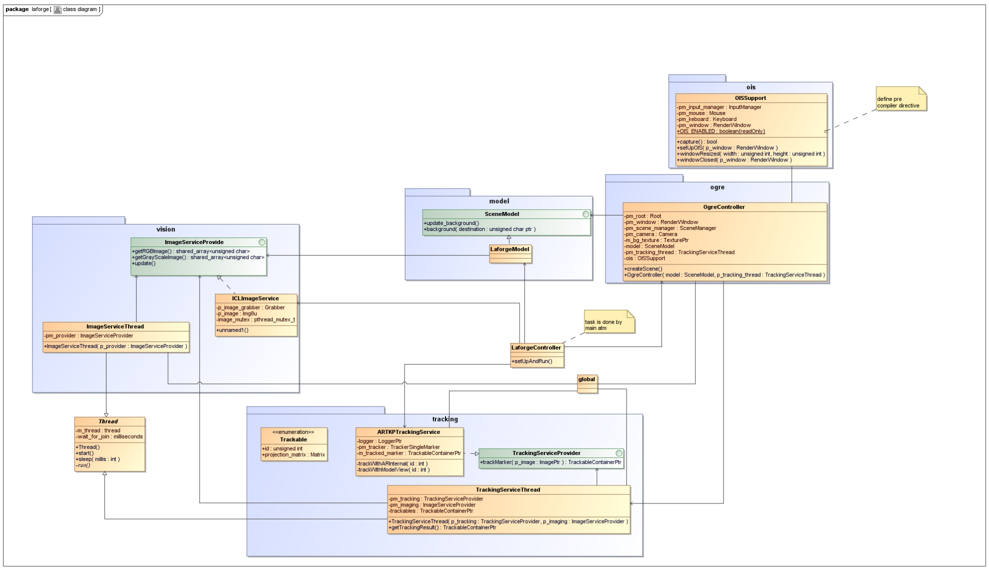

Figure C.2.: The first version of LAFORGE 3. The first increment of LAFORGE 3 was capable of augmenting static objects onto markers. Threading was already introduced and one can already see basic interfacing concepts. see `uml_laforg3_01`.

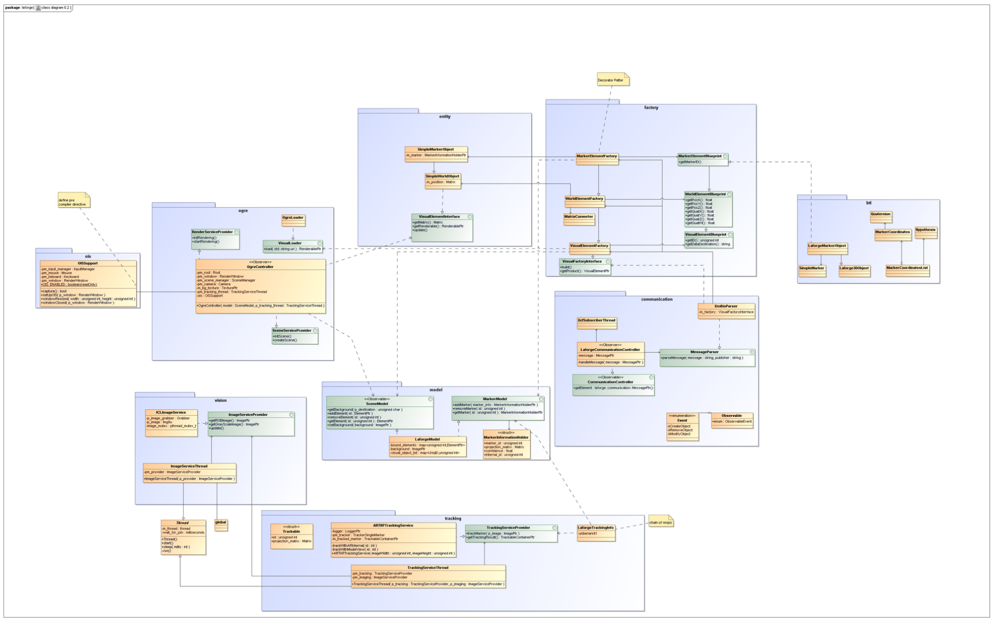

Figure C.3.: The second increment of LAFORGE 3. In contrast to the first increment, XCF communication and BTL usage were introduced. Rendering and 3D scene management were interfaced and object loading features implemented. See `uml_laforg3_02`.

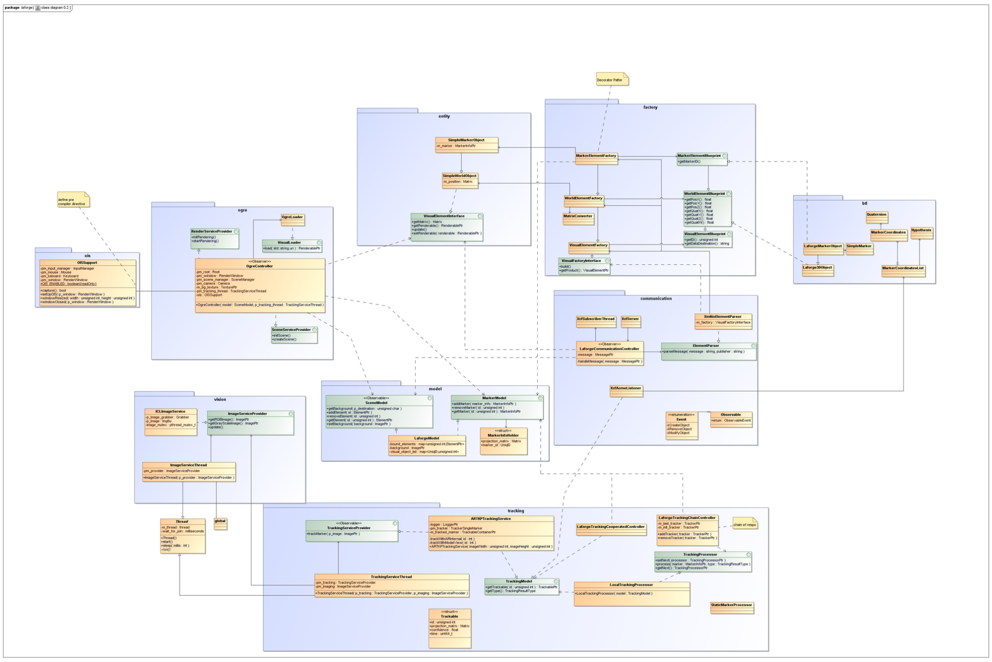

Figure C.4.: The third increment of LAFORGE 3. A first version of cooperated tracking was introduced. The *chain of responsibility* pattern approach was declined later and replaced by a central tracking management. In addition to `uml_laforg3_03`, the MagicDraw project file `l3.mdzip` also contains activity diagrams which describe both tracking management approaches.

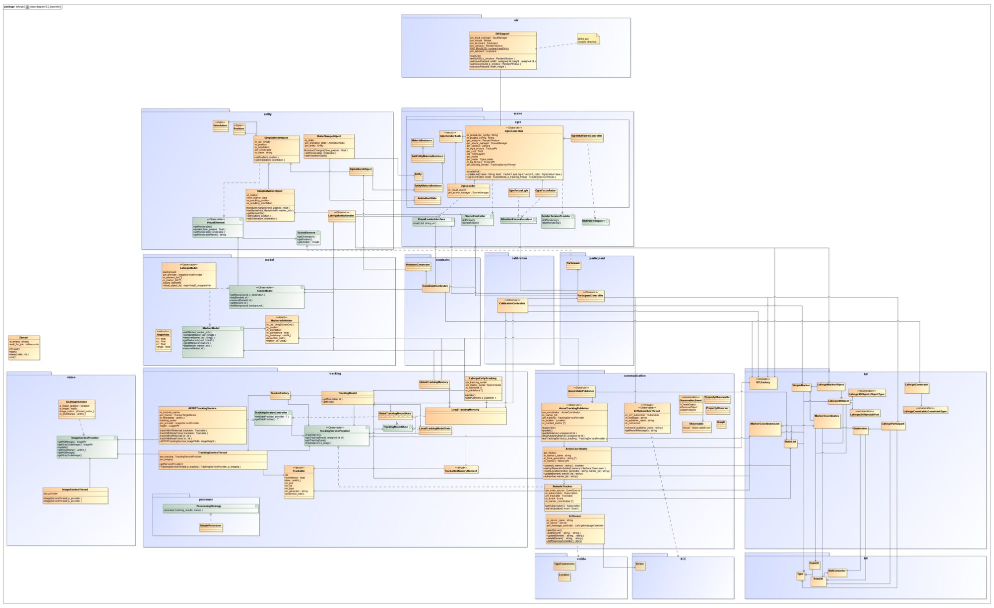

Figure C.5.: The fourth and final increment of LAFORGE 3. The communication architecture was refactored and decentralized. Additionally, the interactive prototypes `Ogre Focus Light` and `Ogre Radar` were introduced. Blackboard functionally was developed separately and finally integrated into this increment. Some class member functions were not documented even though they are available in that version. See `uml_laforg3_final`.

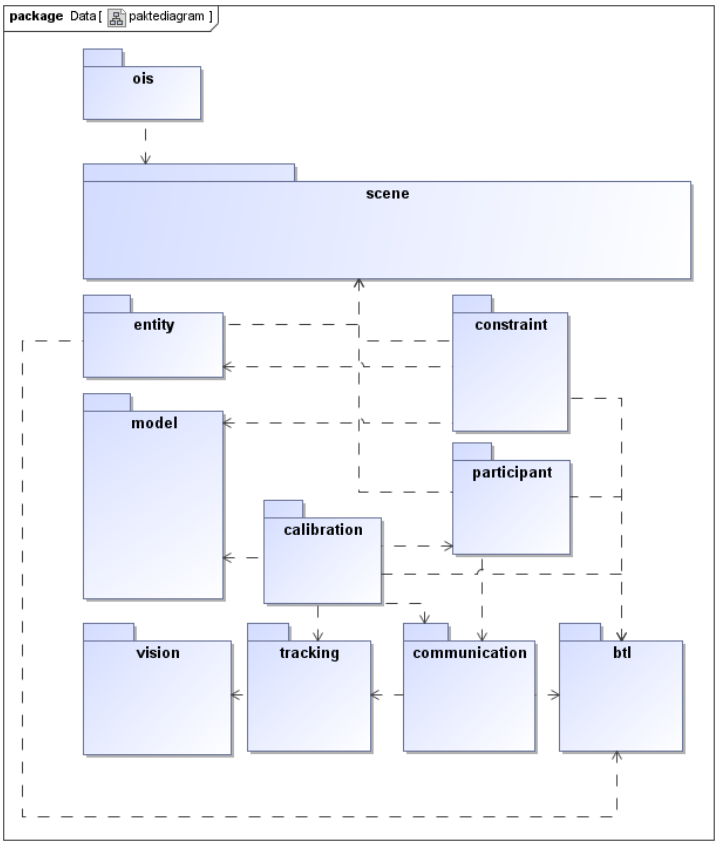

Figure C.6.: LAFORGE 3 package diagram. To visualize internal dependencies of LAFORGE 3, a package diagram was created. Packages are arranged in so-called *service layer* with the exception of the `scene` package. *Service layer* usually forbid dependencies pointing upwards since basic features should be located at the bottom most layer possible. We decided to put `scene` on top to have a *input-output* structure which means that the standard information flow is from the bottom to the top. Since some information is fed back to the blackboard and other communication services, this metaphor lacks consistency as well though. See `uml_laforg3_package`.

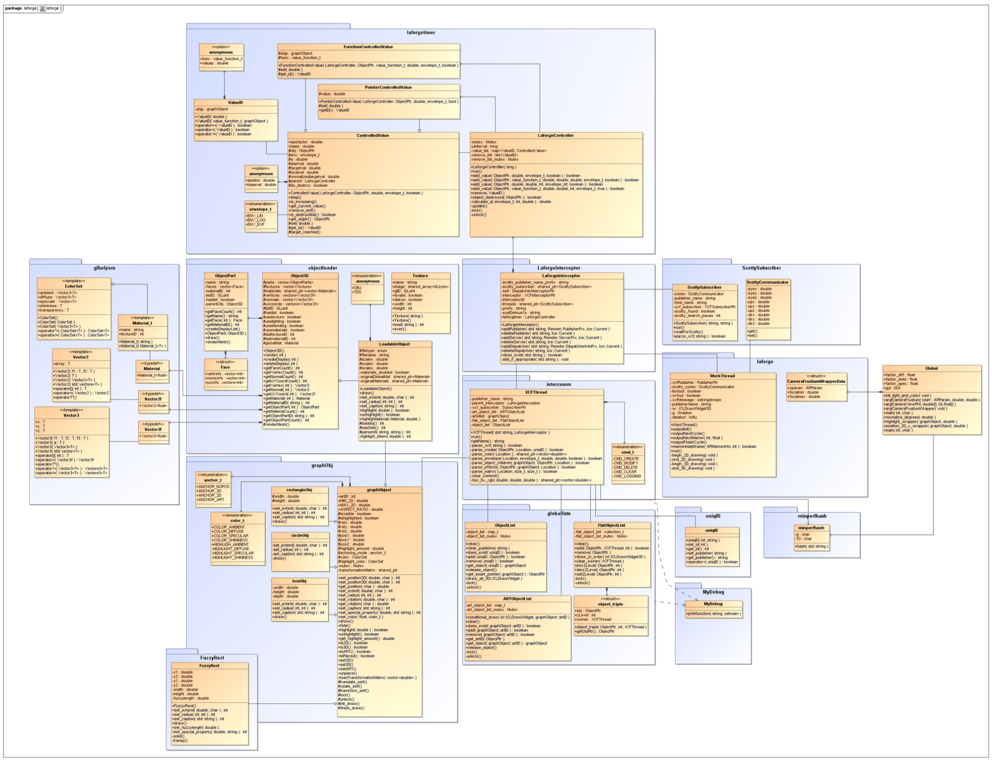

Figure C.7.: LAFORGE package diagram. The image shows a simplified version of LAFORGE's architecture. Is is simplified in the manner that not all dependencies were visualized to prevent clutter. However, all classes and all member functions and variables were included. See `uml_laforge_class`.

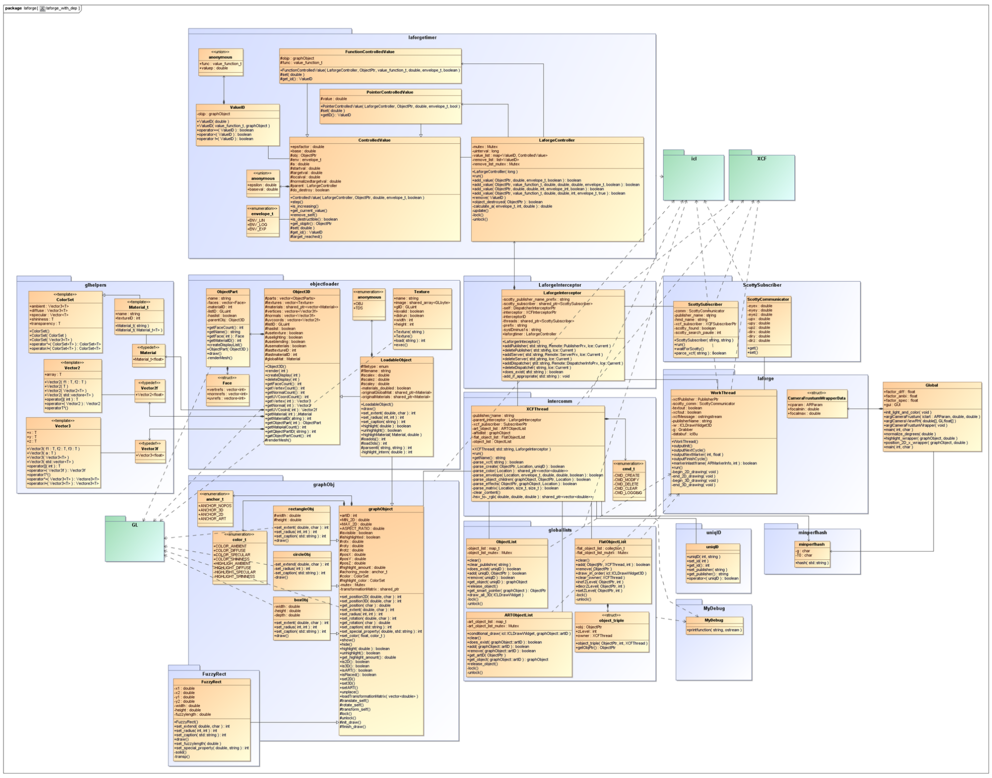

Figure C.8.: External dependencies of LAFORGE. This image shows LAFORGE's software architecture with an added information layer. One can see that external dependencies are spread of major parts of the system. This was one reason why refactoring was considered to be too time consuming. See `uml_laforge_dep`.

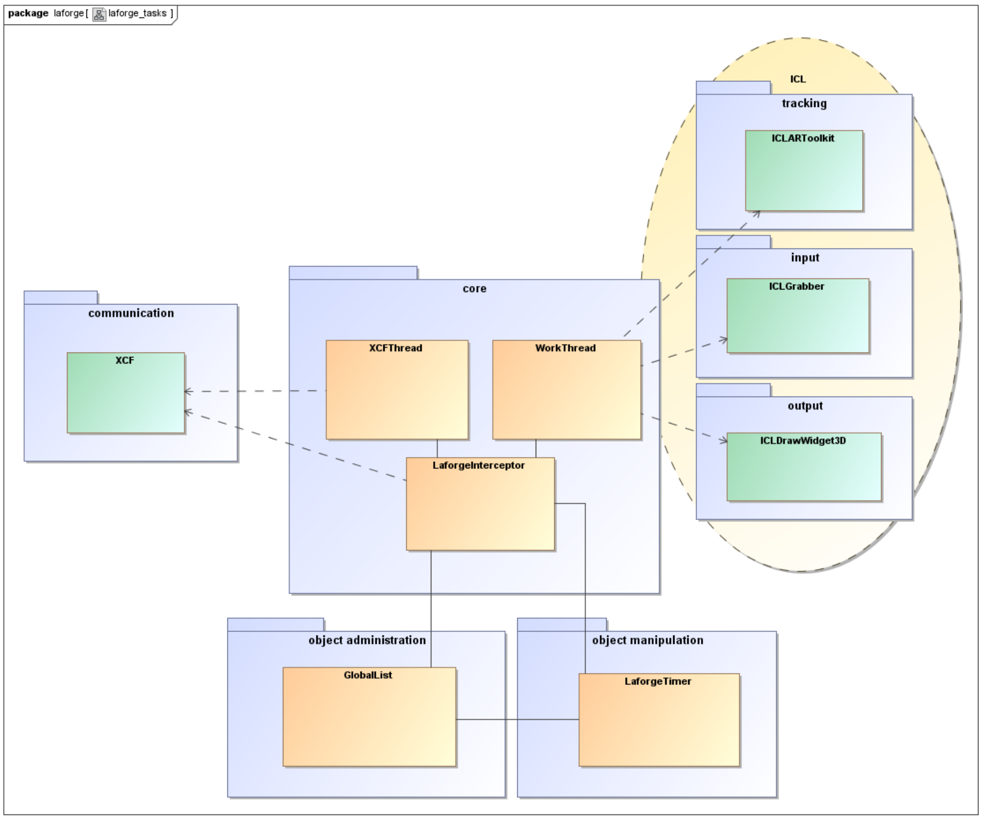

Figure C.9.: This images shows a simplified version of how responsibilities are spread over the system. It was used to clarify and validate review insights with the current C5 project team and Christian Mertes. See `uml_laforge_tasks`.

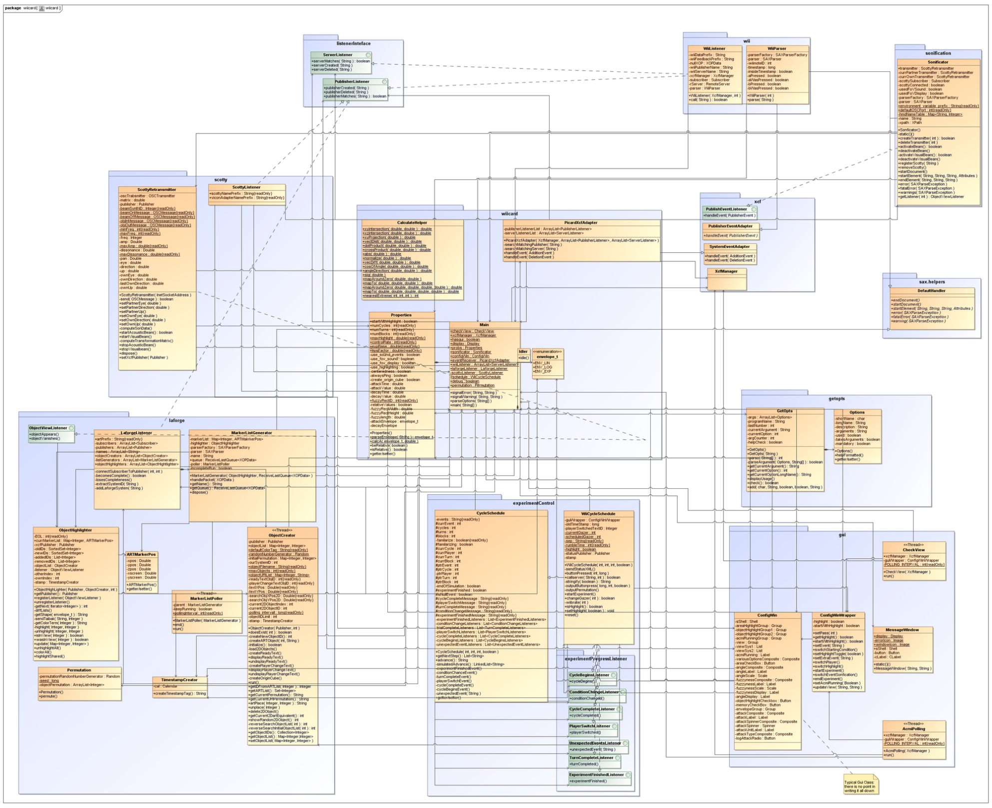

Figure C.10.: An architecture draft of Wiicard, the remote controlling interface of LAFORGE. Due to coupling issues between internal system structures review of Wiicard was ended unfinished. At this time, reimplementing LAFORGE had been already decided. See `uml_wiicard_unfinisheda`.

Acknowledgements

I would like to thank some people which have helped me a lot during my work on this thesis.

First of all I want to say thank you to Rich Potter, who not only spent quite a lot of time modeling, but also spent considerably more time with me figuring out ways to make his models work in Blender and Ogre even though that was unknown territory for him as well. He constantly changed, remodeled and optimized his models and even reviewed my thesis without complaint.

Without Florian Lier I would be still in the labs, trying to setup all the necessary software components. He gave me a head start, instant support during my thesis and was there when my tests were in danger of being canceled due to software issues. His endurance and creativity when it comes to problem solving impresses me a lot.

I would further like to thank Thomas Hermann who kept me on track and supported me with his expertise even when his schedule was completely packed. Thanks to his help and steady guidance I was able to put my initial ideas into practice and put them into shape. I also want to thank my second supervisor Karola Pitsch and Christian Schnier for their valuable inputs and ideas that they contributed to my work.

I am very grateful to Christian Mertes for his constant availability and his will to discuss and review LAFORGE related issues with me even though it required quite a lot of effort to place himself back into the work he had left almost three years ago. The fantastic work he conducted within the C5 project was a valuable source of knowledge and inspiration during my whole thesis.

When talking about C5, Angelika Dierker cannot go without a mention. It’s thanks to her that I was able to use the insights that have been gained in C5 during the last years. I also want to thank her for her time and expertise which supported my work in many ways.

I also want to thank all my other reviewers, namely Andreas Bremges, David Marshall, Maximilian Schnücke and Robert Schulze who reviewed my thesis in record-breaking time even though they were not familiar with the subject matter.

And last but not least all the people supporting awesome open source projects like ARToolkit, Ogre or Blender and sharing their knowledge in wikis, forums or open source applications so that people like me can benefit from their insights. Without their attitude and efforts my work would not have been possible.

D. Declaration

Bielefeld, 16.05.2011

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Alexander Neumann