Doctoral Thesis

Designing for tangible interaction

Author(s): Fjeld, Morten

Publication Date: 2001

Permanent Link:
https://doi.org/10.3929/ethz-a-004237155

Rights / License:
In Copyright - Non-Commercial Use Permitted
ETH Dissertation No. 14229

DESIGNING FOR TANGIBLE INTERACTION

DOCTORAL DISSERTATION
presented to obtain the title
DOKTOR DER TECHNISCHEN WISSENSCHAFTEN
DER EIDGENÖSSISCHE TECHNISCHE HOCHSCHULE ZÜRICH

by
Morten Fjeld
M.Sc. Appl. Math. NTNU
Born 22nd April 1965 in Bergen, Norway

Accepted by:
Prof. Dr. Dr. Helmut Krueger
Prof. Dr. Matthias Rauterberg

2001
Acknowledgments

I greatly thank Matthias Rauterberg, Professor at IPO, Technical University Eindhoven (TUE), in the Netherlands. He was the person who triggered and maintained my interest for the disciplines of Human-Computer Interaction (HCI) and Cognitive Ergonomics (CE). In establishing the research projects BUILD-IT and AMME, he offered me an excellent introduction to these research fields. In the context of the BUILD-IT project, I could carry out system design and empirical research under Rauterberg’s competent supervision. Rauterberg also introduced me to Action Regulation Theory, statistical methods, and Petri-net modelling of user behaviour.

I greatly thank Helmut Krueger, head of the (Institute of Hygiene and Applied Physiology) IHA, who provided a stimulating, enduring, and highly supportive surrounding. Krueger showed great interest and confidence in my dissertation program, and offered a base of continuity and trust. He took particular interest in assuring interaction patterns that would respect users’ perceptual and physiological skills. He also gave productive feedback to assure that extensive empirical work could be carried out.

I greatly thank Sissel Guttormsen Schär, head of the Man-Machine Interaction (MMI) group where I carried out the actual work. In following up on my every day work, Guttormsen Schär proved highly supportive and stimulating. With her great insight and experience, she gave competent advice on the design, execution, and analysis of empirical studies.

Martin Bichsel, head of the BUILD-IT project, was a persistently helpful colleague. Thanks to the groundbreaking and competent contribution of Bichsel, was I able to implemented and perfection the interaction methods presented here.

Kristina Lauche gave a major work-psychological contribution to the CSCW journal paper. She also provided founded advice and structure for the task analysis carried out earlier in the project.

Fred Voorhorst gave convincing advice and help on perception matters, design strategies, and logic reasoning.

Nicholas Ironmonger gave substantial input to the process of writing, not only on the level of English, but also in improving and formulating complex ideas and concepts.

Domenico Signorello designed i) the physical tools used to evaluated BUILD-IT against alternative tools and ii) the tasks used in the study of navigation tools. Both tasks were focal points of his MSc thesis. Domenico Signorello was the first student I advised closely and it was a great pleasure.

Barbara Stuckey greatly helped in giving the CSCW journal paper a readable, consistent, and scientific form. She drew from her long experience in economic and political research and teaching, and made an impressive effort to understand every detail of the technology described. She also taught me important aspects of English writing.
Wolfgang Weber gave grounded advice on collective action regulation, thereby providing theoretical framework for collaborative tangible interaction.

Samuel Schluep, a colleague in the AMME project and office-mate at MMI, also gave major help and lots of moral support throughout the whole project.

Christopher Schierz gave useful advice on the statistical aspects of experimental design.

Georg Kralidis gave professional support and advice on the analysis and interpretation of statistical data. He showed great interest in the problems I gave him.

Thomas Keller was an ever-optimistic BUILD-IT project colleague who took particular care of the marketing aspects of the project. His background in industrial design was also of great help to my activities, e.g. in allowing the realisation of a portable version of the system.

Markus Meier, head of Zentrum für Produkte-Entwicklung (IKB), showed me lots of trust and confidence in financially supporting my employment and travelling and by letting me organise my work according to my needs.

Stefan Dierssen, also a member of the BUILD-IT project, provided Computer-Aided Design (CAD) skills, useful for the input-output aspects described in the dissertation.

Kerstin Höger initiated and carried out a competition employing BUILD-IT as a means for creative collaboration among architectural students. This opened up a new horizon for putting the BUILD-IT system into practical use.

Georg Lorek introduced me to useful aspects of applied psychology, such as Gestalt and Systemic Work. Such insight helped me to function better in a complex professional setting.

Hans Gelke was the artist and producer behind the major video production presented at the CHI 2000 conference. In spite of opposing views and origins, we managed to produce an animated documentation on the navigation methods.

Nicole Kerness read most of the unpublished chapters of the dissertation and gave reflected and substantial advice on how to improve line of though and English language.

Terry Winograd, Scott Macenzie, Bonnie Nardi, and David Redmiles were highly professional reviewers of several of the publications. So where also some important blind reviewers of the CSCW journal paper, CHI2001 conference submission, and the INTERACT2001 conference paper.

Lars Selsbo was not directly involved in this dissertation, but gave me a thorough and professional introduction to the art of programming ADA, X, motif, and TCP. With Selsbo, I developed simulators, measuring systems, and communication packages.

Urs Gasser, my close friend during most of these years, gave endless moral support and could also give some very useful contribution to English writing.
# Contents

**SUMMARY** .................................................................................................................................................. 8

**ZUSAMMENFASSUNG** .................................................................................................................................. 9

**PART I: RESEARCH CONTEXT AND PROTOTYPICAL WORK** ....................................................................... 11

1 **INTRODUCTION** ........................................................................................................................................ 12

2 **DEFINING POTENTIAL BENEFITS FOR THE PROJECT PARTNERS** ......................................................... 13

   Task Analysis: Principles and Practice ........................................................................................................ 13
   Outcome of Task Analysis Carried out During Autumn 1997 ...................................................................... 14
   Project Partner 1: Soudronic, Neftenbach, Switzerland (www.soudronic.ch) ............................................. 14
   Project Partner 2: Mikron, Agno, Switzerland (www.mikron-tg.com) ........................................................ 15
   Project Partner 3: Von Roll Inova, Zurich, Switzerland (www.vonroll.ch) ..................................................... 17
   Project Partner 4: DAI, Zurich, Switzerland (www.dai.ch) .......................................................................... 18
   Other Project Partners .................................................................................................................................. 19
   Outcome and Interpretation of our Task Analysis ......................................................................................... 20
   Follow-up to the Task Analysis ..................................................................................................................... 21

3 **SETTING THE INITIAL DESIGN AGENDA THROUGH A USER STUDY** .................................................. 22

   Subjective Statements and Preferences ....................................................................................................... 23
   Observations ................................................................................................................................................... 24
   Concluding Remarks on the User Study with Respect to Brick Design ..................................................... 24
   Outcome of Part 2: Usability of the BUILD-IT System .................................................................................. 24
   The Statements Given by Planning Experts After Having Used the BUILD-IT System Were ....................... 26
   The Missing Features Were ............................................................................................................................ 27
   The New Functions Required Were .............................................................................................................. 27
   Concluding Remarks on the User Study with Respect to the BUILD-IT System ......................................... 28

**PART II: THEORETICAL GROUNDING AND EMPIRICAL INVESTIGATIONS - PAPER AND VIDEO PUBLICATIONS** ........................................................................................................................................ 29

4 **PHYSICAL AND VIRTUAL TOOLS: ACTIVITY THEORY APPLIED TO THE DESIGN OF GROUWPARE** ........................................................................................................................................... 30

   1. Introduction ................................................................................................................................................ 30
   2. The Concept of Tools in Activity Theory ................................................................................................... 31
   2.1 Tools and Objectifications ....................................................................................................................... 31
   2.2 Common Objectification ............................................................................................................................ 34
   2.3 Goal-Directed and Exploratory Action .................................................................................................... 34
   3. Design Philosophy ....................................................................................................................................... 35
   3.1 Augmented Reality ................................................................................................................................... 35
   3.2 Action and Perception ............................................................................................................................... 36
   3.3 Task Analysis As Part of the Design Process ............................................................................................ 37
   3.4 Exteriorization and Interiorization of Tools ............................................................................................. 37
   4. Design Example: the BUILD-IT System ..................................................................................................... 39
   4.1 Basic Aspects of the BUILD-IT System ................................................................................................... 39
   4.2 Design Practice of the BUILD-IT System ................................................................................................. 42
   5. Conclusion ................................................................................................................................................... 49

**Notes** ............................................................................................................................................................ 51

**Acknowledgments** ....................................................................................................................................... 52
APPENDIX C: COLLABORATIVE DESIGN WITH NEW INTERFACE TECHNOLOGIES: A COMPETITION

Collaborative Design: Exploring New Interface Technologies
From Indirect to Direct Interaction: Beyond Keyboard and Mouse
The System: Coincidence of Action and Perception
Procedure: Constructing Models (Step 1) and Composition (Steps 2)
The Competition Task: Creating Compositions
The Design Process: Generating Coherence
The Interacting Jury: Clarity versus Consensus
Conclusions: New Forms of Creative Collaboration

APPENDIX D: OTHER AUGMENTED REALITY (AR) RESEARCH PLATFORMS
DigitalDesk
MetaDESK
Illuminating Light
Envisionment and Discovery Collaboratory (EDC)
i-LAND
InfoTable / InfoWall

APPENDIX E: ACRONYMS, ABBREVIATIONS, AND VOCABULARY

REFERENCES
CURRICULUM VITAE
Summary

This project investigates the theoretical grounding, the practice of design, and usability aspects of Tangible User Interfaces (TUIs).

The framework for our empirical work is a TUI called the BUILD-IT system. This is a planning tool based on computer vision technology, with a capacity for complex planning and composition tasks. The system enables users, grouped around a table, to interact in a virtual scene, using physical bricks to select and manipulate virtual models. A plan view of the scene is projected onto the table. A perspective view of the scene, called side view, is projected on the wall. The plan view contains a storage space with originals, allowing users to create new models and to activate tools (e.g. navigation and height tools). Model selection is done by putting a brick at the model position. Once selected, models can be positioned, rotated and fixed by simple brick manipulation.

Our design practice is grounded on a work-psychological tradition called activity theory. This theory is based on the concept of tools mediating between subjects and objects. In this theory, an individual’s creative interaction with his or her surroundings can result in the production of tools. When an individual’s mental processes are exteriorized in the form of tools - termed objectification - they become more accessible to other people and are therefore useful for social interaction. We show how our understanding of activity theory has shaped our design philosophy for groupware and how we have applied it. Our design philosophy and practice is exemplified by a description of the BUILD-IT system. Guided by task analysis, a set of specific tools for different three-dimensional (3D) planning and configuration tasks was implemented as part of this system. We investigate both physical and virtual tools. These tools allow users to adjust model height, viewpoint, and scale of the virtual setting. Finally, our design practice is summarised in a set of design guidelines. Based on these guidelines, we reflect on our own design practice and the usefulness of activity theory for design.

Using the BUILD-IT system as a research platform for graspable interaction, our exploration takes the following path: We first introduce some of the problems related to working in physical and virtual environments, then indicates a few guidelines to achieve what we call natural interaction. Then we give more details about the interaction content, which is configuration and planning tasks. We then present new implementations for 3D navigation. As a particular use of hand-held tools, we introduce alternative ways to control model height in the BUILD-IT system. We also discuss the outcome of our design activity and suggests ways to advance the issues presented before.

Besides these major topics, the we also report on advanced uses of the system and similar research platforms for tangible user interaction. Then, employing a checklist stemming from activity theory, we discuss the value of the BUILD-IT system. Finally, we offer a general discussion of our design practice and outline what system-independent knowledge came out of our work, and what are the future challenges in the design and practical use of tangible user interfaces.
Zusammenfassung

Das vorliegende Projekt untersucht die theoretischen Grundlagen, die Designtätigkeit sowie Usabilityaspekte von greifbaren Benutzerschnittstellen.


PART I:

RESEARCH CONTEXT
AND
PROTOTYPICAL WORK
1 Introduction

This report gives the complete outcomes and results of my doctoral dissertation. It is offered as four parts, being:

Part I: Research context and prototypical work. This part offers an overview of task analysis carried out in forefront of the project to find out the needs and potential benefits of the industrial project partners. It also offers a short user study concerning the use of brick as interaction handles.

Part II: Theoretical grounding and empirical investigations - paper and video publications: This part is the central piece of the dissertation and is based on papers (two of them with a video) published or submitted to conferences and/or journals. The first paper offers the theoretical grounding for the dissertation, being activity theory, and gives details on the design process of the system. The second paper (with a video) shows the initial design of the system. The third paper delivers arguments for working with a Tangible User Interfaces, instead of using alternative and/or conventional planning tools. The fourth paper shows the first steps towards the design of navigation methods for a virtual environment. The fifth paper (with a video) gives a full presentation of the final implementation of the navigation methods. The sixth paper offers an empirical evaluation of the same navigation methods.

Part III: Advanced uses, discussion, and conclusion: First, this part shows advances uses of the system. Some of them were developed at the ETH Zurich, other were developed at TU Eindhoven, employing a basic version of the system. Second, this part offers a thorough discussion of the system, using a checklist based on activity theory. Thereby, we reflect upon the outcome of our design process in the same theoretical context as where we started out. Third, this part offers a general discussion of the whole doctoral project, structured along five central topics.

Part IV: Appendixes, glossary, and references: This final part offers three appendixes and a complete list of references. Appendix A offers a description of the BUILD-IT system (or simply: system), how it is assembled, and how it is used. Appendix B offers a description of software design and development. Appendix C reports on a competition where the system was employed by architectural students in a creative collaborative process. Appendix D reports on other Augmented Reality research platforms closely related to our system. Appendix E offers a list of acronyms, abbreviations, and specialised vocabulary.

My personal contribution to the whole BUILD-IT project mainly consisted of software design and development, several user studies and usability evaluations, and the construction of a theoretical grounding for the design process. In particular, I was concerned with the invention, design, implementation, and documentation of the navigation methods for the system.
2 Defining Potential Benefits for the Project Partners

The BUILD-IT project was a collaboration between ETH, four industrial partners, and four non-industrial partners. Since major resources, competence, and material support came from the industrial partners, they expected an outcome which could be useful to their planning departments. Within this context, we set out to define the context, potential uses, expected benefits, and requirements for a future BUILD-IT system (target technology). The method we used to collect information from the industrial project partners is called task analysis. Employing a set of five standardised questions (Q1-Q5), our task analysis was carried out as systematic interviews within the companies Soudronic, Mikron, Von Roll, and dai. In these companies, computer-supported mediation of planning tasks play an increasingly important role. We asked expert planners in these companies how they carry out planning and configuration (target activity) and what they expected from the BUILD-IT system. Based on these interviews, we quote, summarise, and interpret their answers.

Task Analysis: Principles and Practice

Task analysis, as it traditionally is conducted in Human-Computer Interaction (HCI), as well as in traditional systems design more generally, "is based on the idea that a description, containing all necessary information to build the computer application, can be made of the sequence of steps that it takes for a human being (in interaction with a computer) to conduct a task" (Bannon and Bødker, 1991). Such task analysis may contain a detailed description of each step of an expert planner's interaction with the computer application, e.g. as inputs and outputs. Similarly, in traditional systems design, the total information processing of the organisation is described this way.

What is often heard when a computer application fails to function according to the needs and wishes of the users, is that the initial task or flow analysis was "not sufficient". According to the work Bannon and Bødker (1991), rooted in the tradition of activity theory, something more needs to be included. This need is literally made visible in a video recorded with one of the project partners (Rauterberg et al., 1998b). In the first part of that video, we see an expert planner working with a Computer-Aided Design (CAD) system and discussing with a colleague. Many of the single actions and operations she - and they - carry out can hardly be described or captured in a written description.

Hence, a major issue is what information may be captured in the description as a task? This is a problem that has a decisive impact on the design process. When a task is described, we make observations or perhaps interview workers about what they are doing. However, often we are not capable of catching the tacit knowledge that is required in many skilled activities, or the fluent action in the actual work process. Bannon and Bødker (1991) estimate that we will never be able to give a full description of a task.

With these boundary conditions in mind, we report on our practice of task analysis with the four industrial project partners. Our task analysis investigated the domain of pre-CAD and CAD-based collaborative planning. This means, we did not only study single-user computer-mediated planning, but also traditional planning styles using two-
dimensional paper representations and discussions. First, we briefly presented the planning experts with the key features of the planned BUILD-IT system. Then, one or two experts from each project partners were questioned on the context and how they currently carry out planning tasks (Q1), how the new system may be of use (Q2), about the benefits expected (Q3), about user interface requirements (Q4), and about data interface requirements (Q5).

Before giving the answers, we describe each company briefly. The company profile mostly corresponds to the home page of the company. When nothing else is mentioned, the photos come from the same places.

**Outcome of Task Analysis Carried out during Autumn 1997**

**Project partner 1: Soudronic, Neftenbach, Switzerland (www.soudronic.ch)**

Soudronic offers weld systems being used world-wide for the joining of automotive components and products. Custom designed weld systems for specific applications are built on a turnkey basis. Soudronic is the only equipment suppliers for two of the most important welding technologies which are i) the laser and ii) the resistance roller seam welding process for tailored blanks.

![Figure 1: A tailored blank production facility (l.) and tailored blank sheets (r.).](image)

Tailored blanks is a new method to build body structures of cars. Two or more sheet metal pieces of different specification, thickness and shape are joined with laser or mash weld prior to being stamped into the desired car body part. The advantages are lighter cars, less curb weight, improved structural performance, stronger body, better absorption of crash energy, reduced parts count, easier to assemble and thus more cost effective.
Q1: Which characteristics of the planning task are to be supported (target activity)?
Elicitation of clients' needs is currently based on printed material is one characteristic. Another is a set of typical factors which have to be found, being maximum blanks size and width, blanks to be processed, thickness, and weld speed range. A final characteristic is given by marketing, being based on objective information as well as "intuitive" graphical representations and is directed towards CAD-specialists and planning experts.

Q2: Which potential uses are offered by BUILD-IT?
A potential use is in presentation of production plant solutions and computation of alternative solutions. Another use is for presence on the Internet and proliferation as a company employing virtual tools.

Q3: Which are the expected benefits from BUILD-IT?
Shorter time in offer authoring is an expected benefit. Another is visual representation in offers combining animation with key figures. A final benefit is automatic offer authoring through integration of standard solutions.

Q4: What are the user interface requirements to BUILD-IT?
Interactive input of parameters like material input and product output is one user interface requirement. Another is input possibility of CAD models or model parts combined with PDF data like sheet thickness. A final requirement is a transportable version.

Q5: What are the data interfaces requirements to BUILD-IT?
Current contact with clients is more based on Fax than CAD, making the use of Fax into a requirement. Specification and data sheets are communicated using Intranet and Email, making Intranet and Internet access into another data interface requirement. Internal communication in the company is mostly oral, however, this was not seen to have additional impact on the data interface requirements, since BUILD-IT is groupware, allowing for co-located collaborative planning.

Project partner 2: Mikron, Agno, Switzerland (www.mikron-tg.com)
Mikron Technology Group offers a) standard machines, special machines such as Multistart (Fig. 2, left), and key modules (Fig. 2, centre). Typical standard machines are milling machine centres. Typical special machines are high output machining systems like rotary transfer machines and linear transfer machining systems for gear pumps in dialysis (Fresenius) and parts for injection systems (Bosch). Typical key modules are high-performance cutting tools and spindles. The focus in this project was a special Multistar machine.
Q1: Which characteristics of the planning task are to be supported (target activity)?

The major characteristics is the responsibility for planning, being spread over three distinct departments: marketing, offer authoring, and development.

Q2: Which potential uses are offered by BUILD-IT?

A potential use is in a new project for rotary transfer machining, offering a system called Multifast\(^1\) (Fig. 2, right).
Another use is in the presentation of offers.
A final use lies in the planning of complex production plants using virtual models.

Q3: Which are the expected benefits from BUILD-IT?

An expected benefit is less travelling in the development phase.
Another is three-dimensional representation, which is seen as a good marketing argument.
A third is the mediation and support in internal discussions.
A final expected benefit lies in collaborative work and in large displays giving more overview than screens.

Q4: What are the user interface requirements to BUILD-IT?

One user interface requirement is a master-slave system to allow for networking.
Another is the ability for distributed work, typically taking place between the headquarters in Agno, Switzerland and a department in the USA.
Another is the possibility for transportable and laptop versions for marketing activities.
The access to a set of adjustable process security parameters is another.
A final user interface requirement is a database of previous offers, communicating with the BUILD-IT system directly or via Product Data Management (PDM) data.

---

\(^1\) This system is especially suited for multi-operation machining, being cost-efficient, offering high volume production, covering workpieces in sizes of up to 120 mm by 120 mm by 120 mm, and offering high precision and elevated production rates.
Q5: What are the data interfaces requirements to BUILD-IT?

One data interfaces requirement is the ability to use Euklid, Autocad, and Unigraphics, already in use at the headquarters in Agno, Switzerland. Another is the ability to support a separate CAD-standard being used in the USA. The possibility for customer driven design supported by PDM is another. A final data interfaces requirement is the integration of in-house programs for Microsoft Excel computations.

Project partner 3: Von Roll Inova, Zurich, Switzerland (www.vonroll.ch)

Von Roll Inova is specialised on environmental technology. It builds and runs thermal waste treatment plants and supplies technology and know-how for flue-gas purification and by-product recycling.

Figure 3: A thermal waste treatment plant from Von Roll Inova.

Q1: Which characteristics of the planning task are to be supported (target activity)?

A characteristic is the design of thermal waste treatment plants with a need to integrate know-how from architects, civil-engineers, consultants, and experts in process technology. Another is offer authoring of projects lasting more than two years and having a financial size of about 165,000 EUR.

Q2: Which potential uses are offered by BUILD-IT?

A potential use is in three-dimensional (3D) representation for internal meetings. Another use is in collaborative planning within the company and with clients.

Q3: Which are the expected benefits from BUILD-IT?

A benefit is expected from larger display, giving more overview than a screen. Another is a less techno-centric perspective to work processes. Another expected benefit is the reduction of complexity. Benefits are also expected from a natural interaction through co-location, thereby requiring less video-conferencing. Another expected benefit is an earlier discovery of mistakes in the planning process – leading to less costs later in the production cycle. Two final expected benefits are the visualisation of the development processes and greater number of people participating in that processes.
**Q4: What are the user interface requirements to BUILD-IT?**

A user interface requirement is the possibility to set model height. Another is the visualisation and handling of layers. Positioning models "in the air" and not only on the ground (fig. 3), combined with an analogue height-scale is another. A final user interface requirement is the access to different layers marked with different colours to ensure multi-layer interaction.

**Q5: What are the data interfaces requirements to BUILD-IT?**

A data interface requirement is the simultaneous interaction with two-dimensional plan view, side view, and height view. Another data interface requirement is the integration of Prime Medusa Software.

**Project partner 4: dai, Zurich, Switzerland (www.dai.ch)**

dai is a consulting company specialised in corporate design. The company develops models and strategies with their clients with the aim of establishing an unmistakable quality with the clients’ corporate culture. It promotes their clients’ effort in establishing a complete visual identity so that their performance and the position of their brand can be strengthened in the marked place. The company supports their client’s in all phases, from initial briefing to market introduction and assessment of the final effect.

![Figure 4: The company employs real (l.) and virtual (centre) prototyping to support and encourage decision making processes. It also creates logos and integrated graphic concepts (r.).](image)

**Q1: Which characteristics of the planning task are to be supported (target activity)?**

A major characteristic is a focus on interior office design. Another is the use of client interviews to elicit their production processes and their requirements. A final characteristic is the predominance of expert planning.

**Q2: Which potential uses are offered by BUILD-IT?**

A potential use is in communication between client-consultant and CAD expert. Another is the internal company use for pre-CAD planning.
Q3: Which are the expected benefits from BUILD-IT?

An expected benefit lies in the three-dimensional (3D) visualisation with potential clients who normally are not acquainted with thinking in two dimensions. Another lies in making a strong impression and statement with clients. Buying as an "event" and higher identification with the product is another benefit expected. The construction of different product versions is also expected. Another benefit is it enables hands on experience. A final benefit expected is the use of large displays, being seen as a major advantage.

Q4: What are the user interface requirements to BUILD-IT?

A user interface requirement is the communication with an internal USM CAD-system. Another is the quick exportation of a three-dimensional (3D) image as file or printed version. Another is the representation of an interaction with larger scenes. Easy-to-learn and easy-to-handle software is also an expected benefit. Since the price is critical, the possibility for leasing or renting of a system is required. A final user interface requirement is a portable version of the system.

Q5: What are the data interfaces requirements to BUILD-IT?

A data interface requirement is USM CAD-System, being a variety of Autocad. A more general data interface requirement, is that the system should be portable between different operating systems and platforms.

Other Project Partners

For the remaining project partners, no systematic task analysis was carried out. However, we list these partners since their contribution to the project was significant and sometimes decisive.

Project partner 5: Tellware, Zurich, Switzerland (www.tellware.com)

The BUILD-IT technology was invented, designed, developed, and tested in different research projects at ETH Zurich. Based on the know-how established and above all, the ready-to-use interactive system, a spin-off company, called Tellware (GmbH), was established during the project. Tellware's interest is the marketing and support of the BUILD-IT technology. The spin-off company also sells information- and projection-technology, e.g. high resolution video beamers. Finally, the company offers several services and add-ons for the use and operation of the BUILD-IT system.

Project contribution: Consulting, know-how in visualisation, and market research.
Project partner 6: Perspectix, Zurich, Switzerland (www.perspectix.com)

Perspectix facilitates the marketing and sales of modular products via electronic distribution channels through technologically advanced, simple-to-operate 3D product configurators. With its enabling technology Perspectix turns product presentations, "what if" analyses, and custom-tailored orders into an enjoyable and interactive experience fully controlled by the buyer.

**Project contribution:** The MET++ software was the multimedia framework for the BUILD-IT prototype. Perspectix also provided consulting in setting up and maintaining parts of the multimedia software in BUILD-IT.

Project partner 7: Elektro Projekt, Weingarten, Germany (www.epelektroprojekt.de)

E.P. Elektro Projekt products and services primarily address the needs of customers in fields of automation and electronic components - mostly for industrial use. Tasks of this sophisticated level require employees with an extraordinary know-how in the mentioned fields.

**Project contribution:** Various physical tools (which were wireless and tethered) and software drivers for height adjustment in the BUILD-IT system.

Project partner 8: USM Schärer Söhne Münisingen, Switzerland (www.usm.com)

USM Schärer Söhne offers consulting in planning, call-centre tables and chairs, consulting in acoustics, shelving systems, walling systems, and office system units.

**Project contribution:** Virtual models of USM furniture and real table and chairs for the BUILD-IT system.

**Outcome and Interpretation of our Task Analysis**

Even within single professions, we found very different deployment and use of supportive technology for planning activities. We found that for BUILD-IT — as for any three-dimensional (3D) CAD tool — it was necessary to find an efficient way to handle two-dimensional information without having to measure existing buildings and produce paper drawing. In our task analysis, we often observed a sharp division of labour among potential end-users. We realised that the use of BUILD-IT not only could be beneficial in the contact with clients, but also as an internal communication and planning tool. In particular, we decided that our major aim would be to support the design of complex production plants by providing easy and direct interaction with 3D information. We expected that by offering such interaction, planning experts may avoid planning mistakes or, at least, discover their mistakes earlier in the planning process.
Clients of production plants only seldom have sufficient skills to imagine what the outcome of a planning process will be. The reason might be that they are more used to an immediate experience in a physical environment, and that the identification with symbolic information is considerably smaller. The answers we received, indicated that the new interaction technology could improve this lack of identification. The possibility for many participants to work at one table, to discuss, and to try out different alternatives, makes BUILD-IT a tool for collaborative planning. Hence, group work can be achieved without having to modify and refurbish offices.

In our task analysis, we registered a specific need for navigation of the virtual scene within a planning session. An original idea to overcome this need, was the use of an animated human model to control the side view so that system users would see what the virtual model would see. With several models, the human model setting the side view would light up with a "red hat". However, in current multimedia frameworks, humans are only reactive, and cannot be animated. This means that virtual models of humans were not satisfactory because they could only be modelled in terms of machine delays such as loading- and production times. Therefore, the first approaches to navigation of the virtual scene instead would employ a virtual camera. Such cameras simulate what a person at the camera position would see.

**Follow-up to the task analysis**

After our task-analysis, the first BUILD-IT prototype was tried out with designers from companies producing assembly lines and plants. These designers regularly see their customers, and are aware of what a mediating tool should look like. The tests showed that the system is intuitive and enjoyable to use as well as easy to learn. Most designers were able to assemble virtual plants after only a few minutes of introduction. Some typical user comments were:

- "The concept phase is especially important in plant design since the customer must be involved in a direct manner. Often, partners using different languages sit at the same table. This novel interaction technique will be a means for completing this phase efficiently."
- "This is a general improvement of the interface to the customer, in the offering phase as well as during the project, especially in simultaneous engineering projects."
- "The use of this novel interaction technique will lead to simplification, acceleration and reduction of the iterative steps in the start-up and concept phase of a plant construction project".
3 Setting the Initial Design Agenda through a User Study

This section describes a user study carried out early in the project. The aim of the study was twofold. In the first part of the study (Part 1), we set out to know more about the design of physical handles, so called bricks. In the second part of the study (Part 2), we wanted to systemize feedback from participants trying out the BUILD-IT system in an informal evaluation. In both parts, our examination was based on two strategies, namely to:

- Examine user behaviour, and
- Record subjective statements and preferences.

Participants

Twelve participants took part in the user study, all going through part 1 and 2. Seven of them had no previous experience with Computer-Aided Design (CAD) systems and only one was currently active in the field of production plant layout. Two of them were active in the development of CAD systems. All participants were acquainted with the use of computers and some were acquainted with computer-supported planning tools. The participants were not paid.

Procedure

In the first parts, twelve different bricks were tried out, and participants selected one preferred. This brick was used in the second part of the experiment. No particular task was given. Rather, the participants could explore and play with the system. The participants were asked about their expectation to the BUILD-IT system before use. All participants were asked to rank the twelve different forms before and after having used them. Participants could ask questions.

Outcome of part 1: Exploring physical handles - brick forms

The initial motivation to study alternative forms for physical handles, was that user often grasped the bricks from above, instead of from the side. Thereby, the reflective material on the top of the brick was not seen by the image processing software and the virtual model which the participants wanted could not be selected or, if already selected, was deselected. Hence, we designed eight alternative forms where the part of the alternatives would invite users to grasp them differently. This was reached by offering handles being extensions of the main form or by mounting the reflective area at a higher position then were the handle should be grasped. Factors tried out in the alternative brick forms were:

---

2 Prof. Max Engeli, CAD, ETH Z, is a CAD expert and has developed the Euklid system. Dr. Joe Weiss, IHA, ETH Z, is an ergonomic expert who is an expert on design of tangible devices, such as Six-Degree-Of-Freedom (6DOF) mouses.
Circular vs. rectangular, container vs. non-container, and different heights. The different forms and the outcome of this ranking is shown in Fig. 1, numerically in Table 1.

<table>
<thead>
<tr>
<th>Prototypical brick</th>
<th>Rank before use</th>
<th>Rank after use</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modelled brick with handle</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>Plastic brick</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>Classic metal brick</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Part of matchbox, container-principle</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>Petri dish</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Ball-form</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>Chess pawn</td>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>Truck lifter car</td>
<td>7</td>
<td>8</td>
</tr>
</tbody>
</table>

Fig. 1: The prototypical bricks which were tested, and their average rank before and after use.

Subjective statements and preferences
As criteria for their choice, participants mentioned handling quality, clarity, aesthetic, weight, and risk of unwanted de-selection of the different forms. They told that before using the different bricks, such factors as aesthetics was of particular importance. After having used the different bricks, weight, and risk of unwanted de-selection, were more decisive factors. Hence, the favourites turned out to be the classical metal brick and the Petri dish. One participant made this clear by the following statement: "When using
highly technical devices, one is used to a technical look". As for handling of the virtual camera, many participants would have liked to have a specialised brick with an orientation and an indication of the direction in which the camera sees. The truck lifter car got a low ranking since its ability in selecting virtual models was rather poor and it was described as non-ergonomic.

Observations
Altogether, we observed that surface quality, material, and friction to the table were important design factors. We observed that participants easily learned not to grasp the handles from above, but from the side, to avoid unwanted de-selection. When participants tried to position models exactly, grasping the bricks from above was observed even more seldom, and this behaviour was observed more often with the acquainted CAD-users. Other participants rather grasped the brick from above and lifted it off the table.

Concluding remarks on the user study with respect to brick design
A set of guidelines for brick design could be formulated as follows:

• The brick should be circular, edges are irritating in use,
• The quality of surface, material, friction to the table, and weight have major importance,
• Bricks have to be professionally designed,
• In a highly technical application, one is used to find technically designed artefacts,
• Participants quickly learn that bricks should not be grasped from above. When virtual models must be positioned with high precision, the chance that bricks a grasped from above is even less,
• Specialised bricks or bricks with orientation, e.g. a camera with an objective, could be of interest,
• Participants with no planning experience seem to grasp the brick and to lift it (which is not as foreseen), whereas mouse users rather grasp and move the brick on the table surface,
• The truck lifter car is moved in the orthogonal direction to its driving direction and requires two hands in order to deselect the virtual model, and
• Medium-sized Petri dish makes sense. A Petri dish needs an edge with a certain height to be easily grasped.

Outcome of part 2: Usability of the BUILD-IT System
In the subsequent evaluation of BUILD-IT, the participants described it as very stimulating. In handling quality, the system was described with statements ranking from easy to relatively flexible. Most Participants said that they were encouraged to explore the virtual scene, and that it was easy to learn and would be easy to instruct to someone else. Participants gave poor notes on the transparency of the system and the clarity of the different icons, i.e., representations of models and tools in the virtual storage space.
Many participants even though that it was not possible to understand the icons in the menu without a description. Another lack of transparency was experienced with the unintended scrolling. As participants placed the brick at the edge of the plan view, the system started to scroll spontaneously. When this unwanted scrolling should be corrected, several corrections were required to get back to the initial state.

Experts from Computer-Aided Design (CAD) and from the field of ergonomics directed particular attention to the fact that the system is too slow for professional planning purposes. They also claimed that the system is not sufficiently intuitive nor refined. When we observed participants working with the system, we noticed that they adapted their interaction style to the latency of the system and thereby reacted less spontaneously. Moreover, their movements became shorter and less fluid, still, exact positioning proved to be difficult. With all the consequences of the system latency, we declared improved speed a permanent priority in our future work.

Participants mentioned several wishes which would enrich system functionality, such as further camera options (i.e. navigation), the option to let two virtual models connect into one composite mode (i.e. snapping), and the possibility to make one or several steps back (i.e. an undo-function). Some of these functions were already implemented at that point, such as passive scrolling\(^3\) of the plan view and a simple camera for the side view. However, these were very simplistic and minimal options for navigation, and their handling was not satisfactory. Based on these observations, we decided to seek a more active way of scrolling the plan view\(^4\) and a more elaborate way of navigating the side view\(^5\). Finally, participants also wanted icons (models and functions situated in the virtual storage space - or menu) to be easier to understand.

---

3 Passive scrolling implies that a brick is positioned at one of the edges of the plan view. After a short delay, the view scrolls along the orthogonal to the edge where the brick resides. As soon as the brick is removed, scrolling stops. The drawback with this kind of scrolling, was that it lacked of direct control.

4 Active scrolling in the plan view was implemented in the *GroundCatcher*, later in the *FrameCatcher*. These alternative navigation tools for the plan view were compared in an experiment, and the *GroundCatcher* was preferred by users, over *FrameCatcher*.

5 This was implemented in the *EyeCatcher*, in the *Camera*, and in the *ViewFrame*. The two latter methods were compared in an experiment. *Camera* turned to be i) more efficient and ii) preferred by users, over *ViewFrame*. 
<table>
<thead>
<tr>
<th>Observed symptom</th>
<th>Suggested solution</th>
</tr>
</thead>
</table>
| Participants move the brick quicker than the image follows, they complain about the system being slow and show surprised reactions | Shorten the reaction time  
Do not show the movement but only the final position. |
| Participants slow down and try to be exact and complain that it is very difficult to, for example, place a model next to another one or in a corner | Introduce a snap function. |
| Model unexpectedly turns 180 degrees at de-selecting |  |
| After having arranged two or more models, participants wanted to group them in order to move the whole sample | Introduce a grouping function |
| Particularly CAD users lack the comfort of different functions and numerical input |  |
| Participants tried to incline the brick to look into the room from a different angle in the height view |  |
|Scrolling happened to participants surprisingly when they hold the brick over the "dangerous" area |  |
| When intending to scroll, it is difficult to get to the requested point because of the time delay; it usually needs at least one correction |  |
| For the uninitiated, the furniture in the menu is not self explanatory: what it is and what its orientation is was not given | Give text or a 3D view |
| On the quantitative scale, the (no existing) help function and transparency of the system have been judged medium, the rest was fairly good |  |
| Brick form: some participants complained about the edges of the rectangular bricks, others missed orientation with the circular bricks |  |
| Participants sometime expect the brick to have a definite connection to the model, but, sometimes, camera turns unexpectedly 180 degrees |  |

Table 1: Participant remarks and possible solution suggested by the project team.

The statements given by planning experts after having used the BUILD-IT system were:

- Reduce latency and do not show intermediate steps in a fluid brick movement. Only carry out the operations which make sense for human needs using Selspot-based motion analysis.

- On one side, participants adapted their interaction style to system latency and thereby become less spontaneous. On the other side, participants would prefer to be exact, but find this difficult due to system latency.
• For semi-professional users in client-support, more functions coming from CAD-systems are wanted. This means that we need to know more about how CAD-systems function. Even if the system requires some training to be efficiently used, this is not a problem for professional users. These users do not find it of prime importance that the system can be used right away, without any training.

• Commands have to be orthogonal and not overlapping. This requires a well-founded, logical concept for the interaction design.

• In professional product development, planners are used to a broad concept, then they decide what is useful for the marketplace, what to implement, etc. They work with two-dimensional (2D) models before taking this decision, and only later in the process with three-dimensional (3D) models. This mode of development must be considered carefully to assure that it will be supported by the system.

• Help functions and system transparency received negative critique.

The missing features were:

• Passive scrolling starts too easily and there is no active control. As soon as a brick is placed at an edge of the plan view, scrolling may start without it being wanted. Thereby, users have to make scroll corrections in the opposite direction, placing the brick at the opposite edge. However, it proved difficult not to correct too much/little. Therefore, passive scrolling should be possible to shut on/off in the virtual storage space (menu).

• The furniture in the menu cannot be recognised as such. Therefore, they should be shown with more 3D features or be annotated.

• Camera may disappear suddenly, and the reason is not clear.6

• The use of the height slice is not transparent enough, it requires some explanation. It also requires that the camera is de-activated before use.

• By selection and de-selection of a model, the model moves spontaneously, i.e. without moving the brick. This happens particularly often with the camera.

The new functions required were:

• Undo function.

• A swap function so that plan or side view is projected on the table.

• As an add-on to the camera: zoom and tilt handling by a physical control handle, combined with an undo function for the setting of zoom and tilt.

• Modification of the scale in the plan view.

• Snapping to grids

• Display of fewer of the grids, e.g. half of what is currently displayed.

---

6 This problem could be localised as a division-by-zero, i.e. a low-level error in the code of the system. It was then fixed.
• Setting of model angle could be step-wise, e.g. in steps of 5 degrees. This could be set by using an alphanumerically controlled keyboard with a display of the angle set.

• Snapping function, so that models connect to walls (e.g. in kitchen design) or to other models.

• Speed control of the passive scrolling function.

• Grouping of models to allow for handling of multiple objects at a time.

Concluding remarks on the user study with respect to the BUILD-IT system

The original aim with the BUILD-IT system - to create an easy-to-handle interface to CAD models - inviting clients to a play-like interaction style in their planning process - was considered as largely fulfilled. The lay users were highly satisfied. Experts in CAD-based planning noticed that "it is not necessarily of prime importance to produce drawings within a minimum of time". Since CAD experts could live with the latency of the system, we went on, testing which new functionality should be developed in the next steps. In this process, we set a priority to have close contact with CAD experts and to undertake experiments with groups of users and not only with single users.
PART II:

THEORETICAL GROUNDING AND EMPIRICAL INVESTIGATIONS - PAPER AND VIDEO PUBLICATIONS
Physical and Virtual Tools: Activity Theory Applied to the Design of Groupware

Abstract. Activity theory is based on the concept of tools mediating between subjects and objects. In this theory, an individual's creative interaction with his or her surroundings can result in the production of tools. When an individual's mental processes are exteriorized in the form of tools - termed objectification - they become more accessible to other people and are therefore useful for social interaction. This paper shows how our understanding of activity theory has shaped our design philosophy for groupware and how we have applied it. Our design philosophy and practice is exemplified by a description of the BUILD-IT system. This is an Augmented Reality system we developed to enhance group work; it is a kind of graspable groupware which supports cooperative planning. The system allows a group of people, co-located around a table, to interact, by means of physical bricks, with models in a virtual three-dimensional (3D) setting. Guided by task analysis, a set of specific tools for different 3D planning and configuration tasks was implemented as part of this system. We investigate both physical and virtual tools. These tools allow users to adjust model height, viewpoint, and scale of the virtual setting. Finally, our design practice is summarized in a set of design guidelines. Based on these guidelines, we reflect on our own design practice and the usefulness of activity theory for design.

Key words: Design, activity theory, Augmented Reality, Virtual Reality, graspable, groupware, objectification, physical tools, virtual tools, co-located interaction, cooperation, planning, configuration, social, computer.

1. Introduction

The aim of this paper is to explain and illustrate our design philosophy for developing graspable groupware. Our philosophy is based mainly on the concepts of tools and exteriorization found in Leont’ev’s (1978, 1981) and Engeström’s (1990, 1996) work on activity theory. Our practice rests on Bodker’s (1991) and Kaptelinin’s (1996) work, where they apply activity theory to human-computer interaction. In our construction of a system we employed a recent technology called Augmented Reality (AR). Before we began to develop the system, we studied users needs by employing task analysis methodology.

According to Leont’ev, not only is activity shaped by physical surroundings, activity in turn shapes the surroundings. When activity shapes those surroundings what happens is that internal mental activity materializes into artifacts. This process of turning mental activity into an object or objectification is what Leont’ev called exteriorization. While it is obvious that for any individual the moment of exteriorization is an important step in his or her creative design activity, it is perhaps less obvious that this is a crucial step in...
making ideas accessible to others. From this perspective exteriorization is an important social moment which supports mutual understanding in a collective creative design process.

In our design philosophy we take account of the physical and social surroundings as well as the physical and mental faculties of human beings. We draw on Bødker’s (1991) and Kaptelinin’s (1996) work as a basis for our design process. While we do not consider aspects of human consciousness and emotionality in human development, we note the possible connection between these aspects and human-computer interaction (Nardi, 1996b).

A vital part of our design philosophy is the tradition of AR, which enriches natural communication with virtual features. Backed up by activity theory and the usage of AR, we developed groupware for layout planning and configuration tasks; this groupware is called the BUILD-IT system (Rauterberg et al., 1997a, 1997b, 1998; Fjeld et al., 1998a, 1998b, 1999c, 1999a, 1999b, 2000). This system enables end-users, grouped around a table, to cooperate in the active design of a virtual setting, thus supporting co-located, instead of distributed, interaction. The multi-user functionality of BUILD-IT overcomes a serious drawback often seen with computer-supported cooperative work (CSCW) systems, namely that they are based on single-user applications (Grudin, 1988). We believe that co-location is an indispensable factor for the early stages of a complex planning process. Input and output, however, can be prepared and further developed off-line, using a conventional Computer-Aided Design (CAD) system. Other major projects where graspable groupware was constructed to support planning processes are the metaDESK (Ullmer and Ishii, 1997) and the Environment and Discovery Collaboratory (EDC) (Arias et al., 2000).

Section 2 introduces our theoretical background, which stems from activity theory; we discuss concepts such as tools, objectification and collective action regulation. The difference between goal-directed pragmatic action and exploratory epistemic action is emphasized. Section 3 describes our design philosophy in terms of AR. Task analysis and the incorporation of both physical and virtual tools are two methods which we show to emerge naturally out of our design philosophy. Section 4 demonstrates how our design philosophy was used to develop the groupware called BUILD-IT. Our design philosophy is subsequently applied to the design of tools for graspable groupware. We describe the basic principles of human interaction with graspable tools and show how such tools are developed according to the results of task analysis, whereby we focus on the challenges and problems we encountered. In Section 5, the benefits of activity theory for groupware design are discussed and a set of design guidelines is outlined.

2. The concept of tools in activity theory
To provide a theoretical background, we show how our understanding of activity theory has shaped our goals and the design process of AR groupware. First, our account of the tool concept and objectification is given. Then, collective action regulation is explained. Finally, we introduce two types of complete action regulation cycles for goal-directed pragmatic action and for exploratory epistemic action.

2.1 Tools and Objectifications
In the most general sense, activity means a subject’s interaction with his or her surroundings. Modern activity theory originated from Soviet cultural-historical
psychology (Vygotsky, 1978; Leont’ev, 1978, 1981), which in turn is rooted in both eighteenth and nineteenth century classical German philosophy - from Hegel’s idealism to the historical materialism of Marx and Engels, in which the concept of activity was extensively elaborated. These roots are quite unfamiliar to most Anglo-American readers and have therefore been partly neglected (Kuutti, 1996). Yet, Engeström (1991) claims that activity theory today is transcending these origins, becoming truly international and multidisciplinary. Two results of this development are Engeström and Middleton (1996) and Nardi (1996a).

Fundamental to modern activity theory is the idea that the development of thoughts and cognitive activity requires social interaction and exchange with a physical environment. Via the process of internalization, social interaction turns into mental activity. Handling ever more abstract objects and concepts is part of an individual’s cognitive development. Nevertheless, the physical environment remains important, since it is used for the externalization of thoughts and as external memory. This is particularly important for the abstract planning and configuration tasks we focus on in this paper.

Individuals are confronted with tasks that life puts in front of them and they use artifacts as tools or create tools out of their understanding. These tools then become part of the cultural context of other people. A tool mediates an activity, thereby connecting a human being, not only to the world of objects - his or her physical surroundings - but also to other human beings. At the same time the use of a tool appropriates the collective experience of humanity embodied in that tool (Leont’ev, 1982). In this sense we view the developmental processes of human beings, their physical surroundings and social culture as co-evolutionary. As Engeström (1991) puts it: “The idea is that humans can control their own behavior – not ‘from the inside’, on the basis of biological urges, but from the outside, using and creating artifacts”.

For planning activities Engeström’s idea has been applied to artifacts and tools, resulting in sketches, documents, and three-dimensional (3D) objects or devices. (At this point in our research process, we view Engeström’s artifacts and tools as corresponding to Leont’ev’s (1978) objectifications of physical nature.) Hacker et al. (1998) describe the importance of grasping design ideas by sketching and by low-cost prototyping. Such methods may help to achieve design results in a faster and better way than by using abstract design processes. Physical interaction handles, for instance bricks (Fitzmaurice et al., 1995), can be seen as physical devices for exteriorization in a planning process.

To illustrate the tradition of tool development in activity theory, a historic example might be of interest. In the historical collection of the ETH library, we found that mathematicians of the eighteenth century (Diderot and D’Alembert, 1778) employed a variety of physical tools (Fig. 1). Modern mathematics has become even more an abstract field of study.
However, the view of tools presented above can also impose some limitations on the potential applications of activity theory. In Virtual Reality (VR) “the border between a tool and reality is rather unclear; information technology can provide the user not only with representations of objects of reality but also with a sort of reality as such, which does not obviously represent anything else and is intended to be just one more environment with which the individual interacts” (Kaptelinin, 1996, p. 64). This unclear border is a problem VR presents to activity theory; it might be solved by enriching activity theory’s basic principles with new ideas from cultural-historical traditions or other approaches for studying the use of artifacts.

One answer may be found in the distributed cognition approach, in which internal and external representations of artifacts are examined (Flor and Hutchins, 1991; Hutchins 1991). As in activity theory, this approach describes how we may take advantage of artifacts designed by others in collaborative manipulation. Compared with activity theory, “distributed cognition has taken most seriously the study of persistent structures, especially artifacts” (Nardi, 1996c, p. 85). However, the two closely related frameworks show one distinct difference. In activity theory artifacts mediate human thought and human behavior and there is no intrinsic symmetry between people and their tools. Based on the 19th century debates on epistemology and on what a human being is, gaining knowledge can be seen as an individual process, a process of knowing, which can only take place in an individual. In contrast, distributed cognition puts people and things at the same level; they are both ‘agents’ in a system (Nardi, 1996c, p. 86). Hence, the distributed cognition approach ignores the faculties of human beings not found within computers, like motive, emotionality, and consciousness. It also ignores for computers their non-human traits, namely their ability to execute programs in a precise and predictable manner. By focusing on a common capability of humans and computers, much is lost on both sides.

Another answer is found in Bødker’s (1996, pp. 151-152) characterization of the different focuses in the use activity:

- "The physical aspects - support for operations toward the computer application as a physical object. The physical aspects are the conditions for the physical handling of the artifact. ...
- The handling aspects - support for operations toward the computer application. ... The handling aspects are the conditions for transparency of the artifact that allow the user to focus on the ‘real’ objects and subjects of the activity. ...
- The subject/object-directed aspects - the conditions for operations directed toward objects or subjects that we deal with ‘in’ the artifact or through the artifact. ..."
In our work we understand the physical aspects as how to use the hardware to operate the groupware, the handling aspects as how to operate the groupware, and the subject/object-directed aspects as how to use groupware to solve a task.

Bødker talks about ‘real’ objects and subjects of the activity. Working with VR, the object of the activity is represented by a virtual world. What a subject situated within that virtual world would see, is represented by a virtual viewpoint. By interpreting the term ‘real’ in the handling aspects as ‘virtual’, we may overcome the limitations pointed out by Kaptelinin (1996), i.e. we may clearly define the border between tool and reality. This interpretation takes on particular importance when we develop our so-called navigation methods (Section 4.3.3), which are purely based on the handling aspects. In a more general way, we will use all three focuses in the use activity to structure the description of our design process (Section 4).

2.2 Common objectification

In more recent developments, the scope of activity theory has broadened to encompass interaction within a community (Engeström, 1991). Since our design philosophy is centered on supporting co-located groups, this new development has been of particular interest to us. As pointed out above, a tool connects an individual to other human beings by mediating activity, thereby becoming part of a cultural context. Of particular interest to us are tools that are made or used by groups.

The individual processes of goal setting, planning, and action are transferred into collective action regulation (Weber, 1997; 1999). This transfer refers to coordination and allocation processes within a group. Weber introduces the term common objectification to mean materialization of collective action regulation, integrating different schools of activity theory. “The process of common objectification is understood as a process by which all (or several) members of a workgroup mutually transfer their individual knowledge, expertise, and experience into a material form. By doing this, they make their materialized knowledge available to other group members” (Weber, 1999, p. 13). First applied to industrial work groups, common objectification was also found to be of great importance in design and planning teams because planning tasks require even more mutual exchange of knowledge among experts (Lauche et al., 1999). The results of a process of common objectification can take the form of reports, design documents, prototypes, drawings, or software. Therefore, it is part of our design philosophy to facilitate the creation of such results among a group of users.

2.3 Goal-directed and exploratory action

General models for goal-based problem solving were first suggested by Newell and Simon (1972). Drawing from the same sources as activity theory, action theory has focused on goal-directed action at work (Hacker, 1994; Frese and Zapf, 1994; Volpert et al, 1989; Frese and Sabini, 1985). Hacker (1998) introduced the notion of complete action cycle (Fig. 2) for goal-directed pragmatic action, consisting of:

1. setting the next (or first) goal in performing a task,
2. planning according to the conditions of execution, including selection of tools and preparation of actions necessary for goal attainment,
3. physical (or even mental) performance, and
4. control according to the set goal via different sources of feedback.
In our work, however, we have realized that both our own design process and the planning task for which we are designing the groupware are not exclusively goal-directed; they also have exploratory elements. Exploratory epistemic actions (Kirsh and Maglio, 1994) are performed to unveil hidden information or to gain insight that would otherwise require a great deal of mental computation. Exploratory epistemic action means that no specific goal is available for initial action. Only after the receipt of feedback, which gives information on the means available, can a goal be generated. Based on this goal, a new planning stage and a new action phase can be initiated. Thereby, an alternative kind of complete action cycle emerges (Fig. 3).

Our aim will be to design a system which fluently mediates both kinds of actions, goal-directed pragmatic and exploratory epistemic action.

3. Design philosophy
In this section we present our design philosophy, which is based on coinciding action and perception spaces, in its relation to Augmented Reality (AR). Task analysis and a shifting design focus between physical and virtual tools are described as two further design methods emerging from our philosophy.

3.1. AUGMENTED REALITY
Computer-supported cooperative work (CSCW) has enabled distant and asynchronous communication between people and has helped build ‘bridges’ in our global economy. This has brought about many well-advertised advantages, ranging from economic benefit to less status-oriented network communication. However, with many CSCW systems, users hardly interact with their physical environment. They deal only with virtual objects, which is also the case for most single-user applications. Sometimes users
are even embedded in a fully virtual world, unable to draw on any attributes of the tangible physical world. Much of the users’ mental capacity is employed to adapt to the virtual world, leaving less capacity for actual task solving.

An alternative approach offered by AR is to bring the virtual world of computers into the physical world of everyday human activity. This approach includes aspects of natural communication which serve as mediators for mutual understanding: eye-contact, body language, and physical object handling. It is non-intrusive (Vince et al., 1999), using no gloves or helmets, and thereby respects body-space (Rauterberg, 1999). At the same time users can still draw on the advantages of a virtually enriched world, which is of particular importance to planning tasks. The activity of planning is mainly ‘virtual’ because it involves reflecting on and modifying objects that will only exist in the future. Virtual models of these objects can be more easily changed than physical models. They can be stored in external computer memory and can be visualized for interaction purposes. Thus both physical and virtual models have their rightful place in a planning process. A specific aim of our project is to study ways to integrate computer-mediated activity into the physical world. This is how we came to work within the tradition of AR, where computer-generated models and physical objects are handled in one workspace.

Figure 4. AR means that a physical workspace (left) is augmented, or enriched, by a virtual world (right). Even when users interact with the projected, virtual models, they do not leave the physical context (e.g. sketching) and tools (e.g. pencil).

AR was first described by Wellner et al. (1993) and Mackay et al. (1995). The goal of AR is to “allow users to continue to use the ordinary, everyday objects they encounter in their daily work and then to enhance or augment them with functionality from the computer” (Mackay et al., 1995). According to Mackay, AR means that computer information is projected onto drawings so that users can interact with both the projected information and the paper drawing. The first brick-based AR system was described by Fitzmaurice et al. (1995). A more recent example of how AR can be used to support urban planning is given by Arias et al. (2000). Their focus is to create “shared understanding among various stakeholders... [by] ...creating objects-to-think-with in collaborative design activities.” (Arias et al., 2000). Figure 4 illustrates some of these principles. Pen-based input has been studied extensively; in this paper we look at bricks as input medium.

3.2. Action and perception

In his ‘Writings on the philosophy of making’, Aicher (1996) criticizes the lack of doing by planners and the overemphasis on the reduction of real-life to inner, rational activity. He advocates a closer connection between action and mental reflection, indicating a need to enable users to bring together action and reflection in human computer interaction. For our design practice this means to “create possibilities for the users to try out the user interface through use, not only through reflection” (Bodker, 1991, p. 148).
Hence, people must be able to employ their everyday motor faculties in their interaction with computers. We set out to create an interface with various input modes and tactile interaction. This demanded a user interface able to interpret a wide range of human expressions. Such interfaces need powerful computer vision methods (Rauterberg et al., 1997b).

A further important aspect of our design philosophy is the coincidence of action and perception space (Rauterberg, 1995). When handling physical objects, the space in which we act coincides with the space from which we receive (visual) feedback: we can see what we do. This is not the case for the handling of virtual models with a mouse-keyboard-screen interface, where there is a separation between action and perception spaces. Input and output devices are separated. To overcome this separation, Rauterberg (1995) suggested an alternative approach to interface design, an approach where action space and perception space coincide. Support is given by Hacker and Clauss (1976), who found that performance increases when task relevant information is offered in the same space where action takes place. This principle not only applies to visual but also to haptic or tactile feedback. Akamatsu and MacKenzie (1996) showed how including tactile feedback might improve computer-involved, task-solving performance.

3.3. TASK ANALYSIS AS PART OF THE DESIGN PROCESS

An integral part of our design philosophy is to gain a detailed understanding of the target activity: cooperative planning. For CSCW it is of particular importance to investigate which part of an activity is of a genuine cooperative nature and which part merely employs computers for individual work.

In the tradition of socio-technical systems theory (Emery, 1959), a task is seen as the link between the human/social system and the computer/technical system. Based on this tradition, it is not sufficient to take end-user expectations and preferences as guidelines for future design. Such a strategy would focus on what is already available and would be of little use for innovative breakthroughs. For instance, end-users would project their understanding of existing CAD functionality onto new interfaces without taking into consideration the whole socio-technical system they are part of. Therefore, instead of taking account of current personal and technical constraints, our analysis focuses on the task.

The strategy of our task analysis was to interview planning experts and observe their working processes, then to elicit difficulties with the predominant socio-technical system of meetings, drawings, and single-user CAD applications. The experts explained their jobs and provided examples of objectifications they used. Also, interaction with colleagues and customers was part of the interview. The results were used to make decisions about future directions in our design.

3.4. Exteriorization and interiorization of tools

Through our own design practice (Section 4.2) we made the following observations, valid from both a designer and from an end-user point of view. Before users had gained experience with physical tools (first step), user operation of the system was slow. Section 4 offers a few examples, for instance the initial bricks (Fig. 8, right, in center) and the first generation virtual tool for changing model height (Fig. 9). Once physical tools were available and mastered (second step), they became beneficial to task solving. This was the case for the more recent bricks (Fig. 8, right) and the physical tools for changing model height (Fig. 10). Based on our understanding of activity theory, we noticed that the physical tools used in the second step were an exteriorization of the
knowledge we gained from our collective design activity (Section 2.2). Based on the experience gained with physical tools, we even saw one case where a second generation virtual tool could replace the physical ones (third step). This was the case for an elaborated virtual tool for changing model height (Figs. 11-12). In the transition from the second to the third step the knowledge acquired using the physical tool was partly *internalized* (e.g. by giving our design team a common reference about the use of physical tools, even though not in use), partly implemented in the virtual tool (e.g. by mapping physical buttons onto virtual handles).

We observed that our experience is close to an idea of Kaptelinin (1996, p. 62), describing a three-step development of tool usage:

1. "The initial phase, when performance is the same with and without a tool because the tool is not mastered well enough to provide any benefits,
2. the intermediate stage, when aided performance is superior to unaided performance, and
3. a final stage, when performance is the same with and without the tool but now because the tool-mediated activity is internalized and the external tool (such as a checklist or a visualization of complex data) is no longer needed."

Since the three steps we observed and Kaptelinin’s idea have important points in common, we employed his idea as a justification for a repeatedly shifting design focus between virtual and physical tools.
4. Design example: the BUILD-IT system

BUILD-IT is a planning tool based on state-of-the-art computer vision technology (Rauterberg et al., 1997a, 1997b, 1998; Fjeld et al., 1998a, 1998b, 1998c, 1999a, 1999b, 2000). This system (Fig. 5) enables its users to cooperate in a virtual environment for planning a real-world object, such as a room, a school, a factory, or a piazza. Grouped around a table and employing tangible physical bricks, users can select and manipulate virtual models within the setting which they are planning. This use of physical bricks represent a new way of interacting. Tethered bricks (requiring wires or cables) were investigated in the Active Desk by Fitzmaurice et al., (1995). Fitzmaurice and Buxton (1997) later developed wireless bricks, detected by using a Wacom board. In the more recent use of tangible bricks, their surface is detected from above (Ishii and Ullmer, 1997; Ullmer and Ishii, 1997; Rauterberg et al., 1997a, Underkoffler and Ishii, 1998). It was shown that a brick-based interface is significantly easier to use and more intuitive than a mouse-keyboard-screen (Rauterberg et al., 1996).

![BUILD-IT system](image)

Figure 5. BUILD-IT consists of a rack, mirror, table, chairs, and a screen (top left). In addition to a high-end PC, the rack contains two beamers, a video camera, and a light-source. The system offers two perspectives of the same setting: a horizontal plan view for combined action and perception and a vertical side view (top right). Models projected in the plan view can be rotated and positioned using a brick (bottom left). Bimanual interaction is an essential part of the interaction concept (bottom right).

4.1 BASIC ASPECTS OF the BUILD-IT SYSTEM

Based on our discussion of tools (Section 2.1), we have chosen to structure the system description according the focuses in the use activity (Bødker, 1996, pp. 151-152). In this section (Section 4.1), we present all the basic aspects of the BUILD-IT system and its use. In the following section (Section 4.2) we focus on the design of tools, reflected by the physical aspects and the handling aspects. A detailed investigation of the subject/object-directed aspects belongs to future research.
4.1.1 Physical aspects of the system

In our system the position and orientation of the brick on the table top is determined by a computer vision system. Reflective material is applied to the top of each brick. The bricks reflect light from a light-source to a camera viewing the surface from above. Image processing software then recognizes the bricks and determines the two-dimensional (2D) position and orientation of each (Bichsel, 1997). This information is then used to control the groupware and hence update the image projected on the table. For the interaction taking place on the table, our technology respects the principle of coinciding action and perception spaces (Section 3.2). For the image projected on the screen, however, the same principle is not respected.

4.1.2 Handling aspects of the system

In BUILD-IT the users have two up-to-date views of the setting they are creating and manipulating at all times: the plan view and the side view. The plan view is the bird’s eye view from above - which is projected onto the table. The side view is projected onto a screen near the table. In the case of the side view, a virtual camera, which can be located either outside or inside the plan view, allows the users to choose from which position the side view is to be projected. The side view can also be zoomed. In the case of the plan view, the entire projection of the setting can be shifted from side to side, rotated, or zoomed. The plan view also contains a virtual storage space for models not in immediate use. It allows users to create multiple model instances. A model instance brought back to the storage space is deleted from the views. For all handling operations affecting virtual models users draw on basic, everyday manual skills - selecting, placing, rotating, re-positioning, and fixing.

In BUILD-IT mediation between users and the virtual world follows a cyclic order (Fig. 6). Users select a model by putting the brick at a model’s position. The model can be re-positioned, rotated, and fixed by simple brick manipulation. A model can be deselected by covering the brick. Then, another model is selected or the brick is left idle inside or outside the plan view. Using a material, hand-held brick, everyday manual patterns - like grasping, moving, rotating, and covering - are activated. Since all the steps in the cycle are reversible, the cost of making a mistake is low. Thus, exploratory epistemic and goal-directed pragmatic actions are equally supported, as described in Section 2.3. However, there is no possibility of undoing, so users must keep information about previous steps in their planning process. (More comments regarding undo appear later in Section 5.)
4.1.3 Subject/object-directed aspects of the system

The BUILD-IT system can be used by a single individual, but its full potential is realized as a mediator among members of a work group (Fig. 7). Basic usage of the system is acquired within minutes. Therefore, it may stimulate people possessing different sets of skills and/or different modes of knowledge to work closer together and thereby enhance their (verbal) exchange. Also, since the system forces people to work with shared resources, it has a capacity to reveal potential misunderstandings among them. A collective learning process can be triggered through the unifying workspace. This was actually experienced by our design team itself during system development work.

![Figure 7. Typical multi-user (left) and single-user (right) situations. Interaction and display take place in the plan view, whereas an additional perspective is offered by the side view.](image)

Although we have designed a co-located, multi-user groupware system, the physical and virtual tools presented in this paper may be important steps towards a multi-site, multi-user system and therefore represent a technological foundation for CSCW. Based on standard software and hardware, the system will allow for a distributed networking CSCW system in the near future. In our projection of such a system initial configuration will be transferred at start-up time, then only brick data will be transferred at real-time. This will allow the use of standard, commercial communication channels.

4.1.4 From CAD to BUILD-IT: A new tool for planning

In most companies CAD systems have replaced drawings and physical models. This change has helped speed up the design process by systematizing recurrent tasks and reducing the monotonous work of small changes. CAD systems are more than simple drawing tools; they offer interfaces to production monitoring and quality control. In CAD systems part prices or other meta-data can be stored with 3D data. However, not all planners are yet acquainted with CAD systems. This results in a division of labor between planners and CAD specialists, complicating the planning process. Another drawback of CAD systems is that they are not well suited for the early stages of a design process (Hacker et al., 1998) nor for person-to-person or group communication. This is partly because they are based on single-user applications (Grudin, 1988).

Our task analysis involved 16 planning experts from the machinery and processing industries and from the field of architecture. As a result, we distinguished three possible domains of use for the BUILD-IT system:

1. Cooperative planning among a team of experts: For cooperative work and concurrent engineering, the BUILD-IT system offers a multi-user interface. This interface may
enhance common understanding of an early planning process by offering professional tools, like advanced model height adjustment and full 3D navigation.

2. *Interactive planning with customers or clients*: In this domain the most important feature is that the customers can easily step into the early planning process and visualize their own ideas. Clients who are not used to planning - and thereby not used to 2D views - will be more at home with 3D images. The BUILD-IT system gives these users an easier access to 3D vision.

3. *Marketing*: For presentation and marketing purposes, tangible bricks may stimulate intuitive, play-like interaction. Potential clients using the system may be more involved than in conventional presentations, where interaction is fully controlled by a salesperson.

In all three domains BUILD-IT will not replace CAD systems. Rather, it may serve as a pre-CAD complement in the early stages of design. It might also stimulate the evolution of CAD systems.

4.2 DESIGN PRACTICE OF THE BUILD-IT SYSTEM

The challenges we faced were of three kinds. First, there is a potential perceptual problem for the users. The users employ physical bricks to design and manipulate a virtual setting which is simultaneously projected onto the table on which the bricks are sitting. Thus, the users are living and acting in two realities at once (Section 4.2.1). Designing bricks to link these realities fluently is related to the physical aspects of the system. Second, we encountered the technological problem of changing model height (Section 4.2.2) related to shifting focuses between the physical and the handling aspects of the system. The third problem we faced was a clear mixture of a perceptual problem for the user and a significant technological challenge to us: BUILD-IT uses 2D - or planar - interaction to access a 3D virtual setting (Section 4.2.3). The answer we found - a set of navigation tools - leverages alternative handling aspects of our system. The subject/object-directed aspects of the system (Section 4.1) will not be discussed any further.

4.2.1 Physical aspects: Brick design

The question related to the physical aspects was how to design an interface where users can move physical bricks and simultaneously immerse themselves in a virtual setting. Due to the use of reflective material for detection of brick position and orientation by the computer vision system, there is flexibility in the design of the brick size and shape. However, the bricks’ shapes need to help the user keep his or her fingers or hand off the reflective area. To test different designs, we performed some exploratory brick modeling (Lauche, 1998), based on different materials, forms, and metaphors (Fig. 8, left).
The first feasible brick was the block (Fig. 8, right, in center). The advantages of this shape are that users grasp the brick easily and that image detection is simple. However, due to the height of the brick, the continuity of the projected image is broken. Therefore the brick is not always suited to mediate fluently between users and the virtual world. A short experiment showed that this disruption led to a breakdown of the fluent mediation, presenting us with a problem.

To solve such a problem Kaptelinin (1996, p. 50) suggested: “deal with two interfaces instead of one user interface, with two borders, separating (1) the user from the computer and (2) the user and the computer from the outside world.” Kaptelinin draws upon Bateson’s (1972) “blind man’s stick dilemma”: where is the boundary between the individual who uses a tool and the external world? Does it coincide with the individual-tool boundary or with the tool-world boundary? In our case, the tool is the brick mediating between individual and virtual worlds. As an alternative to the original brick (Fig. 8, right, in center), we decided to design a brick so that users perceive interaction with the virtual world, not interaction with the tangible world of the bricks.

Such a brick can be designed by employing a reduction screen (Voorhorst, 1999). Such screens are borders placed in front of a monitor to reduce depth cues. The same idea can be used to construct the brick as an open box with a narrow black frame on the edge (Fig. 8, right). For users, the bottom of the box-like brick is indistinguishable from the table surface, thus minimizing image disruption. This design also reduced the risk of users accidentally covering the reflective area.

4.2.2 Shifting between physical and handling aspects: Changing model height

Up until this point, the system placed all models on a single storey. Task analysis (Section 4.1.4) showed that potential end-users wanted to be able to position models on different stories of a building. To do this we used a three-step design process (Section 3.4) suggested by Kaptelinin (1996). In a first step of that process, subjects work with a virtual tool moving single models into a virtual series of stories. For this step, user performance was not satisfactory and we looked for alternatives. In a second step we tested physical tools which affected a particular virtual floor. A virtual floor looks like a grid-layer. This grid-layer can be adjusted upward and downward. The models selected move along with the vertical placement of the virtual floor. Physical tools improved performance but their use was too tedious (Fjeld et al., 1999). In a third step, a second-
generation of virtual tools was developed, recombining elements of the first and the second step. Hence, the virtual floor could be handled with a virtual tool, the physical tools were no longer needed. The second step was related to the physical aspects of the system, the first and third steps were related to the handling aspects of the system. Each step is described below.

**Step 1: A virtual prototype for height manipulation: The height slice**

Height manipulation by means of side view handling was first achieved by copying a narrow vertical slice of the side view, called the *height slice*, and locating it along one edge of the plan view. The desired view of the height slice is selected and set by the user. Thus, models appear visible in the height slice, can be selected, and can be moved up and down (Fig. 9). When de-selected, models remain at the selected height. Further details are given in Fjeld et al. (1999b).

![Figure 9. Setting model height by selecting (left) and moving (center) a model in the height slice. The model being moved is also seen in the side view (right).](image)

**Step 2: Physical tools as mediators for model height**

Based on physical tools (Fig. 10), we explored three ways to physically handle model height. First, we implemented *Digit*, a digital controller which acts on a selected model and sets the model height by means of up-down buttons and provides a digital display of the height. Second, we developed *Tower*, offering the same buttons as the first tool combined with a luminous, quasi-analog scale showing selected height. The up-down buttons and the visual feedback are organized along the height axis, so action and perception are partly coincident. Third, we implemented *Slider*, a vertical sliding-rule where height is handled and indicated by an up-down handle. With *Slider*, handling and height cues are both organized along the height axis; thus action and perception are fully coincident (Section 3.2). Further details are given in Fjeld et al. (1999b).
Step 3: ‘Back to the virtual’: Floor handling

The resulting solution for selection and manipulation of the models selected among multiple stories is a virtual tool called Floor and is a second generation virtual solution. Floor reuses the height slice from the first solution and is based on the knowledge gained with the second solution. Floor is handled in the height slice, located along one edge of the plan view. The user first selects a number of models in the plan view, forming a group. This group of selected models then moves vertically as a whole when Floor is moved upward or downward in the height slice (Figs. 11-12). De-selected models remain at the last selected height. Only models at or above Floor are visible. Further details are given in Fjeld et al. (1999b).

4.2.3 Handling aspects: Spatial navigation

Our task analysis (Section 4.1.4) showed that experts navigate and inspect virtual environments in a range of activities, such as urban planning and architectural
walkthroughs. Depending on a user's acquaintance with virtual environments, navigation can range from exploratory epistemic to goal-directed pragmatic action. It is necessary to assume different points of view, to get an overview, and to look at things in detail in a fluent manner (Brooks, 1986).

Our answer to this need is offered by combining tangible bricks with 3D view handling of orientation and scale. Brick-based interaction in 2D has already been investigated (Fitzmaurice, 1996, Ullmer and Ishii, 1997). Also, bimanual camera manipulation and model handling in 3D graphics interfaces have been examined (Balakrishnan and Kurtenback, 1999) using two mice, a keyboard, and a screen. Here, we combine the strengths of these two approaches. The multimedia framework (MET++, Ackermann, 1996) we employ allows for full interaction in a 3D world. However, planar interaction with bricks provides only position and rotation but not height or inclination information. The resulting navigation methods are two alternative ways to bridge the gap between planar interaction and 3D view handling.

We first considered a design strategy based on the physical aspects, as was the case for general brick design (Section 4.2.1). This would have resulted in a physical, camera-like brick affecting the virtual side view camera. Such a solution would have required extending the properties sensed by the computer vision input. Based on our experience with the physical height tools (Section 4.2.2), we decided to explore virtual solutions (Fjeld et al., 1999a). The decision we took, is also supported by Ware and Rose (1999), who examined the use of real handles for the rotation of virtual models.

Hence, we investigated a design strategy based on the handling aspects of the system. Our strategy was to handle the virtual world or the virtual viewpoint (Section 2.1). This opened up two alternative design principles and both were investigated. First, in order to handle the virtual world - updating the view directly - we implemented the Continuous Update method. Second, in order to handle the virtual viewpoint - updating the view after viewpoint handling - we implemented the Select and Reframe method, where a frame represents the border of the plan view or the side view. The aim of our future research is to test which of these design principles is best suited for fluent navigation (8).

Each navigation method consists of two tools, one for the plan and one for the side view control (Table I). The tools can be activated in the virtual storage space (Fig. 13). By employing one brick - unimanual handling - shift and rotation of the controlled view can be set. By employing two bricks - bimanual handling - shift, rotation, and zoom of the controlled view can be set (9). Bimanual handling will be illustrated here. The two methods can be combined, but only one tool per view can be activated and used at a time.
Handling the virtual world: Continuous Update

Handling of the virtual world is implemented in the Continuous Update method and consists of two tools: GroundCatcher for the plan view and Camera for the side view. These tools work in a similar way. As soon as the GroundCatcher (Fig. 14) is selected from the menu and placed, it locks to the setting. All subsequent handling affects the plan view. To quit the tool, the bricks are covered and removed. By selecting the Camera (Fig. 15) from the menu, the part of the setting shown in the side view can be set. A zoom handle is selected with a second (here: right hand) brick. By moving the zoom handle and the camera further apart, the side view can be enlarged; by moving them closer, the side view can be focused. Covering and removing the second brick freezes the zoom. The Camera remains visible and accessible in the view. Further details are given in Fjeld et al. (2000).
Handling the virtual viewpoint: Select and Reframe

Handling of the virtual viewpoint is implemented in the Select and Reframe method and consists of two tools: FrameCatcher for the plan view and ViewFrame for the side view. Both tools employ a rectangular frame representing the border of the view.

When the FrameCatcher (Fig. 16) is selected from the menu, the setting automatically is zoomed out to show a wider context. As soon as the FrameCatcher is placed within the frame, it locks to the frame. All subsequent handling affects the frame and the desired part of the setting can be selected. Covering and removing the brick triggers a reframe of the view, responding to user selection. When the ViewFrame (Fig. 17) is selected from the menu, the side view is automatically zoomed out to show a wider context. The desired part of the setting can be selected. A zoom handle is selected with the second brick, thus resizing the window. Covering and removing the second (here: right hand) brick freezes the zoom. Covering and removing the first brick triggers a reframe of the side view, responding to user selection. The ViewFrame remains visible and accessible in the view. Further details are given in Fjeld et al. (2000).
5. Conclusion

The aim of this paper is to explain and illustrate our design philosophy for developing graspable groupware; we have focused on describing our experience with tool design for planning and configuration tasks. The aim of our work was to enable the exteriorization of end-users’ planning processes. We achieved this goal by designing a graspable interface. With this interface several users can communicate and interact in a coincident action perception space and thereby reach common objectification. Our planning tool not only supports goal-directed but also exploratory action. We took an AR approach where virtual models are manipulated through physical bricks. In this paper we present examples from our design practice for the physical and the handling aspects (Bødker, 1996) of our system. Related to physical aspects, we show how we designed brick. Related to shifting focuses between the physical and the handling aspects, we show how we developed solutions for height adjustment. Related to the handling aspects, we show how we developed a set of tools for spatial navigation.

From the experience we gained in using activity theory for the design of graspable groupware we distilled a set of ten design guidelines. These guidelines are presented in Table II and each guideline is discussed. Lastly, we reflect on our design practice and point out shortcomings and challenges for future work.
<table>
<thead>
<tr>
<th>Table II. Set of design guidelines</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Use physical interaction handles as exteriorizations</td>
</tr>
<tr>
<td>2. Assure coinciding action and perception spaces (Rauterberg, 1995)</td>
</tr>
<tr>
<td>3. Support body motions and simple everyday skills (Campbell, 1988)</td>
</tr>
<tr>
<td>4. Respect body-space (Rauterberg, 1999), using less intrusive devices (Vince et al., 1999)</td>
</tr>
<tr>
<td>5. Support a clear binding between physical handles and virtual models</td>
</tr>
<tr>
<td>6. Draw on bimanual coordination skills</td>
</tr>
<tr>
<td>7. Foster exploratory epistemic action by assuring low risk in trying out</td>
</tr>
<tr>
<td>8. Support fluent navigation methods to explore 3D virtual worlds</td>
</tr>
<tr>
<td>9. Support tactile or haptic feedback (Mackenzie, 1994)</td>
</tr>
<tr>
<td>10. Give visual feedback consistent with user expectations</td>
</tr>
</tbody>
</table>

1. The physical handles we use are simple, non-specialized bricks. This means that the bricks do not represent particular objects like chairs, sofas, etc. For future research we want to study the use of specialized bricks. The interest in such solutions has been shown by recent experiments of handling virtual models with real handles (Ware and Rose, 1999).

2. First, the principle of coinciding action and perception spaces is fully implemented for the plan view; it is not implemented for the side view. Second, one of the physical tools developed for height manipulation (Slider) offers a high degree of coincidence between action space and perception space but its use is too tedious (Fjeld et al., 1999b).

3. A single brick is handled by various operations, like selecting, placing, rotating, repositioning, and covering.

4. Our system does not rely on any intrusive devices. The only quasi-intrusive aspect of the system is that users reaching across the table interfere with the light beams and thereby cast shadows.

5. The binding between bricks and virtual models was kept uniform. However, alternative kinds of binding - called locking mechanisms (4) - have been implemented and will be investigated in future research.

6. By offering bimanual interaction, coordination skills from everyday handling are employed, for instance aligning and grouping. Bimanual interaction has been studied extensively by Guiard (1987) and Fitzmaurice and Buxton (1997) and guidelines may be drawn from their results.

7. Since the system is based on natural, everyday modes of communication and interaction (Section 3.1), the end-users were observed to carry out exploratory epistemic action without inhibition. Users are inherently used to a natural style of interaction and thereby perceive low levels of risk in trying out their ideas with the system. To lower the level of that risk even further, we still plan to implement an ‘undo’ functionality. Based on a continuous protocol, users will be able to go back one or several steps in their planning process.

8. The alternative navigation methods (Section 4.3.3) enable users to explore the 3D virtual world. It remains to discover which design principle offers higher performance. The fluent operation of navigation methods partly relies on implementation and partly on computer graphics performance. The current
combination of these factors (Fjeld et al., 2000) already allows for satisfactory
navigation and is planned to be tested for its usability.
9. Tactile feedback is assured by tangible bricks. Placing a higher priority on physical
handles (Ware and Rose, 1999) is a way to offer more tactile feedback. For the future
Fitzmaurice’s (1996) concept of forced feedback by propelled bricks could be of
interest to us.
10. Visual feedback relies heavily on computer graphics performance and at present
results in a delay in updating the image. With rapidly increasing performance
standards of graphic cards and processors, we expect our system to give users an
even more authentic feeling in the near future.

Our experience in applying activity theory has been generally positive. The theory
has brought structure to our thoughts and to our design practice; the vocabulary of
activity theory has proved to be useful for our discussions. However, activity theory still
remains difficult to access for practitioners in the fields of design and computer science.
In fact, the BUILD-IT project was also nourished by thoughts stemming from ‘The
ecological approach to visual perception’ by Gibson (1986). It is currently beyond the
scope of our research to introduce Gibson and to systematically compare the tradition of
activity theory with Gibson’s approach.

In terms of the limitations of activity theory put forward by Kaptelinin (1996) and
discussed in Section 2.1, our work may also stimulate theoretical development. We
believe that even a virtual model of an object may serve as objectification of mental
activity if it can be handled (Bødker, 1991). Which kind of virtuality is still perceived as
exteriorization rather than disconnected outer world - or “just one more environment”
(Kaptelinin, 1996, p. 64) - is an empirical question. We suspect that the degree of
exteriorization depends on users’ acquaintance with virtual tools. For the less
acquainted, only the paper printout counts as an objectification, whereas for the more
acquainted the virtual version - the file - is perceived as an objectification of their inner
activity. Whatever the empirical outcome might be, we also look forward to more
theoretical discussions on what constitutes objectification.

Notes
1. BUILD-IT can also be used by a single person, but that usage is not the focus of this
paper or the work of our group - where social aspects are being discussed and social
uses developed.
2. The worker on the left straightens out a copper sheet on a marble table and controls
its quality. The worker on the right (Fig. 1, left) heats a steel bar at the smithy and
opens a fresh-air supply. Among their tools is a beam compass, a right angle jacket, a
copper fitting mold, and a copper thread dispenser (Fig. 1, right, top to bottom).
3. Compared with physical, model-based layout systems, BUILD-IT offers cheaper,
quicker, and more exact model representation in a virtual environment. Based on a
3D multimedia framework (MET++, Ackermann, 1996), the system can read and
display geometrical forms.
4. At the moment of selecting a virtual model with a physical brick a planar relation - in
terms of position and rotation - is established between the physical and the virtual
worlds. We call this a locking mechanism. A locking mechanism determines how a
physical brick and a virtual model stay connected from the moment of selection until
the moment of de-selection. For circular and rectangular brick forms (Section 4.2.1)
appropriate locking mechanisms must be defined. A particular kind of locking mechanism is based on one particular kind of alignment procedure. In the future we will report on our usability tests of various combinations of brick forms and locking mechanisms.

5. Employing Virtual Reality Modeling Language (VRML), individual models are transferred from a CAD system to BUILD-IT (Fjeld et al., 1998b). After a planning session, the positioned models can be returned to the CAD system.

6. Drisis (1996) predicted that new forms of interaction, for instance graspable user interfaces, could challenge the programming paradigms of CAD design.

7. Some of the tools described in this paper were related to, or based on, metaphors, such as tower, sliding-rule, floor, and frame. It may be of interest to know more about the consequences of using metaphors (Alty, 1998), not only for end-user activity, but also for the process of constructing our system.

8. Usability testing of the alternative navigation methods should call for handling of the plan view and of the side view. Due to direct feedback and a lower number of operations, we expect Continuous Update to deliver higher performance than Select and Reframe.

9. We notice that bimanual interaction underpins exploratory epistemic latitude, by enabling zoom. The manner in which zoom is handled, through two-handed operations, raises the topic of asymmetry (Guiard, 1987). Combinations of the factors (shift, rotation, and zoom) and their relation to one-handed and two-handed interaction will be explored in future research. Help may be found in the concept of time-multiplexed and space-multiplexed input schemes (Fitzmaurice and Buxton, 1997).

Acknowledgments

We thank Barbara Stuckey for her editorial assistance, which substantially improved this paper; Nicholas Ironmonger and Urs Gasser for their editorial assistance, the reviewers and the editors for their constructive feedback; Peter Troxler (peter@klapt.ch) for the photos in Figs. 5, 6, 7 (left), and 9. Morten Fjeld thanks the Research Council of Norway (www.forskningsradet.no) for his Ph.D. fellowship.

BUILD-IT is a research project funded by

- The Swiss Federal Commission for Technology and Innovation (KTI),
- The Center for Integrated Production Systems (ZIP) at ETH Zurich, and
- The Institute of Mechanical Systems (IMES) at ETH Zurich.

BUILD-IT is a registered trademark of TellWare (info@tellware.com).
5 BUILD-IT: a planning tool for construction and design

Video publication - see separate CD

ABSTRACT
It is time to go beyond the established approaches in human-computer interaction. With the Augmented Reality (AR) design strategy humans are able to behave as much as possible in a natural way: behaviour of humans in the real world with other humans and/or real world objects. Following the fundamental constraints of natural way of interacting we derive a set of recommendations for the next generation of user interfaces: the Natural User Interface (NUI). The concept of NUI is presented in form of a runnable demonstrator: a computer vision-based interaction technique for a planning tool for construction and design tasks.

---

6 Alternative Tools for Tangible Interaction: a Usability Evaluation

Abstract
In this paper, we evaluate a tangible user interface (TUI), called the BUILD-IT system, as a research platform for tangible user interaction. The BUILD-IT system is based on computer vision technology and is the interface for a multi-user planning tool called. Following the tradition of Augmented Reality (AR), projected light replaces the use of screens as the output medium. Grouped around a table and employing tangible physical bricks, users can select and manipulate virtual models within the scene which they are planning. The system enables its users to cooperate in a virtual environment for planning a real-world project, such as a room, a school, or a factory. In this evaluation, we compared four alternative tools for single-user problem solving, BUILD-IT being one of them. We wanted to make plausible that BUILD-IT, a tangible user interface, is a productive way to solve spatial problem without having the build physical models of the three-dimesional task domain. In a pilot study, followed by an experiment, all participants solved the same positioning problem, using one of these tools. We measure performance, learning effect, and user satisfaction. The tools had various degrees of three-dimensionality and required various degrees of cognitive effort and previously acquired skills. Between some of the tools, we found significant differences in terms of performance, ease of use, and user satisfaction. Since the difference between BUILD-IT and a conventional, two-dimensional cardboard tool was partly significant, and since there was no significant difference between BUILD-IT and the physical model tested, we conclude that BUILD-IT is a usable research platform to investigate tangible user interfaces.

Introduction
The BUILD-IT system (Fjeld et al., in press) was designed to support multi- and single-user spatial planning activities and may be used to solve problems requiring an analytical, a creative, or an analytical-creative work-style (Fig. 1). In order to establish relevant references in evaluating BUILD-IT, we looked for general principles behind the practice of spatial planning. General principles must be valid across different forms of system and apply to single and to multiple users. As a first step to investigate the system’s validity, we evaluate BUILD-IT against alternative tools as a means to solve single-user analytical planning problems. The specific task we employed was a spatial positioning problem.

---

The aids offered by a tool has been conceived of by Gibson (1986) in terms of affordances. For example, we hold a pencil in such a way that it fits the hand, ignoring less appropriate ways that it might be grasped. "The pencil affords being held in this way as a result of its length, width, weight, and texture, all with respect to the size, configuration, and musculature of our hand" (St. Amant, 1999). Most of these properties and relationships are visible. The possible interaction with an object or an environmental feature can be determined simply by looking at it. Since this evaluation consists of an analytical problem, we may view the affordances of a tool as cognitive support (Bødker, 1991). One productive way to understand the facets of cognitive support is found in the decision support techniques (Zachary, 1986, p. 129). Zachary suggested six decision support techniques, however we view four as relevant to guide our research:

1. "Representation aids, assisting in expression and manipulation of a specific representation of a decision problem."
2. "Information control techniques, helping in storage, retrieval, organisation, and integration of data and knowledge."
4. "Choice models, supporting integration of decision criteria across aspects and/or alternatives."

(1) guides us to focus on problem representation, which we realise through the design and comparison of alternative planning tools. (2) guides us to study learning (here: learning effect), as a process observable through the actions of the participants. (3) guides us to design tools for different strategies and levels of expertise. (4) guides us to design tools facilitating rational decision making. Motivated by the concept of cognitive support, we focus on the design of tools that support these four aspects of decision making.

The remaining two techniques suggested by Zachary (1986, p. 129), "process models, assisting in projecting the future course of complex processes, and judgement amplification/refinement techniques, helping in quantification and debiasing of heuristic judgements" are less relevant since we employ a positioning problem in a static environment operating at one scale at a time only.
support (Bødker, 1991), and guided by the four decision support techniques (Zachary, 1986), we carried out an comparative evaluation of BUILD-IT against a set of alternative planning tools. The evaluation consisted of a pilot study comparing four alternative tools, followed by an experiment comparing a selected sub-set of three alternative tools.

Problem Representations through Alternative Tools

Although numerous prototypes for tangible interaction have been proposed (Ullmer, 2000), less effort has been offered to usability aspects of such systems and to how they may influence the nature of collaborative work. Motivated more by the former and partly by the latter of these deficits, the purpose of this evaluation was to assess the quality of BUILD-IT as a planning tool and thereby to qualify it as a research platform for our investigation into tangible interaction. The BUILD-IT system is typically used for collaborative planning tasks in interior architecture, city planning, or production plant layout. Investigating different levels of abstraction, we found that i) physically down-scaled three-dimensional (3D) replicas and ii) two-dimensional (2D) cardboard models are two characteristic representations. A more abstract representation is iii) a combination of tools like ruler, compass, and calculator. These three representations led us to construct three alternative planning tools. Our tools were synthetically constructed as different means to solve a spatial laser positioning problem and had the sole purpose to evaluate BUILD-IT in a wider context of planning tools.

In a first approach, the cognitive support of a tool is defined by short task solving time (performance), ease of use, and high user satisfaction.

Problem definition

Since we wanted to compare different tools for planning and layout activities, we needed to define a typical problem at an abstract level. In a production plant layout, a typical task is to bring 3D models (physical or virtual), e.g. welders and assembly lines, into a certain spatial relation. In practice, such tasks are carried out under certain spatial and time constraints. We decided to define the task in terms of spatial constraints and to measure time of trial, number of steps required, and user satisfaction as the dependent variables. Hence, we constructed a synthetic laser positioning task, with well-defined spatial relationships and constraints. A schematic description of the task scene and the task, based on Fig. 2, follows:

- The task is to find the block where the laser source \( L \) hits the target \( T \). The light beam should be as close as possible to the centre of the target and there is only one solution.
- The scene is shown from the side (called planimetric side view containing \( T, B, \) and \( L \)) and from above (called planimetric plan view containing \( T \) and \( B \)).
- Nine square blocks \( B \) with different heights form a three-by-three matrix.
- The target \( T \) with height 16 mm is situated at a distance 63.4 mm from the matrix.
- The laser source \( L \) with height 24 mm and slope 0.175 can be placed on any of the nine blocks.
We first describe BUILD-IT, then the other tools in the order of their expected cognitive support (high to low): 

**Tool Description**

**Tool 1 - BUILD-IT** (medium cognitive support expected): This tool employs a virtual modelling of the blocks, target, and laser source. No navigation of the views is permitted - they remain fixed as shown in Fig. 3. The participants use one brick with which they manipulate the virtual laser source.

**Tool-dependent instructions:** The participants were shown how to select and handle the virtual model representing the laser source. It was shown how the plan view and the side view provide complementary, task-relevant information.

Figure 3: Using the BUILD-IT tool, plan view is left and side view is right: a) start-up; b) testing a first block (above the target); c) testing a second block (below the target); d) testing a third block (hits the target).
**Tool 2 - PhysicalBlocks** (high cognitive support expected): This is a physical task realisation (Fig. 4), four times scaled up in relation to the measurements in Fig. 2 (in mm) given in Fig. 2. It consists of nine metal blocks mounted on a cassette, a standard laser source (maximum diameter: 15 mm, length: 50 mm), and a target made of a metal pin with a 5 mm by 15 mm wide metal flag. The height of the metal flag, target position, and blocks positions can be adjusted.

**Tool-dependent instructions:** The participants were shown how to position and rotate the laser source between and within different blocks. They are not allowed to manipulate the cassette, the blocks, or the target flag, which are all fixed to the table. The laser beam hits a wall, indicating the direction of the beam also when it does not hit the target.

Figure 4: Solving a task with PhysicalBlocks. Some typical steps are shown: a) start-up; b) testing a first block (above the target); c) testing a second block (below the target); c) testing a third block (hits the target). (The spot of the laser beam has been redrawn for print reasons).
Tool 3 - Cardboard (medium cognitive support expected): Solving a task with the Cardboard may typically follow these steps (Fig. 5):

a) start-up showing (from bottom to top) floor ruler, combined planimetric plan and side views, laser source, beam ruler, task definition with blocks,

b) selecting a first block,

c) fixing distance of the block with the floor ruler,

d) rotating that distance into the planimetric side view,

e) putting the block at rotated distance,

f) putting the laser source at the block,

g) putting the beam ruler at the laser; testing where beam goes (above the target),

h) testing a second block (below the target),

i) testing the closer edge of the same block (still below the target),

j) testing the farther edge of the same block (still below the target), and

k) testing a third block (hits the target)
This tool is a planimetric tool two times scaled up in relation to the measurements of Fig. 2 (in mm). Most elements in this tool are produced in cardboard and are (Fig. 4a): a cardboard ruler emulating the floor; a combined planimetric plan and side view showing the nine block positions in a plan view aspect and the floor with the target in a side view perspective; a cardboard laser source; a metal ruler emulating the laser beam; the nine cardboard blocks configured by the task definition. Next, we show how a block is tested (Fig. 4).

**Tool-dependent instructions:** The principles of planimetric work were explained, i.e. working in a x-y projection (plan view) and a y-z projection (side view) of a 3D scene. It was explained that these two projections were integrated into one tool. Then, an example given shown how one single block is tested, and in particular how a distance is fixed in the planimetric plan view and rotated into the planimetric side view. The blocks are already at their position, giving the task definition. However, in case a participant looses the position of a block, the link is given by the number on the reverse side of the block and the corresponding height on the position. As seen in Fig. 4, blocks already tested are laid back with a slightly different position, indicating that they were already tested. Height figures are not used to solve the problem as such.

**Tool 4 - Mathematics** (expected low cognitive support): This tool consists of the problem definition in Fig. 2, a paper, a pen, a ruler, a stencil, and a calculator (Fig. 6). The participants were familiar with the task from Grammar school mathematical skills. With the given tools, they could therefore choose to employ geometry, vector or linear function calculus to solve the task.

**Tool-dependent instructions:** A planimetric representation with a x-y (plan view) projection and a y-z (side view) projection of the task domain was used (Fig. 2). The target is marked k” in the side view and k’ in the plan view. There is a laser source marked l with inclination of alpha = 10° (slope m = 0.175). The height of the laser is 24 units.

Figure 6: Mathematical tool: planimetric representation (as in Fig. 2), a pen, a compass, a calculator, a ruler, and a blank sheet.
Relations among tools

We assumed that the three alternative tools, having different commonalties with BUILD-IT, would give different cognitive support (Table I).

<table>
<thead>
<tr>
<th>Tool</th>
<th>Problem representational</th>
<th>Commonality with BUILD-IT</th>
<th>Expected Cognitive Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physical Blocks</td>
<td>Physical, one-to-one</td>
<td>Three-dimensional</td>
<td>High</td>
</tr>
<tr>
<td>BUILD-IT</td>
<td>Augmented Reality (AR)</td>
<td>-</td>
<td>Medium</td>
</tr>
<tr>
<td>Cardboard</td>
<td>Planimetric</td>
<td>Planimetric; two-dimensional</td>
<td>Medium</td>
</tr>
<tr>
<td>Mathematics</td>
<td>Written</td>
<td>Computational resources</td>
<td>Low</td>
</tr>
</tbody>
</table>

Table I: The four tools, their representational characteristics, and expected cognitive support.

Questionnaire

To test user-satisfaction, a questionnaire was established. The first five questions relate to pre-experimental knowledge, the next three questions relate to users' satisfaction with the task-tool combination (questioning clarity of task presentation, difficulty - or rather ease - of task, and suitability of tool. The last question is an open-ended one. The participants were asked to answer the questions by ticking one (unless otherwise stated) box for each questions.

- How often do you use a computer? (The exclusive choices were: less than 1 hour monthly; at least 1 hour monthly; at least 1 hour weekly; at least 1 hour daily).
- For what purpose(s) do you use a computer? (The multiple choices were: games; email and web; planning and construction; text editing and coding; other purposes).
- How much did you know about BUILD-IT before this experiment? (one selection allowed: do not know; have heard or read; have seen how it works; have used).
- Did you previously solve similar tasks? (The exclusive choices were: never; a few times; quite often; very often).
- How was the clarity of the task formulation? (The exclusive choices were: not at all; not; neutral; good; very good).
- How did you perceive the difficulty of the task? (The exclusive choices were: very difficult, difficult; neutral; easy; very easy).
- How was the suitability of the tool you used to solve the task? (The exclusive choices were: of no help at all; of no help; well suited; of help; of high help).
- Please give any other comments, additional remarks, and critique (This was an open-ended question and answers were given in free form).

Pilot Study

The whole experiment should last a maximum one hour for each participant. In the pilot study, we did not look for statistical evidence, but simply for approximate values to guide the set-up of the experiment. The pilot study examined all four tools and aimed at an optimisation of the experimental set-up with respect to:

- Design
- Procedure
• Time per trial (max. 10 minutes)
• Difficulty of the task-tool alternatives
• User satisfaction with the given tool alternatives
• Relevance of tool comparison

Design
To avoid between tool learning, we used a 4 single between group design. Each tool was randomly assigned to two participants, requiring a total of eight participants. For each trial we measured the time of trial.

Participants
The participants were eight undergraduate or graduate students. They all had a Grammar school or equivalent degree, this assured a certain level of skills in mathematics and geometry. There was one female and there were seven male participants aged 25 to 30 years. No importance was given to whether participants were left- or right-handed. The participants were not paid.

Task and Apparatus
All participants solved the same task. Each tool was assigned to two participants.

Procedure
The user study was performed with single participants during daytime where each participant solved one task only. It was carried out in an office with the investigator sitting next to the participant. The pilot study consisted of the following steps:

a) Description of tool, task, and task completion (approx. 10 min.)
b) Task-dependent instructions as given for each tool with a demo trial (1 - 8 min.)
c) One unaided trial with registration of time of trial.
d) The participant filled out the questionnaire.

Results
The data collected with the experimental set-up show a clear relation between tool (independent variable) and time of trial (dependent variable) (Table II). The first three tools appear to give sufficiently small and comparable values for time completion times. The fourth tool, Mathematics, showed to be too different to represent a relevant comparison with the other tools. The questionnaire was slightly optimised.

<table>
<thead>
<tr>
<th>Tool</th>
<th>Mean Time of Trial [minutes]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physical Blocks</td>
<td>0.3</td>
</tr>
<tr>
<td>BUILD-IT</td>
<td>2.0</td>
</tr>
<tr>
<td>Cardboard</td>
<td>3.5</td>
</tr>
<tr>
<td>Mathematics</td>
<td>26.0</td>
</tr>
</tbody>
</table>

Table II: Results of the user study: time of trial is the average for the two participants per tool and seconds are given as decimals of minutes.
**Experiment**

The experiment examined three tools and was carried out with the aim of learning more about performance (time used to solve one or a set of tasks) and use of these tools, with a focus on BUILD-IT.

**Hypotheses**

We conjectured that Physical Blocks give the highest cognitive support, since it allows participants to directly grasp the problem and solve it without too much reflection (Fig. 7). We conjectured that BUILD-IT gives the second best cognitive support, since it offers a representation close to the physical one by using virtual 3D models. Then, we conjectured Cardboard to offer the third best cognitive support, since it applies 2D representation and requires more abstract thinking from the participants. With BUILD-IT as the focal point for our comparative study, the hypotheses were stated in the conventional form, as appropriate for statistical testing (Table III).

![Figure 7: Expected cognitive support of each tool and how the inner relations between each tool is tested by a set of hypotheses, H1-H3, given by Table III.](image)

<table>
<thead>
<tr>
<th>Hypothesis</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>H1: Cardboard</td>
<td>gives less cognitive support than Physical Blocks.</td>
</tr>
<tr>
<td>H2: BUILD-IT</td>
<td>gives more cognitive support than Cardboard.</td>
</tr>
<tr>
<td>H3: BUILD-IT</td>
<td>gives less cognitive support than Physical Blocks.</td>
</tr>
</tbody>
</table>

Table III: Hypotheses.

**Design**

To avoid between tool learning, we used a 3 single between group design. Each tool was assigned to ten participants, thereby excluding potential learning from one tool to another. There were 12 task variations, one for demonstration trial (always the same), one for an aided trial (always the same). There were 10 task variations for unaided experimental trials. This was meant to be sufficient for all participants to solve five tasks correctly without solving any task more than once. These ten variations were permuted by latin squares so that trial-variation combinations were equally distributed.

**Operationalization of Cognitive Support**

Some learning will normally take place during the demonstration trial and the aided trial, additional learning may take place throughout the unaided trials. We chose only to register the learning taking place in the unaided trials.

To examine that conjecture in the form of hypotheses, we broke cognitive support into a set of five measurable variables. These were: i) time of trial, ii) number of blocks tested...
in a trial, iii) difference in time of trial between last and first correctly solved task, iv) difference number of blocks tested between last and first correctly solved task, v) personal satisfaction with the given tool.

For each tool and participant, cognitive support was measured by five criteria:

**C1:** Time of trial is low.

**C2:** Few blocks are tested per trial.

**C3:** Time of trial is lower in last than first trial solved (i.e. learning effect seconds)

**C4:** Fewer blocks are tested in the last rather than in the first correctly solved trial (i.e. learning effect blocks tested).

**C5:** Personal satisfaction with task-tool combination given, given by a weighted mean of task difficulty, clarity of task description, and tool suitability, is high.

**Participants**

The participants were thirty undergraduate or graduate students. There were 13 female and 17 male participant aged between 20 and 36 years. No importance was given to whether participants were right- or left-handed. The participants were paid CHF 10 each.

**Task and Apparatus**

We used the same tasks and apparatus (except Mathematics) as in the pilot study.

**Procedure**

The user study was performed with single participants during daytime and each participant solved one aided and five unaided task varieties. It was carried out in an office with the investigator sitting next to the participant. The experiment consisted of the same steps as in the pilot study:

- a) Description of tool, task, and task completion
- b) One demo trial (three false block, then the correct block, were tested)
- c) One aided trial (questions were answered and help was offered at need)
- d) Each participant had to perform five tasks with correct answer, the three last correct ones in a closed sequence. After having solved a task, the participants rang a bell. For all trials we registered the number of blocks tested, time of trial, and whether the indicated block was the correct one. The next task was initialised without the participant watching.
- e) The participant filled out the questionnaire.

**Logging**

For each trial, the number of tested blocks was registered, being at least one. Blocks tested more than once are counted each time. Time of trial was registered when the participant rang a bell.

**Experimental Results and Discussion**

In order to test the criteria C1 – C4, the analysis was carried out to investigate differences between BUILD-IT, PhysicalBlocks and Cardboard. Consequently, much
like Balakrishnan et al. (1999), we used a multiway ANOVA, here with a General Linear Model (GLM). The five independent variables were tool (PhysicalBlocks, BUILD-IT, Cardboard), participant (1-30), trial (T1-T5), and task. Four dependent variables each tested C1 – C4, and were seconds (C1) and number of blocks (C2), and improvement in seconds from first to last correct trial (C3), improvement in number of blocks from first to last correct trial (C4). Hence, performing four tests, the Bonferroni method was used with $k = 4$, alpha = 0.05/4 = 0.012. Five further dependent variables were measured for descriptive reasons only, and were: seconds total, seconds per trial, blocks total, blocks per trial, and trials needed. Significant effects are shown by p-values less than alpha are marked by a star (*).

We tested C1 and C2 (Table IV) through a first analysis by examining tool, trial, task as independent variables, and participant as a variable hierarchically nested with tool. We found it useful to examine tool task interaction. In a second analysis (Table IV), the dependent variables not having a significant effect were excluded. These results were then used in the post-hoc analyses (Table V).

<table>
<thead>
<tr>
<th>Dependent variable</th>
<th>Independent variable</th>
<th>df</th>
<th>F-ratio</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>seconds (1st analysis)</td>
<td>tool (testing C1)</td>
<td>2</td>
<td>12.652</td>
<td>&lt; 0.001*</td>
</tr>
<tr>
<td></td>
<td>task</td>
<td>9</td>
<td>4.156</td>
<td>&lt; 0.001*</td>
</tr>
<tr>
<td></td>
<td>trial</td>
<td>7</td>
<td>2.121</td>
<td>0.048</td>
</tr>
<tr>
<td></td>
<td>participant (tool)</td>
<td>27</td>
<td>3.387</td>
<td>&lt; 0.001*</td>
</tr>
<tr>
<td></td>
<td>tool * task</td>
<td>18</td>
<td>1.399</td>
<td>0.149</td>
</tr>
<tr>
<td>seconds (2nd analysis)</td>
<td>tool (testing C2)</td>
<td>2</td>
<td>10.199</td>
<td>&lt; 0.001*</td>
</tr>
<tr>
<td></td>
<td>task</td>
<td>9</td>
<td>4.373</td>
<td>&lt; 0.001*</td>
</tr>
<tr>
<td></td>
<td>participant (tool)</td>
<td>27</td>
<td>3.108</td>
<td>&lt; 0.001*</td>
</tr>
<tr>
<td>number of blocks (1st anal.)</td>
<td>tool</td>
<td>2</td>
<td>23.435</td>
<td>&lt; 0.001*</td>
</tr>
<tr>
<td></td>
<td>task</td>
<td>9</td>
<td>6.344</td>
<td>&lt; 0.001*</td>
</tr>
<tr>
<td></td>
<td>trial</td>
<td>7</td>
<td>1.689</td>
<td>0.120</td>
</tr>
<tr>
<td></td>
<td>participant (tool)</td>
<td>27</td>
<td>1.632</td>
<td>0.042</td>
</tr>
<tr>
<td></td>
<td>tool * task</td>
<td>18</td>
<td>2.410</td>
<td>0.003*</td>
</tr>
<tr>
<td>number of blocks (2nd anal.)</td>
<td>tool</td>
<td>2</td>
<td>19.979</td>
<td>&lt; 0.001*</td>
</tr>
<tr>
<td></td>
<td>task</td>
<td>9</td>
<td>6.699</td>
<td>&lt; 0.001*</td>
</tr>
<tr>
<td></td>
<td>tool * task</td>
<td>18</td>
<td>2.455</td>
<td>0.002*</td>
</tr>
</tbody>
</table>

Table IV: Per trial data: GLM results, where significant effects are marked with a star (*).
<table>
<thead>
<tr>
<th>Dependent and independent variables</th>
<th>Pair(s)</th>
<th>Pairwise difference</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>seconds; tool</td>
<td>BUILD-IT &amp; Cardboard</td>
<td>-15.5</td>
<td>0.030</td>
</tr>
<tr>
<td>seconds; tool</td>
<td>Cardboard &amp; PhysicalBlocks</td>
<td>-27.4</td>
<td>&lt; 0.001*</td>
</tr>
<tr>
<td>seconds; task</td>
<td>1 &amp; 3, 5, 7-10</td>
<td>&lt;-40.855, -53.885&gt;</td>
<td>&lt; 0.021&gt;</td>
</tr>
<tr>
<td>number of blocks; tool</td>
<td>BUILD-IT &amp; Cardboard</td>
<td>-4.493</td>
<td>&lt; 0.001*</td>
</tr>
<tr>
<td>number of blocks; tool</td>
<td>Cardboard &amp; PhysicalBlocks</td>
<td>2.811</td>
<td>&lt; 0.001*</td>
</tr>
<tr>
<td>number of blocks; task</td>
<td>1 &amp; 3-10</td>
<td>&lt;-5.967, -8.652&gt;</td>
<td>&lt; 0.002&gt;</td>
</tr>
</tbody>
</table>

Table V: *Per trial data:* post-hoc pairwise comparison.

<table>
<thead>
<tr>
<th>Dependent variable</th>
<th>Independent variable</th>
<th>df</th>
<th>F-ratio</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>learning effect seconds</td>
<td>tool (testing C3)</td>
<td>2</td>
<td>1.564</td>
<td>0.228</td>
</tr>
<tr>
<td>learning effect blocks t.</td>
<td>tool (testing C4)</td>
<td>2</td>
<td>0.529</td>
<td>0.595</td>
</tr>
<tr>
<td>seconds per trial</td>
<td>tool</td>
<td>2</td>
<td>3.809</td>
<td>0.035</td>
</tr>
<tr>
<td>seconds total</td>
<td>tool</td>
<td>2</td>
<td>3.328</td>
<td>0.051</td>
</tr>
<tr>
<td>number of blocks per trial</td>
<td>tool</td>
<td>28</td>
<td>8.887</td>
<td>&lt; 0.001*</td>
</tr>
<tr>
<td>number of blocks total</td>
<td>tool</td>
<td>28</td>
<td>8.769</td>
<td>&lt; 0.001*</td>
</tr>
<tr>
<td>trials needed</td>
<td>tool</td>
<td>2</td>
<td>1.679</td>
<td>0.205</td>
</tr>
</tbody>
</table>

Table VI: *Overall data:* GLM results, where significant effects are marked with a star (*).

<table>
<thead>
<tr>
<th>Dependent and independent variables</th>
<th>Pair</th>
<th>Pairwise difference</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>number of blocks per trial</td>
<td>BUILD-IT &amp; Cardboard</td>
<td>-4.291</td>
<td>&lt; 0.001*</td>
</tr>
<tr>
<td>number of blocks per trial</td>
<td>Cardboard &amp; PhysicalBlocks</td>
<td>2.789</td>
<td>0.035</td>
</tr>
<tr>
<td>number of blocks total</td>
<td>BUILD-IT &amp; Cardboard</td>
<td>-24.900</td>
<td>&lt; 0.001*</td>
</tr>
</tbody>
</table>

Table VII: *Overall data:* post-hoc pairwise comparison.
To know more about learning during the phase of application of a specific tool, we plotted time required (Fig. 9, top row) and blocks tested (Fig. 9, bottom row). In using BUILD-IT, there seems to have been a learning effect in terms of time and number of blocks tested (Fig. 9, left column). In using Cardboard, there seems to have been some learning in terms of blocks tested but less in terms of time used (Fig. 9, middle column). In using PhysicalBlocks, there seems to have been a learning effect in terms of time used but less in terms of blocks tested (Fig. 9, right column). We interpret that with BUILD-IT, performance improved because participants learned how to use the tool. With PhysicalBlocks, performance improved without any observable change in use, indicating that users simply became more confident. With Cardboard, performance did not improve in spite of an observable change in use.)
Subjective Preference

Tables VIII–X gives participants’ ratings; showing for each tool the number of participants selecting each rating. We shifted the classifications to a balanced scale [-2,-1,1,2], going from low to high satisfaction, and the mean rating is given.

<table>
<thead>
<tr>
<th>Tool</th>
<th>-2</th>
<th>-1</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>Mean rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>PhysicalBlocks</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>7</td>
<td>1.7</td>
</tr>
<tr>
<td>BUILD-IT</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>5</td>
<td>1.4</td>
</tr>
<tr>
<td>Cardboard</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>3</td>
<td>1.3</td>
</tr>
</tbody>
</table>

Table VIII: Perceived clarity of the task explanation (very unclear: -2; very clear: 2).

<table>
<thead>
<tr>
<th>Tool</th>
<th>-2</th>
<th>-1</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>Mean rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>PhysicalBlocks</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>1.2</td>
</tr>
<tr>
<td>BUILD-IT</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>5</td>
<td>0</td>
<td>0.5</td>
</tr>
<tr>
<td>Cardboard</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>3</td>
<td>0</td>
<td>0.3</td>
</tr>
</tbody>
</table>

Table IX: Perceived difficulty of the task (very difficult: -2; very easy: 2).

<table>
<thead>
<tr>
<th>Tool</th>
<th>-2</th>
<th>-1</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>Mean rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>PhysicalBlocks</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>1.1</td>
</tr>
<tr>
<td>BUILD-IT</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>5</td>
<td>1.4</td>
</tr>
<tr>
<td>Cardboard</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>0.9</td>
</tr>
</tbody>
</table>

Table X: Perceived suitability of the tool to solve the task (not at all suitable: -2; highly suitable: 2).

To take equal account of task characteristics and tool characteristics in the evaluation, we weighted the ratings. For each tool, adding clarity (weight factor: 0.25), difficulty (0.25), and suitability (0.50), the satisfaction became:

- PhysicalBlocks: 1.3
- BUILD-IT: 1.2
- Cardboard: 0.9

We are now ready to summarise the results on all criteria (C1-C5) and take decisions in terms of the original hypotheses (Table XI).

<table>
<thead>
<tr>
<th>Hypothesis</th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
<th>C5</th>
<th>Criteria fulfilled</th>
</tr>
</thead>
<tbody>
<tr>
<td>H1</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>3 (of 5)</td>
</tr>
<tr>
<td>H2</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>2 (of 5)</td>
</tr>
<tr>
<td>H3</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>0 (of 5)</td>
</tr>
</tbody>
</table>

Table XI: Criteria fulfilled per hypothesis. In one of the significant cases (*), we observed the opposite of what was first conjectured: BUILD-IT implied more tested blocks per trial than Cardboard (Fig. 7, first).
Since only one criterion is fulfilled for H1 and only two criteria are fulfilled for H2, we uphold both of them. This means that while there is no significant difference in cognitive support between BUILD-IT and PhysicalBlocks there is a partial difference in cognitive support between BUILD-IT and Cardboard, based on number of blocks tested and on subjective rating.

Besides these results, we find it valuable to summarise the major findings, being:

a) Less time used with PhysicalBlocks than BUILD-IT.
b) Fewer number of blocks tested with Cardboard than with BUILD-IT.
c) Higher satisfaction with BUILD-IT than with Cardboard.

Further, we summarise the minor findings, being:

d) Considerable, but not significant, learning of BUILD-IT use and performance.
e) Considerable, but not significant, learning of Cardboard use, having little impact on performance.
f) Considerable, but not significant, learning of PhysicalBlocks performance.
g) Overall data (all trials) confirmed tendencies observed for single trials analysed.
h) In terms of satisfaction, Cardboard gave considerably lower satisfaction than BUILD-IT. However, the difference in satisfaction between BUILD-IT and PhysicalBlocks is relatively small.

Participants were also asked to justify their choice and to comment on overall system usability. We classified their statements as positive [+] , negative [−] , or neutral [±] , giving (in brackets) the number of participants making the same statement if more than one:

• PhysicalBlocks
  + : interesting, intuitive tool easy to use, haptic feedback made task solving easy
  − : context of task was not known
  ± : task and tool could not be separated, distance had an impact on height

• BUILD-IT
  + : interesting (3)
  − : low accuracy in rotation difficult (5), difficult to learn coordination, side view of little use, slow graphical update, did not perceive that the laser beam had a slope
  ± : unusual tool, usability of side view only became clear later, learning required, higher performance while using visual cues than reflection

• Cardboard
  + : time passes quickly (2)
  − : accuracy and speed were disparate aims, tool is inaccurate and results must be controlled repeatedly, the highest blocks were clearly not candidates for a solution
  ± : some training improved performance considerably (3), unusual tool, gradually changed strategy from trying to reflecting, discovered that small blocks gave solution when situated left and high blocks gave solution when situated right
Discussion and Conclusion

Hence, we see that Physical clearly gives more cognitive support than Cardboard, and that BUILD-IT gives some more cognitive support than Cardboard. However, there is no significant difference between BUILD-IT and Physical. From this result, we draw that an Augmented Reality system like BUILD-IT has the capacity to emulate the physical world in terms of performance, ease of use, and user satisfaction. At the same time, modelling with BUILD-IT is quicker and cheaper than constructing prototypes like Physical. These results came out of an experiment with single users. Their validity for collaborative planning needs further investigation.

We have seen that different aspects of the decision support techniques may explain our results. In a first approach, the representation aids guided us to focus on problem representation, which we realised through the design of alternative planning tools. The information control techniques guided us to study learning (here: trial effect), as a process observable through the actions of the participants. Learning was more observable in BUILD-IT, and less in the other tools, potentially indicating that BUILD-IT is more adapted to learning. It may also mean that BUILD-IT requires more learning. The analysis and reasoning techniques guided us to design tools for different strategies and levels of expertise. Different strategies were best observable in Cardboard, as the performance varied considerable among participants (Fig. 8, left) without being reflected in a similar variation in ease of use (Fig. 8, right). The choice models guided us to design tools facilitating rational decision making. Since participants tested fewer blocks with Cardboard, we assume that this tool had less ability to facilitate rational decision making.

In the future, we would like to quantify the four decision support techniques more in detail, and find out which of these techniques are best suited to understand difference in affordances among different tools.

Acknowledgements

We thank Elke Reuss, Tilman Schildhauer, and Martina Merz for commenting on this paper. Valuable help also came from Martin Bichsel (development), Georg Kralidis (statistical analysis), and Matthias Rauterberg (interaction design and PhD-mentor). PhysicalBlocks was constructed by Hanspeter Honegger. BUILD-IT is a trademark of TellWare GmbH.
7 Camera Control In a Planar, Graspable Interface

ABSTRACT
BUILD-IT is a planning tool based on computer vision technology, supporting complex planning and composition tasks. It allows a group of people, seated around a table, to interact with objects in a virtual scene using real bricks. A plan view of the scene is projected onto the table, where object manipulation takes place. Hence, manipulation and display are coincident. A perspective view is projected on the wall, controlled by a brick acting on a virtual camera. The camera requires control of position, pan, tilt, zoom and roll attributes. However, planar interaction with bricks provides only position and rotation information. The topic of this paper is how to bridge the gap between planar interaction and three dimensional (3D) camera control. This is done by introducing additional, active objects whose positional relation to the camera controls the above attributes.

KEYWORDS
Graspable interaction, camera control, augmented reality

INTRODUCTION

BUILD-IT (Fjeld et al., 1998a, 1998b) is an application that supports designers in the early design phase of floor-planning and configuration tasks. Originally, the application was designed to support providers of assembly lines and plants. However, potential uses range from visualisation and interior design to urban and city planning.

FIGURE 1: BUILD-IT offers a plan view for combined action and perception and a side view with a perspective of the situation.

The system enables users, grouped around a table, to interact in a virtual scene, using real bricks to select and manipulate objects in the scene (Fig. 1). A *plan view* and a *storage space* are projected onto the table, a *side view* on the wall. The storage space allows users to create (delete) objects which are then placed in (removed from) the plan view. The side view offers a perspective of the same scene.

![FIGURE 2: Positioning and rotation (left) and deselecting (right) of a projected object with a brick.](image)

The basic characteristics of the systems are as follows (Fig. 2). Users select an object by putting a brick at the object position. The object can be positioned, rotated and fixed by simple brick manipulation. As soon as the brick is covered, the object is deselected and stays put. To allow for bi-manual interaction and for multiple user operation, the system supports multi-brick interaction. Graphical display is based on the class library MET++ (Ackermann, 1996). 3D objects come from a Computer Aided Design system to BUILD-IT using Virtual Reality Modelling Language (VRML).

Some basic issues of two-dimensional, brick-based interaction were already explored (Fitzmaurice, 1996). The innovation of BUILD-IT, is that the objects are part of a 3D scene. Hence, the topic of this paper is how to bridge the gap between planar interaction and navigation in a 3D scene.
FROM SIMPLE CAMERA CONTROL TO THE EYECATCHER OBJECT

FIGURE 3: The camera object sets the side view. **Travelling** and **pan** are set by brick positioning and rotation.

A virtual camera, situated in the plan view, sets the side view (Fig. 3). Simple camera control can be performed using one brick only. Hence, **travelling** (lateral placement) is set by brick positioning, **pan** (orientation) by rotation of the brick. There can be one or many cameras in the plan view, but only one is effective at a time. The **effective** camera is the one that was manipulated most recently.

To offer users complete camera control, including **tilt** (slope of line of sight), **zoom** (field of view) and **roll** (rotation around line of sight) (Arijon, 1991), it is necessary to overcome the limitations of planar interaction. These limitations are that a brick only provides its position and rotation in the x-y plane.

One strategy which was considered is the use of a specialised brick modelling a real object with added properties. This would require extending the properties sensed by the computer vision input. Since we want to explore software solutions, this approach was not pursued.

Instead the strategy used was to introduce **active**, virtual objects. Active objects feature **intelligent** behaviour and support **complex operations**. They do not have real-world analogues, but are **synthetic**. They only exist during the operation they support, so they are **transient**. In the following, we show an example of how the specific need for camera control led to an active object.

An active object, the EyeCatcher, was realised. With this object, camera attributes depend on the positional relation between camera and object. The EyeCatcher is offered in the storage space (Fig. 4). Using one, or two, instances of this object, it is possible to control the following attributes of the effective camera: **pan**, **tilt**, **zoom** and **roll** (Fig. 5). The pan angle controlled by an EyeCatcher overrides the pan angle set by simple camera control. EyeCatcher orientation has not been given any function so far.
FIGURE 4: Camera, EyeCatcher and other objects are situated in the storage space, left, whereas the plan view with placed cameras and objects is to the right (left). New instances are created by picking an object and placing it into the plan view (right).

FIGURE 5: Using one or two instances of the EyeCatcher object makes it possible to set pan, tilt, zoom (not shown) and roll.
USING ONE EYECATCHER

The *pan* and the *tilt* angles (Fig. 6) are specified so that the EyeCatcher is kept in the centre of the side view (Fig. 7). Since the EyeCatcher normally is situated at *ground-level* in the virtual scene and the camera at *eye-level* (1.6 m above the ground), the camera will tilt downwards. What happens when an EyeCatcher is put onto other objects is discussed below.

FIGURE 6: Schematic drawing of pan and tilt.

FIGURE 7: Using one EyeCatcher and one brick. Side view (left) and plan view (right).
**USING TWO EYECATCHERS**

The *pan* and the *tilt* angles are specified so that the mid-point between the EyeCatchers is kept in the centre of the side view (Fig. 9). The *zoom* angle is specified so that the EyeCatchers are situated at the side view edges (Figs. 8 and 9). The *roll* angle is specified so that the EyeCatchers, as seen in the side view, are connected by a horizontal line.

**FIGURE 8:** Schematic drawing of zoom and roll.

**FIGURE 9:** Using two EyeCatchers and two bricks. Side view (left) and plan view (right).
PUTTING THE EYECATCHER ONTO OTHER OBJECTS
When putting the EyeCatcher onto another object (Fig. 10), it puts itself on top of this object. This effects the tilt and roll angles. Pan and zoom angles, however, are not influenced, as they are independent of EyeCatcher height.

FIGURE 10: Putting EyeCatcher on other objects.

DEACTIVATING THE EYECATCHER
When an EyeCatcher is deactivated it also disappears (Fig. 11); it is transient. However, all the attributes of the camera remain the same. If an other EyeCatcher still remains and is used any further, the situation will adjust to the case of one EyeCatcher. If no EyeCatcher remains, and a brick sits on the camera, then this brick resumes its original control of the camera.

FIGURE 11: Deactivating one EyeCatcher, a second remains.
DISCUSSION
Of particular interest is the fact that the virtual camera attributes have some analogues in real-world camera handling and human perception. Travelling, pan, tilt and roll have direct analogues in human perception, zoom has little. Being more specific, tilt and roll correspond to different head movements. However, when performed, tilt and roll are compensated in reference to some world frame or horizon. Hence, there are issues to resolve about the sense of controlling such attributes which are, at the same time, compensated by humans. In implementation, the same issues arose.

These observations indicate that design of the camera control should largely depend on basic characteristics of human perception. The choice of putting the camera at a human eye-level may be well justified. There seems to be a need for usability studies where alternative control strategies of camera attributes are explored in order to support efficient task-solving behaviour.

CONCLUSION
A concept for 3D camera control based on active objects was realised. Active objects prove an advantage over specialised bricks, since they require no extended input sensing. However, the implementation raises questions about the sense of controlling camera attributes that are compensated in human perception. Finally, it may be of interest to generalise the suggested concept for 3D control to other domains than camera control.
8 Navigation Methods for an Augmented Reality System

Video publication - see separate CD

ABSTRACT
BUILD-IT is a planning tool based on computer vision technology, supporting complex planning and composition tasks. A group of people, seated around a table, interact with objects in a virtual scene using real bricks. A plan view of the scene is projected onto the table, where object manipulation takes place. A perspective view is projected on the wall. The views are set by virtual cameras, having spatial attributes like shift, rotation and zoom. However, planar interaction with bricks provides only position and rotation information. This paper explores two alternative methods to bridge the gap between planar interaction and three-dimensional navigation.

9 Design and Evaluation of Four AR Navigation Tools Using Scene and Viewpoint Handling

Abstract
In an Augmented Reality (AR) system using a brick-based tangible user interface, we present and evaluate alternative techniques for scene navigation. Going from two-dimensional brick-based input to three-dimensional navigation presents design issues. There are two fundamental methods for scene control: scene handling or viewpoint handling. The system has two views (plan and side), presenting action-perception spaces which are coincident and separate. Four tools were developed to explore design solutions, testing the alternative methods in each view. In a quantitative user experiment with a search-and-position task, we evaluated the four tools, measuring performance by trial completion time. Results showed that scene and viewpoint handling performed equally well in the plan view. In the side view, scene handling performed better. Subjective ranking showed that scene handling was always preferred to viewpoint handling. Results indicate that when action-perception spaces are coincident, the choice of handling method is less critical than when separate.

Keywords: augmented reality, tangible user interface, bricks, 3D navigation, usability evaluation

1 Introduction
The context of our research is Augmented Reality (AR), which aims to bring interaction with virtual environments out into the physical world. One domain in AR research is Head Mounted Displays (HMD), another is Tangible User Interfaces (TUI), which we study here. In TUIs, physical objects are used as handles to represent and interact with models in a virtual scene. Previous studies have investigated the use of bricks as input medium for TUIs. Handling of models using brick-based TUIs for three-dimensional (3D) graphics has been explored for simple tasks. The integration of input and display devices, being termed action and perception spaces, is a major concern in the design of TUIs. Such spaces may be coincident or separate, and this is a design issue.

Putting a TUI into effective practice presents new challenges. In real-world applications, the size of the virtual environment often exceeds the physical interface and hence users need means to navigate the scene. Systems with two-dimensional input operating on virtual environments require a mapping between the 2D control surface and the 3D scene. Control of the positioning of a virtual scene may employ two alternative fundamental methods, these being scene handling (SH) and viewpoint handling (VH). To investigate these issues, we extended an existing brick-based TUI to perform navigation in a 3D virtual scene.

The system we worked with has two views, called plan and side view, presenting action-perception spaces which are coincident and separate (Fig. 1). Four new tools were developed in order to explore design solutions, testing the alternative methods in each view. In extending the TUI, the development of the tools was based on two principles for interaction design, being:

- Bimanual interaction (Fitzmaurice et al., 1997)
- Pragmatic and epistemic action (Kirsh et al., 1994)

In a quantitative usability evaluation with a search-and-position task, the four tools were evaluated using protocol data and subjective rankings. We defined performance as trial completion time. Our main interest was to find which handling method will give better performance and which method will be preferred by users. We studied each view separately, but did not compare the views. To examine the handling methods in more detail, we defined and measured user actions in terms of bimanual interaction and epistemic action.

2 Background

2.1 Augmented Reality (AR)

Augmented Reality was introduced by Wellner (1993). The goal of AR, as described by Mackay (1995), is to "allow users to continue to use the ordinary, everyday objects they encounter in their daily work and then to enhance or augment them with functionality from the computer" (like Fig. 2). In the AR research of Mackay, computer information is projected onto drawings so that users can interact with both the projected information and the paper drawing (Mackay et al., 1995). An early brick-based AR system was described by Fitzmaurice et al. (1995). A more recent example showing how AR can support urban planning is given by Arias et al. (2000). A spatially continuous workspace called Augmented Surfaces (Rekimoto et al., 1999) and BUILD-IT (Rauterberg et al., 1997) are two other AR examples.
2.2 Tangible User Interfaces

Tangible User Interfaces (TUI) have been studied by various interaction researchers as a new kind of input medium. An extensive survey of TUIs was given by Ullmer et al. (2000). Tethered bricks, requiring wires, were investigated in the Active Desk by Fitzmaurice et al. (1995). Wireless bricks, detected by a tablet, were later developed by Fitzmaurice et al. (1997). In a more recent use of tangible bricks, the surface is detected from above (Rauterberg et al., 1996; Ullmer et al., 1997). It was shown that a brick-based interface was significantly more effective than a mouse-keyboard-screen interface or a touch-screen (Rauterberg et al., 1996).

2.3 Bimanual Interaction

Two-handed, termed bimanual, interaction calls upon everyday coordination skills such as aligning and grouping (Fitzmaurice et al., 1997). Bimanual brick-based interaction in two dimensions has already been investigated (Fitzmaurice et al., 1995; Ullmer et al., 1997). Bimanual viewpoint control and model handling in 3D graphics interfaces have been studied using two mice, a keyboard, and a screen (Balakrishnan et al., 1999b). The relation between two-handed movements and input performance in separate action and perception spaces has been examined (Balakrishnan et al., 1999a). Here, we make use of bimanual brick-based interaction for 3D scene navigation, and in this way, our work is novel compared to the state-of-the-art.

2.4 Epistemic Action

Depending on users’ acquaintance with virtual environments, navigation can range from epistemic action to pragmatic action. Epistemic (exploratory) action is performed to unveil hidden information or to gain insight that would otherwise require considerable reflection, pragmatic action directly leads to goal attainment (Kirsh et al., 1994). Due to ease of use and direct interaction, Tangible User Interfaces (TUI) may encourage more use of epistemic action (Fitzmaurice et al., 1997) than traditional ones.

2.5 The BUILD-IT System

The TUI we use is based on computer vision technology and is the interface for a multi-user planning tool called BUILD-IT (Fig. 1). Following the tradition of AR, projected light replaces the use of screens as the output medium. Grouped around a table and employing tangible physical bricks, users can select and manipulate virtual models within the scene which they are planning (Fig. 1). The system enables its users to cooperate in a virtual environment for planning a real-world project, such as a room, a school, or a factory. In the BUILD-IT system, the position and orientation of each brick on the table top is determined by a computer vision system. Multiple bricks allow for
uses such as groupware and bimanual interaction. Here, bimanual interaction is studied for single users.

Users have at all times two up-to-date views of the scene they are creating and manipulating: the plan view and the side view (Fig. 1). The plan view is the bird’s eye view from above – which is projected onto the table. The side view is projected onto a screen near the table. In the case of the side view, a virtual camera, which can be located in the virtual scene either outside or inside the plan view, allows the users to choose from which position the side view is to be captured. It can also be zoomed. In the case of the plan view, the entire projection of the scene can be moved around, rotated, or zoomed. The interaction surface also contains a virtual storage space – or menu – for models not in immediate use (Fig. 3). It allows users to create new model instances. Models returned to the menu are deleted.

Figure 3: The system in use: menu with navigation tools (left) and model handling (right).

2.6 TUIs and the Need for Navigation
Several TUIs employ front-projected tables while others employ back-projected workbenches, both giving a plan view (Ullmer et al., 2000). Similar to the BUILD-IT system, some of these offer a second display, giving a perspective side view. TUIs have been applied to a range of application domains, such as visualisation, simulation, modelling, collaborative work, and education. Features such as orientation, scaling, and navigation were offered only by a few of the systems, requiring specialised physical handles. Given the wide use and diverse domains of applications for TUIs, further exploration of navigation tools is worthwhile to study.

Navigation of 3D scenes in TUIs requires viewpoint control (pan, rotation, zoom). However, planar input using physical handles provides only position and rotation. The chosen tool designs have to bridge this disparity between the two-dimensional control surface and the three-dimensional virtual environment.

3 Design of Navigation Tools
To explore alternative handling methods, a series of design choices had to be taken. In 3.1, we first introduce the design options. In 3.2, we explain our initial decisions which led to the experimental factors. In 3.3, we assign update mechanisms and finally, in 3.4, we show how the tools were implemented and are used.

3.1 Design Options
The design options were as follows:

- Action and perception spaces may be coincident or separate according to input and display devices.
Handling methods concern the relationship between user input actions and the resulting effect on the displayed image. One may choose between handling the placement of the scene itself, termed scene handling (SH), and handling the point of view, termed viewpoint handling (VH). Equivalent handling can be seen in two-dimensional browsers like Acrobat Reader 3.0 (Fig. 4).

Update mechanisms concern the update of the display of the scene. They may be continuous or discrete. Continuous update responds directly to user’s handling actions. Discrete update is triggered by the user after handling and then updates the scene.

Degrees of control may be shift, rotation, zoom, tilt, and/or roll, as with a camera.

Physical handles concerns the input devices in a TUI, and how they represent and affect virtual models and tools. They may have a generic or specialised function.

Number of physical handles available may be one, two or many.

3.2 Deciding Experimental Factors

In the context of the system we used, we next settled each of the design options. Two of the options became experimental factors; one was set accordingly; the reminder were fixed.

Action and perception spaces: For plan view interaction, action-perception spaces are coincident. For side view interaction, these spaces are separate. We decided to investigate both views, making this issue an experimental factor.

Handling methods: Within each view of the system, we were interested in alternative handling methods. This issue is the second experimental factor.

Update mechanisms: The update mechanism for each handling method and view was chosen based on logistical requirements of the implementation. Hence, update mechanism did not act as an independent experimental factor. We elaborate on this in 3.3.

The remaining three design options were fixed for the whole experiment as follows:

Degrees of control: First, a pilot-study where users could control all five degrees of control (shift, rotation, zoom, tilt, roll) of the side view showed that they felt uncomfortable. Some reported that the horizon of the side view was too unstable. Second, the plan view is a horizontal work-bench, implying shift, rotation, and zoom as...
the only feasible degrees of control. Third, we tried to make control of each view as similar as possible. Altogether, this gave us grounded reasons for excluding tilt & roll and to offer pan, rotation & zoom in both views.

- **Physical handles:** The use of specialised *physical handles* for navigation of TUIs has been studied in different frameworks. For instance, in a TUI called metaDESK (Ullmer et al., 1997), a physical input device was specially designed for combined zoom and rotation of the plan view. This device showed the limits in the use of specialised input devices, and indicated that the use of generic input devices, such as bricks, might be suited for navigation in TUIs. For the BUILD-IT system, various height tools were evaluated and generic ones preferred (Fjeld et al., in press). Based on these observations and aiming to keep hardware complexity low and software flexibility high, we chose to use *generic handles*, these being rectangular bricks.

- **Number of physical handles available:** With generic bricks and with the need to control shift, rotation, and zoom, each view required two bricks. Since navigation of each view are orthogonal functions and may be performed at the same time, at least four bricks must be available at a time.

### 3.3 Assigning Update Mechanisms

In cross-combining the two factors, being handling method and view, we got four tools. For each tool, the update mechanism used was decided as follows:

- **Scene handling of the plan view:** Update had to be continuous, since discrete update would have given insufficient feedback on user actions and led to a breakdown of the coincident action-perception space.

- **Viewpoint handling of the plan view:** Since the viewpoint is a virtual camera not visible in the scene, it needs to be represented. The representation had to be a model which could be shifted, rotated and scaled. Such model handling could not be solved with a simple scroll bar and needed to operate within the plan view. In this case, continuous update is not possible, hence discrete update was used, employing a reference frame model.

- **Scene handling of the side view:** Since we tried to make handling methods as similar as possible in both views, we chose to perform side view scene handling using continuous scene update.

- **Viewpoint handling of the side view:** To make handling methods as similar as possible in both views, viewpoint handling of the side view was chosen to use discrete scene update, also employing a reference frame.

### 3.4 Tool Implementation and Use

In the following, we give details on the use of the four tools (Table 1). Further details on design and use were given in a video (Fjeld et al., 2000).
The tools can be activated in the menu (Fig. 3). When employing one brick, called the *first brick*, shift and rotation of the controlled view can be set. Adding a second brick, called the *zoom brick*, activates zooming as well. The use of two bricks is called bimanual navigation. For plan view navigation, these two bricks have equal functionality, for the side view they have different functionality. For each tool we explain how to perform bimanual navigation.

**Scene handling of the Plan View: GroundCatcher**

A *GroundCatcher* (Fig. 5) is selected from the menu and as soon as it is placed, it locks to the scene. Subsequent handling controls the plan view with continuous update. A second *GroundCatcher* is likewise selected and locks to another part of the scene. De-selecting the bricks quits the tool.

![Figure 5: GroundCatcher; here zooming and rotating the plan view: zooming in (l.) and zooming out (r.).](image)

**Viewpoint handling of the Plan View: FrameCatcher**

When a *FrameCatcher* (Fig. 6) is selected from the menu, the scene automatically zooms out to show a wider context, and a frame appears. As soon as the *FrameCatcher* is placed within the frame, it locks to the frame. Subsequent handling controls the frame, selecting the desired part of the scene. A second *FrameCatcher* is likewise selected and locks to another part of the frame. De-selecting the bricks triggers a discrete scene update of the framed region.
Figure 6: FrameCatcher, here zooming and rotating the plan view: frame control (l.) and updated scene (r.).

• Scene handling of the Side View: Camera

By selecting the Camera (Fig. 7) from the menu, the part of the scene shown in the side view can be continuously set. Zoom is selected with a second (here: right hand) brick. By moving the zoom brick and the Camera further apart, the side view can be enlarged; by moving them closer, the side view can be focused. De-selecting the zoom brick freezes the zoom. If a second Camera is selected from the menu, the first one disappears.

Figure 7: Camera, here zooming the side view: zooming out (l.) and zooming in (r.).

• Viewpoint handling of the Side View: ViewFrame

When the ViewFrame (Fig. 8) is selected from the menu, the side view automatically zooms out to show a wider context. The desired part of the scene can be selected. Zoom is selected with the second (here: right hand) brick, thus resizing the window as with the Camera. De-selecting the zoom brick freezes the zoom. De-selecting the first brick triggers a discrete scene update of the framed region. If a second ViewFrame is selected from the menu, the first one disappears.

Figure 8: ViewFrame, here zooming and rotating the side view: frame control (l.) and updated scene (r.).

4 Hypotheses

A pilot-study indicated that the strengths of scene handling (SH) using continuous update are direct feedback, lower number of menu selections, and intuitive use. The strengths of viewpoint handling (VH) using discrete update are better tolerance for graphics latency and improved overview for users.
H1. No difference in performance between SH and VH
H2. No difference in bimanual interaction between SH and VH
H3. No difference in epistemic action between SH and VH
H4. No difference in subjective preference between SH and VH

Table 2: Null hypotheses H1–H4.

H1: Due to the established benefits of direct feedback, we conjectured that SH would perform better than VH.
H2: According to the reported benefits of bimanual interaction (Fitzmaurice et al., 1997), we conjectured more bimanual use for the expected higher performing method, this being SH.
H3: According to results concerning epistemic action (Kirsh et al., 1994), we conjectured more use of epistemic action for the expected higher performing method, this being SH.
H4: Considering user preferences, we drew upon the indication that epistemic action may provide an "enhanced sense of engagement of the 3D scene" (Balakrishnan et al., 1999b). Additionally, SH appears to be closer than VH to users’ natural expectations for performing search-and-position problems in the real world. Hence, we conjectured that user preferences would favour SH to VH.

The hypotheses were stated in the conventional null form, as appropriate for statistical testing (Table 2).

5 Usability Evaluation

5.1 Participants and Apparatus

Sixteen graduate students, four women and twelve men, aged 24 to 35, volunteered to participate, being paid a small fee. They had no former experience with BUILD-IT and had to acquire a significant number of skills within the experiment.

The BUILD-IT software runs on a standard 400 MHz PC with 128 MB RAM, 10GB disk space, a frame grabber card, and two OpenGL graphic cards. The computer reads table images with a video camera and provides output via a pair of standard projectors, all sitting in a rack together with an IR source and a mirror. There were five rectangular bricks with reflective material, all identical. All experiments took place in a room without daylight and dimmed ceiling lighting.

5.2 Task Scene and Task

We chose a 3D search-and-position task, which is one of the simplest tasks typically used in studying human performance in computer input control. Other typical tasks, like path following and pursuit tracking, are more difficult (Balakrishnan et al., 1999b). The task was to search for models in a maze and to position each of them at their correct place, requiring the use of both views and bimanual navigation.
The task scene consisted of a maze and an elevated cubic box with four faces and face replicas on the ground (Fig. 9). Each face was split into nine cells, one of which contained a stimulus, identified by colour (red, yellow, green) and form (disk, square, triangle). Due to shields, each stimulus was visible only when viewed from the front of the face. On the ground was a replica of each face for model placement. For each stimulus, a model with the same colour and form was hidden in the maze. Since a pilot study showed that four models were too time consuming, we simply required any two of the four models to be found and positioned. The task was defined as follows: For any two of the four stimuli, not necessarily sequentially (Figs. 9a-d):

**Figure 9:** Four stages of a trial; left half is plan view, right half is side view: a) start-up situation with cubic box containing stimulus, b) searching for models in the maze, c) a model has been found, and d) the model is positioned at the correct cell of the face replica on the ground.

i) View colour and form of stimulus (a).
ii) Search for and retrieve matching model (b-c).
iii) Position model at the correct cell of the face replica on the ground (d).

We generated twelve variations of the task differing in colour and form of the stimuli and in position and orientation of the cubic box and models. Four of these were used as demo and practice tasks, eight were used as main tasks. The main tasks were permuted giving eight experimental trials per participant.

### 5.3 Operationalization

**H1.** Good performance was operationalized by low trial completion time, measured from stimulus appearance until the last model or tool was de-selected.
H2. To select *zoom brick* from the menu and thereby to start a bimanual interaction sequence, the system requires a *first brick* to be activated and stay selected. Hence, *bimanual interaction* in navigation was measured by the number of zoom selections from the menu.

H3. We assume that users performing pragmatic action will not stop for reflection in the middle of a movement. When they stop, we consider them to reflect and the action is classified as *epistemic*. This can be classified by the number of step-by-step brick movements and was measured when a brick was re-positioned after a pause, ignoring pauses shorter than half a second. (This is more than the system latency and the sampling interval for protocol logging.) The *first brick* and the *zoom brick* were registered separately.

H4. Preference was operationalized by asking participants to i) rate their satisfaction with each tool on a [very low, low, high, very high] scale indexed by grades [4,3,2,1] and ii) indicate one preferred tool per view and justify their preference.

H1 is one-sided using two categories. H2 and H3 are one-sided using ordinal data with several categories. H4 is one-sided using interval data.

### 5.4 Design

A two-by-two within-group design, enabling the investigation of subjective tool preference, was used. Pairing each plan view tool with each side view tool resulted in four experimental conditions (Table 3).

<table>
<thead>
<tr>
<th>Experimental condition</th>
<th>Plan view</th>
<th>Side view</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SH:Ground Caster</td>
<td>VH:Frame Caster</td>
</tr>
<tr>
<td>1st</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>2nd</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>3rd</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>4th</td>
<td>x</td>
<td></td>
</tr>
</tbody>
</table>

Table 3: The two-by-two design gives four conditions.

Each participant was given a permuted sequence of the four conditions. To equalise potential learning effects, sequences were chosen so that always two new tools appeared in the first condition, one new tool appeared in the second and third condition, and no new tools in the fourth condition. This resulted in 16 of the 24 possible permutations, hence the number of participants. The presentation order of the main tasks was chosen by the latin square procedure so that task-condition combination was counterbalanced. The order of the demonstration and practice tasks did not vary.

### 5.5 Procedure

Each participant was welcomed by the investigator and performed the experiment seated next to him at the table.

*Introduction*: The investigator explained the system by loading a *furniture scene*. Hardware components were briefly pointed out: rack, mirror, camera, computer, two
projectors, and five bricks. Operation was demonstrated in terms of plan view, side view, menu, model selection, positioning, rotation, and de-selection. Assisted practice with one, then five bricks, was given. Navigation was introduced and handling methods, as found in Acrobat Reader (Fig. 4), were explained. It was shown how SH and VH had been implemented for the plan and the side view (Table 2) and how the four tools combine into four conditions (Table 3). Taking the tools given by the first condition for the participant, navigation was demonstrated for each of the plan and side views, first with one, then with two bricks. Introduction lasted approx. 25 minutes.

**Trials sets:** The task scene was now loaded. There were four trial sets according to the sequence of tool pairing conditions given for the participant. For each trial set, the investigator explained and performed a demonstration task with one model (Fig. 9); the participant did an assisted practice task with two models; then performed two different main tasks as fast as possible. The investigator checked task completion and initiated the next task. The eight main tasks formed the experimental trials. Trial sets lasted approx. 45 minutes.

**Subjective rating:** At the end of the experiment, the participant rated their satisfaction with each tool, then selected the preferred tool per view and justified their choice. Subjective rating lasted approx. 10 minutes.

### 5.6 Logging

The software was instrumented to log brick movements in real time, giving the position and kind of virtual model the bricks operated on. A sampling rate of approx. 0.3 sec. was used. From the log of each trial we extracted i) trial completion time (tct), and for each tool ii) number of zoom selections (nzs), and iii) number of re-positionings of first brick (nfrp) and zoom brick (nzrp).

### 6 Experimental Results

Our aim was to examine within each of the views differences in performance and use between the alternative tools, these being based on SH and VH. However, participants employed two complementary tools at a time, one for each view, giving the four experimental conditions (Table 3). In order to test the hypotheses H1 – H3 (Table 2), the analysis needed to reveal individual differences within each view between the alternative tools. Consequently, much like Balakrishnan et al. (1999b), we used a multiway ANOVA, here with a General Linear Model (GLM). The five independent variables were plan view method (SH, VH), side view method (SH, VH), trial, task, and user. The four dependent variables were trial completion time (H1), zoom selections (H2), first brick and zoom brick re-positionings (H3). For H2 and H3, we analyzed these separately for plan and side view navigation. The Bonferroni method was used with k = 7, alpha = 0.05/k = 0.007. Significant effects are shown by p-values less than alpha are marked by a star (*). Below, we give the ANOVA results for the strong cases (Tables 4 – 6) and for the significant effects note the supporting data averaged over trials.
6.1 Trial Completion Time (H1)

<table>
<thead>
<tr>
<th>Source</th>
<th>df</th>
<th>F-ratio</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plan view method</td>
<td>1</td>
<td>0.391</td>
<td>p = 0.533</td>
</tr>
<tr>
<td>Side view method</td>
<td>1</td>
<td>8.144</td>
<td>p = 0.005 *</td>
</tr>
<tr>
<td>Trial</td>
<td>7</td>
<td>5.210</td>
<td>p &lt; 0.001 *</td>
</tr>
<tr>
<td>Task</td>
<td>7</td>
<td>3.146</td>
<td>p = 0.005 *</td>
</tr>
<tr>
<td>User</td>
<td>15</td>
<td>2.063</td>
<td>p = 0.018</td>
</tr>
</tbody>
</table>

Table 4: Trial completion time: Significant effects for side view method, trial, and task.

**Plan view method (Table 4):** No significant effect and H1 is upheld.

**Side view method (Table 4):** SH (tct=150s.) gave better performance than VH (tct=183s.) and H1 is rejected.

**Other effects (Table 4):** Trial (learning effect) and task had a significant effect.

6.2 Bimanual Interaction (H2)

<table>
<thead>
<tr>
<th>Source</th>
<th>df</th>
<th>F-ratio</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plan view method</td>
<td>1</td>
<td>11.885</td>
<td>p &lt; 0.001 *</td>
</tr>
<tr>
<td>Side view method</td>
<td>1</td>
<td>0.053</td>
<td>p = 0.818</td>
</tr>
<tr>
<td>Trial</td>
<td>7</td>
<td>0.583</td>
<td>p = 0.768</td>
</tr>
<tr>
<td>Task</td>
<td>7</td>
<td>4.376</td>
<td>p &lt; 0.001 *</td>
</tr>
<tr>
<td>User</td>
<td>15</td>
<td>1.715</td>
<td>p = 0.061</td>
</tr>
</tbody>
</table>

Table 5: Zoom selections in plan view navigation: Significant effects for plan view method and task.

**Zoom selections in plan view navigation (Table 5):** More zoom selections per trial with SH (nzs=2.4) than VH (nzs=1.5) and H1 is rejected. Task was significant.

**Zoom selections in side view navigation:** No significant effects.
6.3 Epistemic Action (H3)

<table>
<thead>
<tr>
<th>Source</th>
<th>df</th>
<th>F-ratio</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plan view method</td>
<td>1</td>
<td>14.188</td>
<td>p &lt; 0.001 *</td>
</tr>
<tr>
<td>Side view method</td>
<td>1</td>
<td>2.177</td>
<td>p = 0.143</td>
</tr>
<tr>
<td>Trial</td>
<td>7</td>
<td>3.119</td>
<td>p = 0.005 *</td>
</tr>
<tr>
<td>Task</td>
<td>7</td>
<td>3.365</td>
<td>p = 0.003 *</td>
</tr>
<tr>
<td>User</td>
<td>15</td>
<td>4.731</td>
<td>p &lt; 0.001 *</td>
</tr>
</tbody>
</table>

Table 6: First brick re-positionings in plan view navigation: Significant effects for plan view method, trial, task, and user.

First brick re-positionings in plan view navigation (Table 6): Fewer first brick re-positionings per trial with SH (nfrp=14.9) than VH (nfrp=21.6) and H3 is rejected. Trial (less use), task, and user had a significant effect.

Zoom brick re-positionings in plan view navigation: More zoom brick re-positionings with SH (nzrp=13.0) than VH (nzrp=5.1) and H3 is rejected. Task was significant.

First brick re-positionings in side view navigation: Significant effects for trial (less use) only. H3 is upheld.

Zoom brick re-positionings in side view navigation: No significant effects and H3 is upheld.

6.4 Subjective Preference (H4)

<table>
<thead>
<tr>
<th>Tool</th>
<th>Rating</th>
<th>Mean rating</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>v.low, -2</td>
<td>low, -1</td>
</tr>
<tr>
<td>GroundCatcher</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>FrameCatcher</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>Camera</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>ViewFrame</td>
<td>1</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 7: Overall tool rating selections.

<table>
<thead>
<tr>
<th>View</th>
<th>Tool</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plan view</td>
<td>GroundCatcher</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>FrameCatcher</td>
<td>4</td>
</tr>
<tr>
<td>Side view</td>
<td>Camera</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>ViewFrame</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 8: Tool preference per view.

Table 7 gives subjective ratings; showing for each tool the number of participants selecting each rating. The indices are shifted to a balanced scale [-2,-1,1,2] and the mean
rating is given. We note that the mean rating of SH tools was higher than for VH tools. Table 8 gives for each view the number of participants who preferred each tool. In both cases, SH tools were preferred to VH tools for both views and H4 is rejected.

6.5 Further Subjective Statements
Participants were also asked to justify their choice and to comment on overall system usability. We classified their statements as positive [+] , negative [−], or neutral [±], giving (in brackets) the number of participants making the same statement if more than one:

- **GroundCatcher**
  + : feedback (7), ease of use (4), combined shift and zoom, engagement of 3D scene, underpins search
  − : 3D graphics slow, zoom requires learning

- **FrameCatcher**
  + : provides overview, gives a steady image
  − : plan view orientation

- **Camera**
  + : feedback (10), plan view orientation (5), ease of use (2)
  − : separate action and perception spaces, zoom not needed

- **ViewFrame**
  + : side view orientation (2), provides overview
  − : plan view orientation (2)

- **Overall system usability**
  + : engagement of the 3D scene, interesting tools
  − : unwanted de-selection through shadowing of bricks (4), operation tiresome, 3D graphics is slow, active state of plan view tool unclear, no undo
  ± : needs practice (2), frame with SH maybe of interest

7 Discussion
We discuss the major results according to the views.

7.1 Plan View
*Scene handling* (SH) and *viewpoint handling* (VH) performed equally well for plan view navigation – being against our expectations. So the strengths of direct feedback in SH did not appear as performance benefits. However, SH was preferred to VH in subjective ratings and obtained more positive user statements, particularly related to *feedback*. There is more *bimanual interaction* in SH than VH. *Epistemic action* shows opposite effects for *first brick* and *zoom brick*. For the *first brick*, less *epistemic action* in later trials indicates a learning effect.

7.2 Side View
SH outperforms VH for side view navigation – confirming our expectations. Also users rated SH higher than VH, mainly due to better SH feedback. *Bimanual interaction* and
epistemic action gave equal results for SH and VH and do not explain the difference in performance. This leaves the subjective factor of direct feedback in SH as a likely cause. Again, for the first brick, less epistemic action in later trials indicates a learning effect.

8 Conclusion

We presented the design of four tools for navigation in an Augmented Reality system; two of them based on scene handling (SH), the other two based on viewpoint handling (VH). The two system views, called plan and side view, present action-perception spaces which are coincident and separate. One tool of each handling method was used to control each view.

A usability evaluation of the tools was undertaken with 16 users, recording trial completion time, user actions and preferences. We tested hypotheses regarding performance, bimanual interaction, epistemic action, and subjective preference. Expressed in terms of handling methods, and generalising from the views to action and perception spaces, our three main findings for this system were:

1) When action and perception spaces coincide, SH and VH perform equally well.
2) When action and perception spaces are separate, SH performs better than VH.
3) Users prefer SH in both cases.

The results indicate that when action-perception spaces are coincident, the choice of handling method is less critical than when they are separate. For future research, the design choice of update mechanism and its influence on task solving efficiency is worthy of further evaluation. Due to space, we do not report on details related to symmetry in bimanual interaction. We leave it for others to determine how far our findings apply to other Tangible User Interfaces.

Acknowledgements

We thank Scott MacKenzie, George Fitzmaurice and Terry Winograd for commenting on an earlier version of this paper. We also thank the INTERACT reviewers for constructive feedback. Valuable help came from Domenico Signorello (task design), Martin Bichsel (development), Fred Voorhorst (design), Christoph Schierz (experimental design), Georg Kralidis (statistical analysis), and Matthias Rauterberg (interaction design and PhD-mentor). BUILD-IT is a trademark of TellWare GmbH.
PART III:
ADVANCED USES,
DISCUSSION, AND CONCLUSION
10 Advanced Uses and Scenarios of the BUILD-IT System

In this section, we report on four advanced uses and two future scenarios of the BUILD-IT system. Two of the advanced uses were fully developed at the Swiss Federal Institute of Technology (ETH) Zurich in Switzerland. The two other advanced uses employ the BUILD-IT technology and were realised at the Technical University of Eindhoven (TUE) in the Netherlands. The two future scenarios represent planned development and use of the BUILD-IT technology. Both scenarios are foreseeing a cooperation between different resource centres at the ETH, being Architecture, Product Development, Work Psychology, and Applied Physiology.

Advanced Use 1: Communicating with CAD Systems (ETH)

For most planning tasks in systems engineering and architecture, drawings and two-dimensional (2D) models have been replaced by Computer-Aided Design (CAD) applications. This change has brought about a range of supportive tools for drawing and information processing. However, it also implies less immediate contact among CAD users, planning experts and sales people.

We began our work by performing a task analysis with potential user groups for our system. We observed that they spent a great deal of time in discussions with their clients and noticed that off-line CAD support is hardly available during sales trips. This lack of support sometimes caused misunderstandings with the designers ‘at home’, trying to communicate their solutions to the travelling sales people. Also, some of the customers were not familiar with 2D layout techniques; they were unable to imagine what an object would look like in three dimensions. Therefore, an easy-to-handle, three-dimensional (3D) planning tool proved to be of high interest to planning experts and to sales people. A distributed, networked system would additionally allow for interaction among users located at different sites.

Actually, modern management concepts like Simultaneous Engineering are based on dynamic interaction among cooperating experts. In this context, a tool should encourage team cooperation rather than each person working in front of a separate screen. Such needs can hardly be met by existing technologies like video conferencing. An adequate solution has to offer a more intuitive, natural interaction.

All these considerations were taken into account in the design process of BUILD-IT. A system bringing support to prototyping and design processes was the result. This tool is not intended as an alternative, but rather as a complementary aid for CAD systems. It allows for ready-made applications in various field, such as machine configuration, city and urban planning, architecture and interior design.

Compared with physical, model-based layout systems, BUILD-IT additionally offers handling of CAD-objects and data management. It features cheaper, quicker and more exact object representation. The potential of computer mediated work is made readily available through automatic calculation of prices and time-to-delivery. Two-way communication with external CAD systems is assured, whereas animation and simulation offer design support at an expert level.
Working with VRML Data and Meta-data

The BUILD-IT system understands two different 3D-CAD data formats: Virtual Reality Modeling Language (VRML) data and meta-data. We will focus on VRML data, because they describe the complete geometry and visual characteristics of an object.

Additionally, depending on the field of application, users also need auxiliary object information. First, if configuration costs of the currently handled object is of interest, product name and unit price may be required. Second, in the case of process simulation (e.g. welding of metal sheets), different objects (e.g. a robot, a welding or a cleaning machine) and their characteristics (e.g. machine type, capacity, preparation-, processing and welding time) are needed. In both cases, object specific numbers and figures, named meta-data, are required. Such information is treated as separate data structure(s), and stored as meta-information (".mif") files.

Data exchange between a 3D-CAD system and BUILD-IT can be handled in two different ways: i) by the CAD-connection, and ii) by the Product Data Management (PDM)-connection.

**Figure 1: Data flow, between the 3D-CAD, the PDM and the BUILD-IT system, is based on CAD-connection, PDM-connection, integration and parts list integration.**

**CAD-connection**

The most direct connection between a 3D-CAD system and BUILD-IT is the CAD-connection (Fig. 1). CAD users are presented with a list of all available objects and can select the geometric data required for their specific planning session. The selected geometric data is converted to VRML format and offered by the CAD system as world (".wrl") files. Using the CAD-connection, the selected geometric data is then sent as ".wrl" files to BUILD-IT. For each ".wrl" file, a ".mif" file is generated. A ".mif" file contains additional object information like unit price and simulation parameters.

A VRML based connection offers the important advantage of data compression, allowing for reduced information flow and less object complexity. This feature is just as vital to object handling on the Web as with the BUILD-IT system. Without data...
reduction, only high performance CAD systems would be able to deliver multiple 3D
object within acceptable time.

However, conversion, i.e. data compression and complexity reduction, also induce
serious limitations. Circles are displayed as multi-edge polyhedrons, preventing an exact
geometric object interpretation. Users wishing to position one object along the tangent
of a second, circular object, with millimetres precision, can no longer be supported. A
further consequence of data conversion is that direct feedback from BUILD-IT to the
CAD system cannot be offered. Such feedback is impossible, because exact volume and
surface information are lost through conversion, and the original parts of an object can
no longer be reconstructed. For this very reason, bi-directional communication of
geometric and configuration data is not possible with the direct CAD-connection.

To make the description of the CAD-connection complete, we mention that meta-data,
in this case "mif" files, are also communicated via this connection. Since meta-data is
exclusively being used by the BUILD-IT system, no feedback is needed, so the one-way
CAD-connection is fully sufficient in the context of meta-data.

PDM-connection and integration

A more elaborate way to connect BUILD-IT with CAD systems became possible with
the advent of PDM systems. PDM systems do not only manage geometric data, they also
offer product information such as parts lists. Parts lists are normally managed by larger
database systems. By complete integration (Fig. 1) of the PDM and CAD systems,
geometric data can be converted into VRML data without having to interact with the
CAD system. Selected objects are actually converted into VRML and meta-data in one
integrated operation, called transfer process (Fig. 1). This process is similar to the
VRML-converter and the meta-data file generator, put together.

There is one major difference between the CAD and the PDM-connection. As soon as a
PDM user selects an object, a pointer is set on the corresponding parts list. This pointer
is stored in the object's meta-data. Supported by such pointers, it is possible, at any time,
to load original parts lists and geometric data from the PDM system and to display them
with the CAD system.

Connecting PDM with 3D-CAD and BUILD-IT systems, opens up new possibilities, far
beyond managing parts list and geometric data. The main advantage of this combination
is the concept of parts list integration (Fig. 1). Parts list integration means bi-directional
communication between the PDM system and BUILD-IT. Henceforth, it is possible to
harvest the full advantages of a BUILD-IT planning session.

BUILD-IT users can assemble objects without having to care about causing any harm to
original parts lists or geometric data. As soon as a planning process is accomplished,
BUILD-IT generates a coordinate list. The list is the final result of the planning session
and describes all the assembled objects. Supported by the parts list integration,
communicated via the PDM system, the CAD system can now access, integrate and
display the result of the planning session. Object modification that took place during the
BUILD-IT planning session has no effect beyond that session unless required.

Advanced Use 2: Simulation (ETH)

Important aspects in the design of assembly and manufacturing lines, are the expected
behaviour and the performance of a configured layout. In order to support designers
clarifying such aspects, a simulation program, SIMPLE++ (AESOP, 1997), was connected to BUILD-IT to carry out simulation of designed production systems. To merge the simulation capabilities of SIMPLE++ with the interaction features of BUILD-IT, the same model machines, processing stations, stores, conveyors and transporters must be offered by the BUILD-IT object menu as by the SIMPLE++ model menu. When the layout design is finished, BUILD-IT internally represents the outcome as a graph. In this graph, vertices correspond to the objects, edges link neighbouring objects. The graph is then stored in a file readable by SIMPLE++. Hence, corresponding models of the SIMPLE++ menu can be called upon, and integrated in the simulation model. Further information needed for simulation, such as set-up, processing, and interruption times of each processing station, are defined interactively by the designer. Once this data has been transferred to SIMPLE++, simulation can take place. While the simulation is running, all events triggered by the SIMPLE++ simulation, are logged according to the following data set:

- Name of the object that triggered the event
- Name of the object where the event actually took place
- Point in time when the event occurred

Figure 2: Data flow between SIMPLE++ and BUILD-IT.

As soon as an event has taken place, this data set is sent back to BUILD-IT. Hence, SIMPLE++ provides the information necessary for BUILD-IT to perform animation. Figure 2 shows the data flow between SIMPLE++ and BUILD-IT.

Additionally, simulated data can eventually be provided according to the users’ needs. The following example will clarify the general concept, and then focus on the additional, simulated data.

Implementation

The example considered here comes from one of the BUILD-IT industrial partners (Soudronic AG), manufacturing and selling welding systems for metal sheets. We call the engineers of this partner company the designers, we call the clients the customers. Both designers and customers can be seen as users of the BUILD-IT system.
Before a welding system can be designed, it is necessary to determine the customers’ requirements. There are different parameters that give the designer an idea about which type of welding system is appropriate to meet customer needs. Some of these parameters are:

- Size of the sheets to be welded,
- required set-up time of the welding system for producing each type of metal sheet,
- welding speed,
- production rates, i.e. number of metal sheets to be welded for a given period of time, and
- time-to-delivery of the produced sheets.

Figure 3: Layout of a welding system.

Figure 3 shows the layout of a common welding system. It consists of: a mobile destacking station (1) with 2 or more cars, where the sheets to be welded are loaded and carried onto the welding line (2-7); destacking robots (2) grab the sheets and place them in the welder (3); sheets on both sides of the welder table are welded together by passing them through the laser beam (4); then, sheets are cleaned in a seam cleaning station (5); a dimpling press (6) and a turning station (7) prepare the sheets for stacking; finally, a stacking robot (8) puts the sheets onto the cars of the mobile stacking station (9).

According to production rates required by the customer, a welding system can be designed with one or more welders, each of them has to be equipped with stacking, destacking, cleaning, turning stations, dimpling press and (de-)stacking robots.

Depending on the number of cars used, stacking and destacking stations, can be double (4 cars) or simple (2 cars). The designer has to find out, which combination of processing stations best meets the customers’ requirements considering performance and cost.

Thus, a first design was made, its performance estimated, and discussed with the customer. If the customer wants to modify parts of the proposed design, the designer has to take action and call in for a new discussion. This can be a time consuming task, that can be significantly improved by using BUILD-IT in connection with SIMPLE++. 
To make the two systems communicate, each of the processing stations of the welding system (Figure 3) has to be modelled both as a BUILD-IT object and as a corresponding SIMPLE++ model. Also, metal sheets to be welded existed in both systems. Using BUILD-IT, customers and designers add, delete and replace objects in the layout. They also assign the type (or types) of metal sheets to be processed and produced in each production run. After the customer has agreed with the resulting design, simulation mode is activated and the layout of the welding system is automatically expressed as a graph file, readable by SIMPLE++. In this stage, simulation models can be constructed. By means of a program that takes, as input, the above-mentioned parameters, processing and set-up times of each station are computed and assigned to the corresponding objects.

In this state, simulation is automatically started. Each time an event takes place, it is registered and sent to BUILD-IT, where animation of the layout is carried out. Relevant data about the performance of the welding system, such as, number of welded sheets and system status are produced throughout simulation. Additionally, the storage of produced sheets is simulated and displayed to give the users an idea about the size of the required storage area with given production rates and times-to-delivery requirements.

**Advanced Use 3: Three-dimensional Manipulation for Volume Data Browsing (TUE)**

A further advanced use of the BUILD-IT system was reported by Subramanian et al. (2000) at the Technical University of Eindhoven in the Netherlands. Based on the VIP platform, which is a modified version of the BUILD-IT system, they have developed the Rigid Intersection Selection Prop (RISP).

![Figure 4: Rigid Intersection Selection Prop (RISP) in use (Subramanian et al., 2000).](image)

Subramanian et al. (2000) suggested the following interaction style: "The user will be provided with a RISP with which (s)he can view cross-sections for investigation (Fig. 4). The RISP can be operated with the non-dominant hand and will be tracked using the cameras and computer vision techniques. The user can interact with the cross-section by means of a pointer prop (PP) in the dominant hand. The pointer prop will also be tracked using computer vision techniques" (Subramanian et al., 2000). This use of the BUILD-IT technology is similar to the Personal Interaction Panel (PIP) (Zsolt, 1999) developed by a computer graphics group in Vienna. However, RISP also takes into account design requirements for a more natural interface (Subramanian and Ijsselsteijn,
In particular, the VIP platform is free of intrusive devices and there are no wires hindering free movement. This gives RISP an advantage over PIP.

Other fields of application related to RISP are i) exploration of geological data for determining the potential of a gold mine (Johnson and Bacigalupo-Rose, 1993) or an oil well (Fröhlich and Plate, 2000; Smith, 1999) and ii) exploration of car collision data (Smith, 1999) to analyse the potential damage to passengers. Subramanian et al. (2000) report that "the task that occurs repeatedly in all the above applications can be described as obtaining spatial awareness, selecting cross-sections and planning trajectories."

**Advanced Use 4: Asymmetric Networking (TUE)**

As a final advanced use of the BUILD-IT system, de Greef and Ijsselsteijn at the Technical University of Eindhoven in the Netherlands have suggested an asymmetric network set-up. Reusing the BUILD-IT-based VIP platform, they have developed a Photo Share tele application (de Greef and Ijsselsteijn, 2000).

![Photo Share tele application](image)

Figure 5: The PhotoShare tele application (de Greef and Ijsselsteijn, 2000). The presenter is left, the viewer is right.

Connecting two VIP platforms over a network, they studied the effect of social presence and satisfaction of informal home-environment users. The networked application offers shared viewing of photos, like holiday and family pictures, between two (or more) users at remote locations (Fig. 5). The distributed platform is asymmetrical, meaning that there was a presenter at one side and a viewer at the other side. The presenter uses the plan view to select pictures to be showed in a scaled-up version in the side view. The distributed platform also offered audio and video to enable so-called natural communication between the distant users. The side view also offered an optional communication space which was used to run a video conference between the two sides. As their most important finding, de Greef and Ijsselsteijn reported "the communication space had a positive effect on the feeling of social presence and that extensive functionality in the action-perception space was of little importance and could even diminish social presence".

**Future Scenario 1: Symmetric Networking: CADShare (ETH)**

Planning of complex machines, buildings and plants involves many partners situated at different locations. Such experts can cooperate only by sending each other different versions of their planning result, for instance Computer-Aided Design (CAD) models. But since they are separated in space and time, they do not really work closely together. One way to overcome this separation is to connect the planning experts by networking
their tools. In this proposition, we combine the ideas of networking design tools with a novel way of interaction called graspable user interfaces.

A task analysis was carried out with 16 planning experts from the machinery and process industries as well as from the fields of architecture and interior design (Fjeld et al., in press). In each of these settings the planning processes were analysed by interviewing, video-typing and studying documents. The experts explained complex projects and relevant examples of design and implementation. In addition, they answered questioned about interaction with colleagues and customers, about the socio-technical organisation of their companies, and about available support technology. An outcome of the task analysis was at least two situations in which a distributed, graspable interaction tool could be useful:

1. **Cooperation among planning experts with different sets of skills**: As industrial projects become increasingly complex and time constraints grow in an ever-changing global economy, the promotion of Concurrent Engineering and teamwork is essential. Such teams need an infrastructure that stimulates dynamic interaction among cooperating experts, rather than one-person, one-screen set-ups. A distributed, graspable interaction tool can serve to enhance common understanding of the planning process. It could help make clear how diverse design elements, promoted by different experts, may be combined. This is of particular importance for early decisions on size and function of modules. Graspable interaction and three-dimensional (3D) large-scale visualisation, which is offered by BUILD-IT, could support this early phase, where experience and intuition lead to decisions.

2. **Communication with customers**: Customers are often laypersons to the planning tasks and are not used to thinking in terms of two-dimensional (2D) views. Therefore, a play-like interaction can be the most valuable form of communication. At the same time, customers are considered as a valuable source of information on the object or setting to be planned. Lay advice may come from operators of machinery, safety specialists in a factory or service people in a hospital. Involving the customer in the planning process is also a marketing argument, creating a higher customer commitment to the resulting product. A typical example is interior design.

For both these situations, networking BUILD-IT would serve as a pre-CAD complement in the early stages of design. According to Drisis (1996), systems like BUILD-IT could also stimulate the evolution of CAD systems themselves. Drisis predicted that new forms of interaction, for instance graspable user interfaces, could challenge the programming paradigms of CAD design.

As a first step of this project, some fundamental questions required clarification:

- **Number of systems**: Should two or more system work together in a distributed set-up?
- **Master-Slave**: Should one system be master and the other ones slaves?
- **Computing priority**: Should it be granted to assure local update or to consistency among networking systems?
- **Networking**: How to use Tokenring, Ethernet, Internet, TCP/IP Sockets?
- **Deadlocks**: How is deadlock analysis performed for the system and what guidelines apply for the design?
Answers to these questions may be found in areas like Human-Human Interaction (HHI), Human-Computer Interaction (HCI), Computer Supported Cooperative Work (CSCW), and Usability Engineering (UE).

In a second step, having answered the preceding questions, an implementation phase will follow. Implementation will be directed accordingly:

- Based on simulation files, a mock-up of two networking systems is simulated.
- Protocol is described.
- Master-slave concept or alternative control management is worked out and realised.
- One network solution is chosen and implemented.
- Start-up and shut-down procedures are designed and realised.
- Control algorithms for intra-system consistency are described and realised.
- Control algorithms for consistency among networking systems are described and realised.
- The system is realised in Java programming language, which is the same language as the latest version of the BUILD-IT software.

Taking a step aside, we close this scenario by discussing its innovative content. The strength of combining ubiquitous computing with networking was first pointed out in the cutting-edge work of Mark Weiser (1993). The innovative feature of the Computer-Supported Cooperative Work (CSCW) system BUILD-IT (Fjeld et al., 1999, in press), beyond the brick-based interaction, is that the objects are part of a three-dimensional (3D) setting. Two-dimensional, planar brick-based interaction has already been explored (Balakrishnan and Kurtenbach, 1999; Fitzmaurice et al., 1995). Also, bimanual navigation and object manipulation in 3D graphics interfaces has been investigated (Ullmer and Ishii, 1997) using two mice, a keyboard and a screen. In the current version of the BULD-IT system, the strengths of these two approaches are combined. Not only does the system offer a novel way of interaction. The multi-user nature of BUILD-IT overcomes a serious drawback often seen with CSCW tools, namely that they are based on single-user applications (Grudin, 1988). Networking combined with the current BUILD-IT systems offer a unique vantage point for distributed planning.

**Future Scenario 2: Teaching Network (TN) (ETH)**

This project is a direct result of the BUILD-IT competition, where five teams of architecture students designed free-form virtual scenes. As an outcome of that competition (Fig. 6), it became clear that BUILD-IT was well suited to pursue goals in the domain of teaching with new technologies. Drawing on i) the experience gained in the competition and ii) an established, trans-disciplinary network of lecturers at ETH Zurich, new technologies are currently being introduced in various courses and teaching methods. In this context, BUILD-IT will be employed as a case-study for new technology in teaching. At the same time, web-based environments will enhance and

---

14 The description given here is based on the project proposition and the project web page: http://caad.arch.ethz.ch/research/buildit_tn/.
sustain the communication and information exchange among the participants - teachers, students, and the Teaching Network (TN) project team.

Figure 6: Typical situation when mechanical engineering students are using BUILD-IT.

The main goal of the Teaching Network is to help teachers to get confident with new technologies and to integrate such technology quickly into the curriculum they lecture. The Teaching Network is a trans-disciplinary project, integrating expertise and know-how of various ETH institutes: Architecture + CAAD (CAAD), the Centre for Product Development (ZPF), the Institute of Work and Organisational Psychology (IfAP), and the Institute of Hygiene and Applied Physiology (IHA). It will follow these steps:

**Step 1:** The development of Strategies for the Integration of New Technologies in the Curriculum is aiming to:
- Make teachers and students confident with new technologies and tools,
- facilitate dialogue and negotiation of ideas between teachers,
- allow teachers to share, transform and build on each others insights,
- produce and spread situated knowledge across disciplines (synergy effect), and
- create strategies and mechanisms responding to the needs of academia and industrial applications (e.g. ETHWorld and the Virtual Campus of Switzerland).

**Step 2:** A case-study in collaborative teaching using modern learning tools, aiming to:
- Support and enhance simultaneous computer-mediated team work (up to 6 people),
- allow for intuitive ways of human-computer interaction (no prior knowledge required, easy to learn),
- be employed as a co-located multi-user and distributed multi-site environment,
- offer teachers and students the opportunity to explore and develop new teaching and learning methods, such as: cycles of generation and evaluation, project-orientation, new forms of collaboration, interesting ensuing artefacts, and
- enable the use of novel technologies, such as Mobile Computing and Design Virtual Environments (DVE).
Step 3: Establish an on-line environment facilitating communication and information flow in the Teaching Network. This implies:

- Supporting the exchange of contributions, ideas, and experience between all participants, especially between the teachers (new teaching methods), but also between the students and the project team,
- preserving and displaying a group’s achievement, so that others can build on it and thereby assuring transparency,
- sustaining collaboration and knowledge transfer over time and space, and
- offering technical support for the participants.

Step 4: Establish scenarios by testing and exploring the integration of BUILD-IT into existing courses. This implies:

- Developing scenarios that integrating BUILD-IT in existing courses, collaborating with faculty from different disciplines, such as architecture, urban design, building construction, product development, chemistry, and pharmacy,
- rethinking traditional teaching methods and develop new forms, and
- breaking conventional means of design collaboration.

Step 5: The evaluation of the established scenarios in the Teaching Network will:

- Evaluate the BUILD-IT system according to the requirements of the different scenarios - feasibility, use of technology, and team work,
- evaluate the Teaching Network itself, and
- establish documentation on the overall online environment.

Step 6: The refinement will be the final step and will be guided by the results of the evaluation. The tasks in this refinement will be to:

- Optimise the scenarios and teaching methods,
- integrate and adapt the BUILD-IT system more tightly in to the context of learning, and
- improve the overall potential of the Teaching Network.
11 Evaluation of the BUILD-IT System Using The Activity Checklist

Activity theory is a framework helping researchers to describe, understand, and analyse human activity. In this project, we view activity theory as a concept to give structure to and to help improve human work practice. The theory does not provide ready-made solutions, but offers a variety of tools helping us to ask meaningful questions. An overview of such tools for application with information systems was given by Hasan (1999). In her paper, she refers a handful of tools using activity theory to support the design processes in Human-Computer Interaction (HCI). In our view, one of the most successful efforts is seen in an analytical tool called The Activity Checklist (Kaptelinin et al., 1999). The strength of The Checklist lies in its relatedness to problems raised by practitioners in the field. A section of that list, called the Sample Questions (Table I), is intended for the early phases of system design. While we view our research as a considerable endeavour in exploring the relatively new field of Tangible User Interfaces (TUI), we still consider our work to be early steps towards the integration of physical/tangible and digital/virtual realms. From this venture point, we use The Activity Checklist to give meaning and structure to the final system critique of our prototypical work.

The list puts a heavy emphasis on the principle of mediating tools, which is actually in line with the theoretical underpinning we have chosen (Fjeld et al., in press). A tool mediates an activity, thereby connecting a human being, not only to the world of objects - his or her physical surroundings - but also to other human beings. The specific aim of our project was to evaluate different ways to enrich computer-mediated activity with tangible tools from the physical world. Before we discuss our achievements and shortcomings in fulfilling this aim - using The Sample Questions - we proceed with a few definitions. Users of the systems can be described by the following three categories; for each kind of user group appropriate target technology and target actions can be defined:

DEVELOPERS

- **Who are they?** They are system designers, meaning computer scientists, work psychologists, and industrial designers.

- **What is their target technology?** Their target technology is the BUILD-IT system with all advanced features described in this project. Developers have full access to system specifications, to multimedia framework and source code of the system and even to development environments such as debugging and version checking.

- **What are their target actions?** Their target actions are design, development and usability testing of the whole system, including software and hardware components. Mostly single-person use, but multi-person use when this is part of a discussion among developers. Some target actions are specifically part of user-studies and usability experiments.
EXPERT USERS

- **Who are they?** They are users who stay in close contact with the developers, but who are affiliated outside the institution of the developers, such as other research institutes or industry. They have received first-hand training in use of the system.

- **What is their target technology?** Their target technology is the BUILD-IT system with most of the advanced features described in this project (depending on software version and releases). Expert users can modify all parameters of the system as well as menu files which show the contents of the menu and the plan view.

- **What are their target actions?** Their target actions are preparation for end-use sessions by modifying system parameters and menu files. This preparation is aimed at supporting end-users in their target actions. Expert-users perform brick-based use of the system to check end-use functionality.

END-USERS

- **Who are they?** They are architects, designers of production plants, or professionals at a similar level of technical involvement with the system. Also, they are participants in the usability experiments, who fall into this category, because they exclusively use the brick-based interface to the system.

- **What is their target technology?** Their target technology is the BUILD-IT system with a standardised set of functions, delivered to the clients of the system. There are different releases of hardware and software.

- **What are their target actions?** Their target actions are brick-based use of the system to carry out planning and configuration tasks. Single-, but mostly multi-person use.

In the rest of this section, expert users are the default users group if nothing else is mentioned. However, we will also discuss aspects concerning developers or end-users specifically, or combinations of the user groups. That is, only in cases where we discuss topics with wider relevance than expert use, this will be mentioned.

The Sample Questions are meant to analyse how people use, or will use, computer-based technology. In our case, we use it to summarise and predict how well developers, expert-, and end-users can employ the system. The Checklist consists of four sections, which are four different approaches to describe the target technology and how it supports target actions (Kaptelinin et al., 1999, p. 4):

1. "Means and ends - the extend to which the technology facilitates and constrains attaining users’ goals and the impact of the technology on provoking or resolving conflicts between different goals;

2. Social and physical aspects of the environment - integration of target technology with requirements, tools, resources, and social rules of the environment;

3. Learning, cognition, and articulation - internal vs. external components of activity and support of their mutual transformation with target technology;

4. Development - developmental transformation of the above components as a whole."
<table>
<thead>
<tr>
<th>MEANS/ENDS</th>
<th>ENVIRONMENT</th>
<th>LEARNING/COGNITION/ARTICULATION</th>
<th>DEVELOPMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Are all target actions actually supported?</td>
<td>*Are concepts and vocabulary of the system consistent with the concepts and vocabulary of the domain?</td>
<td>*Is the whole &quot;action life-cycle&quot;, from goal setting to the final outcome, taken into account and/or supported?</td>
<td>*What are the consequences of implementation of the target technology on the target actions? Did the expected benefits actually take place?</td>
</tr>
<tr>
<td><strong>Is there any functionality of the system which is not actually used? If yes, which actions were intended to be supported with this functionality? How do users perform these actions?</strong></td>
<td>*Is the target technology considered an important part of work activities?</td>
<td>*Does the system help to avoid unnecessary learning?</td>
<td>*Did the users have enough experience with the system at the time of evaluation?</td>
</tr>
<tr>
<td>Are there actions, other than target actions, which are not supported, while users obviously need such support?</td>
<td>*Are computer resources necessary to produce a certain outcome integrated with each other?</td>
<td>*Is externally distributed knowledge easily accessible when necessary?</td>
<td>*Did the system require large time/effort investment in learning how to use it?</td>
</tr>
<tr>
<td>Are there conflicts between different goals of the user? If yes, what are the current tradeoffs and rules/procedures for resolving the conflicts?</td>
<td>*Is target technology integrated with other tools and materials?</td>
<td>*Does the system provide representations of user’s activities which can help in goal setting and self-evaluation?</td>
<td>*Did the system show increasing or decreasing benefits over the process of its use?</td>
</tr>
<tr>
<td><strong>What are the basic limitations of the current technology?</strong></td>
<td>*Are characteristics of target technology consistent with the nature of the environment (e.g., central office work vs. teleworking)?</td>
<td>*Does the system provide problem representations in case of breakdowns which can be used to find a solution or formulate a request for help?</td>
<td>*Are users’ attitudes towards the system becoming more or less positive?</td>
</tr>
<tr>
<td>Is it necessary for the user to constantly switch between different actions or activities? If yes, are there &quot;emergency exits&quot; which support painless transition between actions/activities, and, if necessary, returning to previous states, actions or activities?</td>
<td>*Are there external representations of the users’ activities which can be used by others as clues for coordinating their activities within the framework of the group or organization?</td>
<td>*Are there negative or positive side-effects associated with the use of the system?</td>
<td></td>
</tr>
</tbody>
</table>

Table I: The Sample Questions taken from The Activity Checklist (Kaptelinin et al., 1999). Sample Questions treated in more detail are marked with two stars (**), those treated in less detail are marked with one star (*).

The authors of The Checklist suggest not to use it in a linear fashion. Rather, "practitioners should look for patterns of related items, even if these items belong to different sections." (Kaptelinin et al., 1999, p. 4). Here, we chose to elaborate on some,
but not all of the questions. In our research, we focused on the single-user aspects of the system, keeping in mind that the system was designed for multi-person use.

As developers, we mainly carried out controlled single-user experiments. Whereas field-studies with multiple users would also have been of interest, it is beyond the scope of this project. With these conditions in mind, we gave different priorities to different questions, leaving out some of them and paying more attention to others. The Sample Questions treated in more detail in our discussion are marked with two stars (**), those treated with less detail, are marked with one star (*) (Table I).

**MEANS / ENDS**

Is there any functionality of the system which is not actually used? If yes, which actions were intended to be supported with this functionality? How do users perform these actions?

**Hardware (HW):**

We refer to two central example from hardware developments, namely, *brick design* and *design of physical height tools*. There were also other hardware aspects of the project, such as the design of the rack with a mirror, the integration of video camera and IR source into one physical unit (called space-observer box), the use of beamers, and the development of a portable system. However, the two examples we mention here came out of design discussions and prototyping that lasted one year and involved such different competencies as work psychology, industrial designers, and computer scientists.

- **Brick design:** (relevant for: developers and end-users) Different brick forms are currently not used, only multiple instances of a rectangular brick shape. The underlying intention by using alternatively shaped bricks, was to offer more diversity in the physical realm. Instead, users employed the diversity offered by the realm of virtual models and tools. One reason for this choice is the outcome of a user study where we examined alternative brick forms and found that a simple, rectangular form was preferred by the participants to more elaborate shapes. Another reason that alternative brick forms are not used, is that the current system does not distinguish bricks tagged with codes or identifiers. Thereby, a consistent binding between brick semantics and system syntax cannot be supported.

- **Design of physical height tools:** (relevant for: developers) The physical height tools are not actually used. These tools were intended to change the height of one or several virtual models. Instead, this is performed using the virtual floor and ceiling. We already described the design process of height tools — where we made design choices between virtual and physical tools.
Software (SW).

Considering the whole project, the major effort in development was invested in software. Therefore, we refer to several examples of software design issues, which involved discussions and prototyping taking place over several months.

- **Locking mechanisms**: (relevant for: developers and end-users) At the moment of selecting a virtual model with a physical brick, a planar relation - in terms of position and rotation - is established between the physical and the virtual worlds. We call this a locking mechanism. A locking mechanism determines how a physical brick and a virtual model stay connected from the moment of selection until the moment of deselection. These different locking mechanisms were intended to be used dependent on the form or character of the virtual model or tool. A particular kind of locking mechanism is based on one particular kind of alignment procedure. For circular and rectangular brick forms appropriate locking mechanisms were foreseen, but never put into use. Instead, one single locking mechanism is used all the time and this has proved to be sufficient for the target actions we addressed in our research.

- **Snapping**: (relevant for: developers and expert users) This is a mechanism which implies that two virtual models connect into one, which has been implemented, but is still not used. The idea behind snapping, was to assign one or several snapping points to each virtual model. When snapping points of two aligning models would come close, the two models would connect. This was meant to enable composite modelling and to make model handling more efficient. Instead of snapping, complex model composition is carried out off-line, while configuring the system.

- **Side view navigation by the EyeCatcher**: (relevant for: developers and end-users) In this navigation mode, full control of the side view is possible, meaning shift, rotation, zoom, tilt, and roll. The meaning for offering all degrees of control, was to enable exploratory navigation of the side view. However, in a pilot study, users reported that they found tilt and roll disturbing, since these two functions modified the horizon. When there was no fixed horizon, users reported that they lost orientation in the virtual scene. Therefore tilt and roll were not offered in later use, and exploration of a virtual scene was sufficiently attained by using shift, rotation, and zoom.

- **Simulation**: (relevant for: developers and expert-users) This function was aiming to animate virtual models being controlled by an external simulation program. The program should send information about models and their modified positions at a given rate. This would allow users to follow a typical production cycle, and even to interact with the simulation software by changing the limiting condition for the simulation and to see how these changes would affect production parameters. This function could not be replaced or emulated in any other way.

- **Communicating with CAD-systems**: (relevant for: developers and expert-users) This function was meant to offer real-time bi-directional communication of virtual models and scenes between the BUILD-IT system and any CAD system. The use of this functionality was never improved and required special competence, not widely established in the development team. Therefore, new virtual models are now...
imported off-line into the system, and model transfer in the other direction is not possible.

- **Database connection**: (relevant for: developers) This involved annotating virtual models with text, like name, price, etc. Finally, due to lack of resources in development, this was not offered. However, the same or similar information is accessible in sources external to the system.

- **Real-time printing**: (relevant for: developers) Printing allows immediate physical versions of planning results, however, this was only partly implemented. Instead, graphical files can be stored and printed after planning session.

**What are the basic limitations of the current technology?**

- Latency in input: Image processing is slow in the detection of bricks.

- Latency in output: Three-dimensional (3D) graphics update of virtual scene is time consuming.

- Top projection: Physical objects and/or hands above the table may obstruct image projection.

- Brick-detection by image processing: Physical objects and/or hands may obstruct brick detection. As a consequence, virtual models are de-selected, even if this is not wanted by the users. Such de-selection may have disturbing consequences: either through lost position and/or accuracy (when the model can be re-selected) or breakdown of a navigation sequence (when a tool disappears at de-selection).

- Portability: The system software requires a specially configured PC with a 3D graphics card and a video frame grabber card. This means that the software cannot run on standard PCs.

- Price: The price for a system is quite high, approx. CHF 50K / EURO 30K. This is gives a single-user licence. Site and/or educational licences were not reported so far.

- End-user programming: This means that end-users, who have not necessarily been taught how to write code in conventional programming languages, write computer programs. Examples include spreadsheet users who write formulas and macros. The source-code for the BUILD-IT system studied was written in C++, later it was ported to Java. This source-code is not open to the end-users of the system and can not be modified by them as they might need. It can only be modified by the company owning the rights to and selling the system, this caused considerable expenses. If the source-code would have been open to end-users, there would still have been a question if this would have been profitable to them, depending on their competence in code-development. Their benefit would also have depended on, and impacted by, the maintainability of the software (SW).

- Maintainability of the software (SW): Whether the code is developed further by the developers, end-users, or both, there is anyhow a need for maintenance. There are
several criteria for software to be maintainable. At least, it should have consistent structure, be modular, well-commented, contain some sort of exception handling, be portable from one compiler to another, and have intelligible names etc. Since the system was developed mainly as a research prototype and not as an industrial product, all these criteria could not be consistently fulfilled. For code production and (re-)use over many years, however, these criteria must be fulfilled. If not, a project is threatened by moral stagnation in programmers, technical shortcomings in development, or exponentially raising costs. With the experience and insight gained as system developers, we rank the following criteria according to their fulfilment (Table II):

<table>
<thead>
<tr>
<th>Criterion for maintainability of the Software (SW)</th>
<th>Fulfilment [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modularity, i.e. a consequent use of functions, procedures, and files</td>
<td>95</td>
</tr>
<tr>
<td>Well-commented</td>
<td>80</td>
</tr>
<tr>
<td>Intelligible and consistent use of names</td>
<td>80</td>
</tr>
<tr>
<td>Explicit exception handling</td>
<td>50</td>
</tr>
<tr>
<td>Portability between different compilers and processors</td>
<td>20</td>
</tr>
<tr>
<td>Transparency to major virtual model formats</td>
<td>20</td>
</tr>
<tr>
<td>Portability between different 3D graphics cards</td>
<td>10</td>
</tr>
<tr>
<td>Portability between different video frame grabber cards</td>
<td>10</td>
</tr>
<tr>
<td>Peer-reviewed</td>
<td>10</td>
</tr>
<tr>
<td>Minimal low-level configuration of the PC required</td>
<td>5</td>
</tr>
</tbody>
</table>

Table II: Criteria for maintainability of the Software (SW) and their fulfilment in percentages.

- Maintainability of the hardware (HW): With the experience and insight gained as system developers, we rank the following criteria according to their fulfilment (Table III):

<table>
<thead>
<tr>
<th>Criterion for maintainability of the Hardware (HW)</th>
<th>Fulfilment [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Installation of beamers, fixation, screws etc.</td>
<td>95</td>
</tr>
<tr>
<td>Reduction of beamer lifetime, caused by non-standard positioning</td>
<td>70</td>
</tr>
<tr>
<td>Adjustment of the video camera lens</td>
<td>80</td>
</tr>
<tr>
<td>Positioning of the video camera box</td>
<td>80</td>
</tr>
<tr>
<td>Robustness of the mirror</td>
<td>80</td>
</tr>
</tbody>
</table>

Table III: Criteria for maintainability of the Hardware (HW) and their fulfilment in percentages.

**Is it necessary to switch between different actions and activities?**

Actions: Yes, in breakdown situations like unwanted de-selection it is necessary.

Activities: Yes, to generate new virtual models outside of the system it is necessary.
If yes, are there "emergency exits" which support painless transition between actions/activities, and, if necessary, returning to previous states, actions or activities?

Manual file storage: This is available, but requires the use of mouse and keyboard.

Automatic file storage: This takes place at regular intervals, given in "etrc.dat" (Appendix 2) by "buildit.MenuUpdateWaitTime" in milliseconds (Appendix 2). Both parts of the menu and the plan view content are updated.

Undo: This does not exist. A prototype was developed, but never but into regular use.

ENVIRONMENT
Are concepts and vocabulary of the system consistent with the concepts and vocabulary of the domain?

This depends on the domain, and here we discuss a few:

Production plant layout: The concept of a three-dimensional (3D) side view is not really used in this kind of planning, but only a 2D plan view, which our system offers at the table. Therefore, the system offers a certain redundancy for expert planners. The concept of using physical bricks, however, corresponds well with physical down-scaled models of robots, welders, and other production components.

Architecture: The concept of a three-dimensional (3D) side view is highly welcome in collaborative work among architects. This effect came out clearly in a design competition referred to in Appendix A. Similar effects were reported by Arias et al. (2000).

Is target technology considered an important part of work activities?

Production plant layout: For many years, CAD is being used in this kind of planning, and has a key role in that activity. However, paper drawings still remain important as well. Nevertheless, working with virtual models is well supported in this field.

Architecture: The same is the case in this domain.

Are computer resources necessary to produce a certain outcome integrated with each other?

First, it is necessary to work with a three-dimensional (3D) modelling tool to generate VRML-models, and the integration with such modelling tools is one still open for improvements to assure ease-of-use.

Second, the system is a pre-CAD system, and offers a bi-directional interface to CAD systems. However, data transfer is better developed for transfer in one direction than in the other. Data transfer from a three-dimensional (3D) CAD system to the BUILD-IT system (i.e. when new models are generated in a CAD system) is currently offered to end-users, although the feature has not yet undergone usability studies. Data transfer
from the BUILD-IT system to CAD (i.e. results of planning activity, consisting of multiple models) has been implemented, but was never offered to the end-users nor tried out under realistic conditions.

**Are characteristics of target technology consistent with the nature of the environment (e.g., central office work vs. teleworking)?**

Central office work: The system may only partly be integrated in central office use. The reasons are that it needs to run on a separate, proprietary computer, where basically no other major applications should be installed or used. The installation of beamers in the BUILD-IT system is partly proprietary. The beamer giving the plan view resides in the system, and can not really be used for any other applications. The beamer giving the side view stands freely in the system, and can easily be moved and be used in other applications.

Teleworking: Due to the fact that only few or no other applications can be installed and run on the same computer, video-conferencing based on the same PC as where the system is running is not possible.

**LEARNING/COGNITION/ARTICULATION:**

*Is the whole "action life-cycle", from goal setting to the final outcome, taken into account and/or supported?*

The use of the system requires instruction. A manual has been written, but the use of this manual as sole instruction to work with the system has not been usability tested so far. Most likely, live instruction is needed. Different modes of work, depending on professional background or proficiency in using the system itself, have not been conceptualised or determined so far.

**Does the system help to avoid unnecessary learning?**

This depends on the users’ qualifications, i.e. is the user an novice or an expert? We chose to view this question in relation to the tools which are meant to be replaced and/or enriched, e.g. CAD-stations. CAD-instruction requires considerable time and requires that users can handle a computer at all. Users of BUILD-IT do not need to know how to use a computer.

**Is externally distributed knowledge easily accessible when necessary?**

Accessibility of externally distributed knowledge (Table IV).

<table>
<thead>
<tr>
<th></th>
<th>Using PC with standard SW</th>
<th>Using BUILD-IT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paper documents, e.g. drawing</td>
<td>Low</td>
<td>High (within plan view)</td>
</tr>
<tr>
<td>Documents as files, e.g. Word</td>
<td>High</td>
<td>Low (requires keyboard)</td>
</tr>
<tr>
<td>Virtual model files, e.g. VRML or 3D</td>
<td>Medium</td>
<td>Medium (if in menu)</td>
</tr>
</tbody>
</table>

Table IV: Accessibility of externally distributed knowledge.
Does the system provide representations of users’ activities which can help in goal setting and self-evaluation?

History: Since many bricks can be used at a time, some may be in actual use, others may hold models or tools not currently used for seconds or minutes. Hence, physical bricks can very well serve as a kind of external memory. This was observed in an experiment with alternative navigation methods. There, some participants used bricks to hold navigation tools available, perform model handling, and then turn back to navigation. Some participants were observed to use one brick to hold the first model found, while looking for the second one. Only after they had found both models, they positioned them correctly, thereby completing the task.

Personal system configuration /etrc-file: These files may hold personal settings like locking mechanism, the content of menus, and the content of the view. While we have observed that these features have been useful in defining goal-setting, we do not know whether they may support self-evaluation.

Saving latest results or intermediate results: The latest results are saved at a regular basis (e.g. every 5 or 10 minutes). In the latest version of the system, programmed in Java, there was a possibility to save the current version at any time. We have not investigated whether this feature was useful in self-evaluation.

Does the system provide problem representations in case of breakdowns which can be used to find a solution or formulate a request for help?

Last version before breakdown: Depends on the regular update, taking place with an interval set in the parameter file (etrc.dat).

Undo: This does not exist.

Exception handling: This is only available for developers.

Intermediate versions: This exists, but requires the use of keyboard access to files.

Debugging: This requires programming-skills.

Are there external representations of the user’s activities which can be used by others as clues for coordinating their activities within the framework of the group or organisation involved?

Files show the positioning of all virtual models. However, such use requires another system to visualise and continue working with the file-content. Paper printouts of the results can be used by others within a group or organisation, but are not always available.

DEVELOPMENT

What are the consequences of implementation of target technology on target actions?
Multi-user usage: Users work together around a table instead of using single computers. We still do not know whether this implies that users more easily or more surely can achieve common concepts.

Single-user usage:

**Did expected benefits actually take place?**

Research benefits: Yes, single-user benefits were observed, however multi-user benefits still have to be quantified and proved.

Commercial benefits: These were only partly fulfilled.

**Did users have enough experience with the system at the time of evaluation?**

Users studies: In these studies, learning was not quantified.

Experiments: Graphical representation of learning in the navigation usability evaluation (Fjeld et al., 2001) (Fig. 1), indicates that some learning took place within the experimental trials, but that this levelled out towards the end of each session / each participant.

**Did the system require large time/effort investment in learning how to use it?**

Developers: The learning of the system in terms of programming and (re-)configuration was a lengthy process, and was only achieved by two to three of the team members. It would have been advantageous if more, e.g. four or five, of the team-members had known how to program, or at least how to (re-)configure the system.

End-users: From usability evaluations, we know that the time needed for end-use of the system is a matter of five to ten minutes.

**Did the system show increasing or decreasing benefits over the process of its use?**

Developers: This was not investigated. However, judging the difficulty in learning how to program the system, it was not surprising that the number of persons able to program and (re-)configure the system remained at a low level (two to three persons, all team members).

End-user: We did not observe end-use of the system over longer time. However, we know that the benefit in the competition among architecture students was not reached in a follow-up project for educational purposes. The reason for the lack of success in the follow-up project, may be that the persons who were able to program and (re-)configure the system, were not longer available to offer support.

**Are users’ attitudes towards the system becoming more or less positive?**

To answer this question, we found it useful to look at the participant statements from the first user study (alternative brick forms) and compare them with those from the latest experiment (alternative navigation tools). In both cases, participants were asked to give
general statements on the use of the system. We observed approximately the same kind and number of positive statements in both cases.

**Are there negative or positive side-effects associated with the use of the system?**

Negative side-effects: *Transportation:* In transportation, the system is bulky. Even the transportable system is bulky, but may be transported by one person between car / train / airplane. It is an open question whether the portable system may sustain to be transported as air-cargo or needs to be taken as hand-luggage. *Use:* In the navigation usability experiment (Fjeld et al., 2001), one participant reported that the use of the system was tiresome.

Positive side-effects: These have been reported as participant statements from the user studies and the experiments and will not be repeated here.
12 Discussion and Conclusion

While the BUILD-IT system has become paralleled by a variety of alternative frameworks for tangible interaction (Ullmer and Ishii, 2000), which part of the knowledge acquired in this project remains valid? That is, which parts of the design insights and empirical results remain valid for future design, use, and evaluation of Tangible User Interfaces (TUI)?

Trying to answer such questions, this chapter will discuss our design activity and empirical research at a higher level of abstraction. First, it suggests conclusions with general validity for TUIs, second, it points out what the next few steps in this field of research may be. Organised along five distinct topics, the discussion and attentive conclusions will be given in the following steps:

Activity Theory: This part discusses the motivation for using activity theory to underpin our research and the outcome of that choice. For each of the following points, we will relate our results to the use of activity theory and make statements on the theory’s predictive power.

Concepts and Terminology: This part reformulates some of the basic assumptions and concepts underlying our research. It also restates some of the key words that have followed us throughout the project.

Human-Computer Interaction (HCI): This part discusses how we designed for effective and efficient interaction and the outcome of usability evaluations.

Computer Supported Cooperative Work (CSCW): This part restates the end-user oriented aims we tried to reach in this project and evaluates to what extent they have been fulfilled. In particular, we look closer at the capacity of tangible groupware systems to foster common concepts in an early phase of team-based planning activities. As the focus of the empirical investigations of this project has been single users studies, we will not report on validated results for multi user processes, but simply indicate what we assume to be general principles.

Technical Aspects and Implementation: This part discusses the technically enabling and limiting factors in terms of hardware and software. In particular, we discuss the importance of designing non-proprietary and portable systems.

Activity Theory

Activity theory is an alternative to existing methodologies used by practitioners in Human-Computer Interaction, "regarding people (computer users, developers, and managers) as subjects of dynamic, purposeful activities that are mediated by various tools (physical, mental and virtual) and peculiar to the context in which they occur" (Hasan, 2000). Activity theory as a framework for our design activity partly gave us concepts and partly guided the steps of our design process. The theory is also the background for a checklist (Kaptelinin et al., 1999), which we successfully employed to evaluate the value of the BUILD-IT system as a tool for collaborative planning.
Concepts and Terminology

From early on in the project, there was a need to make a distinction between the following concepts: real, physical, virtual, digital, tangible, and graspable.

In a panel discussion among several researchers of the Augmented Reality community (chaired by Eisenberg and Mackay, 1996), a variety of distinct visions for integrating "real-world" and computational media were defended. The chair persons defined virtual and being non-physical artefacts. They advocated that "friends meet in virtual 'chat rooms'; children are educated in virtual classrooms; academics attend virtual conferences. The products of computer-based work and education - spreadsheet models, educational simulations, Web-based art galleries - likewise have an almost relentlessly intangible nature." (Eisenberg and Mackay, 1996). Hence, the chair persons oppose the word virtual with the words physical and tangible. At the same time, they talk about the "integration of computers and real-world objects".

In opposition to the view of the chair persons, is a statement by Sheila Lehman: "Now we have virtual reality, the World Wide Web - technologies that challenge us to revise perceptions of physical reality, and relationships to space and place, just as earlier technologies have done. Space is more and more plastic, and it is beginning to be understood as a human construct, in part because we can create virtual environments to suit our needs and adapt ourselves to the new behavioural settings they afford. Howard Rheingold suggests that we need 'virtual communities' because the informal public meeting places which once fostered social life are disappearing from our 'real lives'. When he says that 'real' and 'virtual' spaces have begun to overlap and intertwine, I think he is really suggesting that we need to redefine both." (Eisenberg and Mackay, 1996).

With these opposing understandings of basic vocabulary, we believe there is a need to redefine the concepts of "real" and "virtual". However, we do not see it as the aim of this project to offer those new definitions. Rather, we prefer not to deal with the term "real"\textsuperscript{15} and to understand "virtual" as something opposed to the physical and tangible, like the chair persons did (Eisenberg and Mackay, 1996). Such a choice is backed up by activity theory, where "purposeful activities are mediated by various tools being either physical, mental, or virtual" (Hasan, 2000) and where the term real hardly is used (Nardi, 1996). Since we do not study mental tools in this project, this leaves us with a focus on physical and virtual tools. In our project, interactive functions were offered either through physical tools, through virtual tools, but mostly through a combination of both.

When it comes to the actual modelling of a setting (or: layout) and how to transform this setting through a planning session, we chose to use the following terminology: The setting - which is to be planned or configured - is modelled by a virtual environment (VE), containing virtual models. Not all virtual models of a VE are visible and accessible for user interaction at all times.

\textsuperscript{15} What is perceived by users as real, is treated in more detail at a theoretical level (Fjeld et al., in press, a). However, we do not investigate what computer users may perceive as real and what they may perceive as virtual, since such questions are considered beyond the scope of this project.
Virtual models may take on several functions. One major role of virtual models is to represent static artefacts of the physical world (e.g. furniture, machines, or buildings). Another major function of virtual models, is to serve as tools operating on the scene or on other models. In this project, these two roles were not combined within one virtual model, although that would certainly have been a possibility.

The word object is reserved to describe goal-oriented human activity in the context of activity theory. That is, in the process of reaching the aim of their activity, humans transform an object. Hence, the object of a planning session is the virtual environment. The concepts of object, objectification, and exteriorization are mainly introduced and explained in Fjeld et al. (in press).

**Human-Computer Interaction (HCI)**

A series of HCI topics still remain open in the design of effective interaction with TUIs. To give structure to the discussion of HCI topics, we re-state the guidelines suggested (Fjeld et al., in press) and structure our discussion along the same points.

| 1. Use physical interaction handles as exteriorizations |
| 2. Assure coinciding action and perception spaces (Rauterberg, 1995) |
| 3. Support body motions and simple everyday skills (Campbell, 1988) |
| 4. Respect body-space (Rauterberg, 1999), using less intrusive devices (Vince et al., 1999) |
| 5. Support a clear binding between physical handles and virtual models |
| 6. Draw on bimanual coordination skills |
| 7. Foster exploratory epistemic action by assuring low risk in trying out |
| 8. Support fluent navigation methods to explore 3D virtual worlds |
| 9. Support tactile or haptic feedback (Mackenzie, 1994) |
| 10. Give visual feedback consistent with user expectations |

*Table I*. Set of design guidelines

1. Physical handles may be seen as part of the interface (which is the case for the bricks), but may just as well be seen as an extension of the user’s hand (which would be the case for pen-like input devices). Other devices may combine both, and be part of the world, part of the user, at different moments of use. This dichotomy calls for further investigation, and help may be found, rather in the works of Bateson (1986) and Gibson (1972), than in activity theory, where perceptual matters and the study of tools still shows deficits.

A further aspect of using multiple handles at the table-top, is how to design for efficient multi-user concurrent input.

A final aspect of using physical bricks, is the integration and use of specialised (or: tagged) bricks. Such bricks may be specialised from a machine point-of-view, from a user point-of-view, or both. Each of these three possibilities imply different priority on hardware and software development.

2. In the study of alternative navigation methods, we observed that the plan view (offering coinciding action and perception spaces) made no difference in performance between the two alternative handling methods. For the side view
(offering separate action and perception spaces) there was a difference in performance. This indicates that coinciding action and perception spaces may give more freedom in the design of interaction methods.

3. Even though simple everyday skills are sufficient to operate the system, we observed that the need for learning to operate advanced functions, is not necessarily seen as a problem by the users. Hence, a TUI should offer basic functions which are easy to use, but also advanced functions, requiring some learning to be operated satisfactory. Hence, the potential tradeoff between ease-of-use and learning-by-doing is a relevant research aspect.

At this stage, we find it worth to point out that the design for different kinds and levels of users' skills is worth of further exploration. This is not only a question of proficiency, but just as much a matter of differences in personal interaction style.

4. The system does respect body-space and uses devices, bricks, which are not intrusive. The physical height-tools which we investigated, however, were partly perceived as intrusive, by obstructing users' free hand movements across the table. This may have been one reason that they were not well received by users. Another, partly intrusive aspect, is the image based tracking of the bricks. Such tracking is not suited to let users move their hands freely across the table. One solution may be found in electromagnetic brick tracking from below, as suggested by Patten et al. (2000).

5. Different locking mechanisms between the bricks and the virtual models was implemented, but never explored in empirical studies. However, we believe that the problems of accuracy and latency in connecting physical and virtual will have to be solved prior to the choice of locking mechanism.

6. The efficient use of bimanual input, i.e. two-handed input, remains a research topic (Guiard, 1987), although it could not explain the differences in performance we observed in our usability evaluation.

7. The implication for epistemic action on performance could not be empirically proved. However, it remains a research topic to find the benefits of designing for exploratory interaction (Kirsh et al., 1994).

8. Fluid navigation is tightly related to the update mechanism, being either continuous or discrete. The central research question is, when is continuous and when is discrete update appropriate? So far we found that for the plan view, fluid navigation is less critical. For the side view, however, fluid navigation has a decisive impact on task solving performance.

9. The use of tactile, or haptic, feedback, calls for the use of propelled bricks (Fitzmaurice, 1997). The research question coming out of this observation is; how to incorporate and take effective use of bricks, not only as input devices and as handles to virtual models, but also as output devices?

10. In our studies, we have only investigated user expectations in the final part of our usability evaluations, and only with the intention to explain differences in performance and use. However, the information at hand, may just as well be used to redesign the system to be more in accordance with users' expectations, in particular, regarding locking between physical bricks and virtual models and regarding rotational clues offered by the bricks.
Other major points related to HCI are:

- The outcome of task analysis, as carried out in forefront of our design activity.
- The distinction between activity, action, and operations, and related to these three entities, the analysis of breakdown situations.

**Technical Aspects and Implementation**

There is a series of technical matters still to be addressed in the design of TUIs. These can be listed as follows:

- Taking account of standard usability questions in the development process. That is, assure that standards functions offered in most graphical applications are available, such as: open, save, save as, quit, exit, copy, delete, and paste. In this project, very specific technical questions such as computer vision, graphical display, and physical-virtual binding were given far more attention than assuring the standard functions mentioned here. This choice may offer a way to understand the challenges met in continuing the development of the BUILD-IT system and turning it into a commercial product.
- Assure compatibility with VRML-standards and CAD-standards.
- Apply standard software quality measures to the use and integration of a multimedia framework (Ackermann, 1996).
- Assure modular coding of the application.
- Design for fluid interaction and navigation. This depends on image based tracking and graphical update latency (Fjeld et al., 2001).
- Choose an appropriate tracking of the bricks, either using computer vision (from the top) or using electromagnetic techniques (from the bottom). In their state-of-the-art system called *Sensetable*, Patten et al. (2001) present a solution that "electromagnetically tracks the positions and orientations of multiple wireless objects on a tabletop display surface". They claim that their system offers two types of improvements over existing tracking approaches such as computer vision. First, "the system tracks objects quickly and accurately without susceptibility to occlusion or changes in lighting conditions". Second, "the tracked objects have state that can be modified by attaching physical dials and modifiers". Hence, the system can detect these changes in real-time. *Sensetable* has been put into practical use in two fields of applications: chemistry and system dynamics simulation.
- The use of non-appropriate computer / PC.
- The use of generic and/or tagged specialised (Underkofler and Ishii, 1998) bricks.
- The use of propelled bricks (Fitzmaurice, 1996).
- The design of a portable system.

**Computer Supported Cooperative Work (CSCW)**

CSCW is a generic term which combines "the understanding of the way people work in groups with the enabling technologies of computer networking, and associated hardware, software, services and techniques" (Haake and Wilson, 1992). In the future,
we will try to draw second order knowledge from our design experience to make general suggestion for computer mediated collaborative work.
PART IV:

APPENDIXES,
GLOSSARY,
AND
REFERENCES
Appendix A: System Delivery, Assembly, and Use

System Delivery

Figure 1: The system is delivered in a rack and consists of the parts listed in Table 1. The rack is covered by a sheet.

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
<th>#</th>
</tr>
</thead>
<tbody>
<tr>
<td>PC</td>
<td>Desktop, ASUS Mainboard P2L97-S ATX, Pentium II 300 MHz, 64</td>
<td>1</td>
</tr>
<tr>
<td>Sony VPLX600E</td>
<td>High resolution LCD Projector, 1024*768 points, 600 ANSI Lumen</td>
<td>2</td>
</tr>
<tr>
<td>Camera</td>
<td>IR-sensitive camera, cable, and lens</td>
<td>1</td>
</tr>
<tr>
<td>BUILDIT®-Box</td>
<td>BUILDIT® Visualisation box</td>
<td>1</td>
</tr>
<tr>
<td>Brick</td>
<td>Passive interaction handler</td>
<td>2</td>
</tr>
<tr>
<td>SW BUILD-IT®</td>
<td>Single User Software License BUILD-IT®</td>
<td>1</td>
</tr>
<tr>
<td>Papers</td>
<td>1 Box containing papers and a Video</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 1: Packing list.

System Assembly

Figure 2: There is already a beamer and a camera mounted in the rack. The camera points backwards. Turn around the board with the camera so that camera points forwards. Then, place the second beamer, sitting on a board as well, next to the camera. Place the computer in upright position into the lower part of the rack.
Installation, Calibration, and Start-Up

Calibration is needed after system assembly or at too low accuracy between a brick and the bounding box of the selected virtual model.
Double-click install.bat (see: Appendix 1: Directories and files). You see the image processing window in the area projected on the table. The video camera detects bricks as bright areas. Place two bricks in opposite corners (Fig. 6, left) and make sure that these bricks are detected. To get a sharp picture, you normally need to adjust the video camera. To stop the operation, click the X in the upper right window corner.

Figure 6: Place two bricks in image corners (left). Calibration by putting brick at a reference section displayed in the corners of the plan view (right).

Now double-click calibrate.bat (see: Appendix 1: Directories and files). As soon as a brick is introduced to the area projected on the table, an orange virtual model appears (Fig. 6, right). Place the brick at that orange model, and track it when it moves to the following three positions. When the orange model is gone, stop this operation by clicking X in the upper right window corner.

You can now work with the system. To start the system, double-click buildit.exe (see: Appendix 1: Directories and files). To stop the application, click the X in the upper right window corner.

**Basic Model Handling, Create, Delete, Save And Print**

BUILD-IT is a planning tool based on state-of-the-art computer vision technology. The tool enables its users to cooperate in a virtual planning process for a real-world setting, such as a room, a school, a factory or a piazza. Grouped around a table and using real bricks, users can select and manipulate virtual models within the setting which they are planning (Fig. 7).
At all times the users have two up-to-date views of the setting they are creating and manipulating: the plan view and the side view. The plan view is the bird’s eye view from above - which is projected onto the table. The side view is projected onto a wall near the table. In the case of the side view a virtual camera, which can be either outside or inside the plan view, allows the users to choose from which position the side view is to be projected. The side view can also be zoomed. In the case of the plan view the entire projection of the setting can be shifted from side to side, rotated or zoomed. The plan view also contains a virtual storage space (or: menu) for models not in immediate use (Fig. 8). It allows users to create multiple model instances. For all handling operations affecting virtual models users draw on basic, everyday manual skills - selecting, placing, rotating, re-positioning and fixing (Fig. 9). Models brought back to the storage space are deleted (Fig. 10).
Figure 9: The basic steps for user manipulations with the interaction handler (brick).

Figure 10: Removing a virtual model by moving it back into one of the menus.

The contents of the view and the menus can be saved and printed (Fig. 11). Printing only takes place if `buildit.EnablePrinting(Bool)` of the ETRC file (Appendix 2) has been set. Save is performed at the same time as Print is selected. Also, save of the menus and views is performed periodically, with a time lap given in the ETRC file (Appendix 2). This time lap is called `MenuUpdateWaitTime` and given in milliseconds.

Figure 11: Save and Print of the views.

Some of the standard tools are illustrated in the virtual storage space (Figure 12).

Figure 12: Some of the tools and their position in the virtual storage space.
Constructing Single Virtual Models

**Inline**

Most virtual models are Inline. They are based on Virtual Reality Modelling Language 1 (VRML 1) data files, that are converted to *.3D file. For more information, see ‘Input Of 3d-Cad Models: Vrml Data’ and Appendix 4, Installing a new project (see later in this section)

**TexturedBox**

All other virtual models are of a type called "textured box". This is a cube covered by 1, 2, 3 or 5 different pix maps. The number of pixmaps is given by the number of gif files following the textured box declaration in the menu or view file.

If 1, displayed on top of virtual model.
If 2, first on top of virtual model, second on two opposite sides - mirrored on one of them.
If 3, first on top of virtual model, second and third each on two opposite sides - mirrored on one of them.
If 5, first on top of virtual model, the following ones for each side.

**Wall**

A wall model has the quality of being stretchable. Two or more walls may be connected by their respective edges. The lower part of a wall is limited by the floor, the upper part by the ceiling. This feature requires further tuning and refinement.

**Way**

Way is a carpet that is used to define a certain area. It is also an model, so that other models positioned onto it, will move along when the carpet is moved.

**TextObject**

A model with a text attribute, used to assign text to another model. It is associated to the other model by using stacking and fix.

**Colour**

Colour is a property of any Inline model. Only one colour can be assigned.

**Model Handling**

**Rotation**

Rotation is performed by selecting the first (or the second) rotation tools (Fig. 13). The tool is put onto the model to be rotated and left there. The model rotates around its own x (or y) axis. The sense of rotation is given by the orientation of the brick in relation to
the axis of rotation. Rotation stops when tool is removed from model, and the tool disappears when deselected.

Figure 13: Rotation tools as they appear in the menu.

Scaling

Scaling is performed by selecting the scaling tools (Fig. 14). It is put on the preferred model and drawn away (or towards) the centre of that model, making it bigger (or smaller). Scaling along the x or y axis of the model is independent, so that the model may loose its original proportions. The scaling tool disappears when deselected.

Figure 14: Scaling tool as it appears in the menu.

Handling Relation Between Virtual Models

The two first features, stacking and fixation, can be used while working with the system and require no preparation. Snapping and layers each require a file to be prepared in advanced.

Stacking

When one model is dragged over another one, the selected model jumps onto the other one. This feature can be used repeatedly, so that many models can be stacked on top of each other.

Fixation

This tool offers normal fix and elastic fix. Both are handled with the model fix (Fig. 15). When this model is rotated, it turns green, meaning release. Turning back makes it red, and thereby fixed, again.

Figure 15: Fix as it appears in the menu. It may be red (as shown here) or green, depending on the model being fixed or not.

Normal fix means that one model is fixed to the model right below. If no model is situated right below, the model is fixed in the environment. This is achieved by selecting fix in the menu and positioning it the wanted model, giving a beep. Turning fix releases the model. This is signalled by a beep and fix is turning green. When fix is deselected it disappears.
Elastic fix is valid when one model sits on a second one. When fixed and then moved away, they are connected. This means that when one model is moved, the second follows.

**Navigation**

The navigation tools are also offered in the virtual storage space. They are: GroundCatcher, Camera, FrameCatcher, ViewFrame (Fig. 16, l) and the EyeCatcher (Fig. 16, r). For further description, please consult separate section later in the book.

![Fig. 16: Four of the five navigation tools: From close to far: GroundCatcher, Camera, FrameCatcher, and ViewFrame (l.). EyeCatcher is offered separately in combination with the Camera (r.).](image)

**Building Composite Models: Snapping**

See Appendix 3 for detailed description of the file defining snapping relations (snapinfo.dat). buildit.SnapInfoFile and buildit.AngularSnapRange (default 10) are set in the ETRC file.

**Layers**

1. **Assigning models to layers**

Any model can be assigned to a layer with the command

```
layer name
```

where `name` is the name of the layer

This command can be placed in any of the files `Menu_left.dat`, `Menu_right.dat`, or `View.dat` after the model specification of the concerned model and before the next model specification

*e.g.* `layer furniture`

Models remain in the same layer when moved from the storage area to the working area.

All the visibility and mobility of all models in a layer depending on the presence and selection of Layer Activator models

2. **Activating layers**

Any model can act as a layer activator with the command

```
layerActivator name causeLocation cause effectLocation effect
```

- `name` is the name of the layer
- `causeLocation` in `{CauseInMenu, CauseInPlanView}` specifies the location where an action of the layer activator shows an effect
- `cause` in `{SelectDeselect, CreateDelete}` specifies the type of action for which the layer activator shows an effect
• **effectLocation** in \{EffectInMenu, EffectInPlanView\} specifies the location where the effect occurs

• **effect** in \{MakeVisibile, MakeMobile, MakeReplicable\} what effect the LayerActivator produces

An number of layerActivator commands can be placed in any of the files Menu_left.dat, Menu_right.dat, or View.dat after the model specification of the concerned model and before the next model specification. All layerActivator commands belonging to a single model, however, must use the same layer name.

All mobile models are also visible. If menu or plan view have neither a MakeVisibile nor a MakeMobile **layerActivator** then all models are mobile

Example in Menu_right.dat:
```
Inline 1
./projects\house\objects\Sofa.3d
rotposition 0.000000 0.000000 0.000000
scale 1.000000 1.000000 1.000000
rotation 0.000000 0.000000 0.000000
position 0.679319 -2.432012 0.000187
LayerActivator furniture CauselnMenu SelectDeselect EffectlnMenu MakeMobile
LayerActivator furniture CauselnMenu SelectDeselect EffectlnPlanView MakeVisible
LayerActivator furniture CauselnMenu CreateDelete EffectlnPlanView MakeMobile
```

**Input Of 3D-CAD Models: VRML Data**

The application is designed to support providers of assembly lines and plants in the early design pro-processes. However, it can easily be prepared to support a range of other applications, such as interior architecture, city and urban planning. Graphical display is based on the class library MET++ (Ackermann, 1996). The system can read and render arbitrary virtual 3D models. These models are sent from a Computer-Aided Design (CAD) system to BUILD-IT using Virtual Reality Modelling Language (VRML). The system has been engineered to send and receive numerous forms of metadata.

The BUILD-IT system understands 3D-CAD models on the VRML format. VRML data describe the complete geometry and visual characteristics of a model. Data exchange between a 3D-CAD system and BUILD-IT is handled by the CAD-connection.

The connection between a 3D-CAD system and BUILD-IT is called CAD-connection (Figure 17). CAD users are presented with a list of all available models and can select the geometric data required for their specific planning session. The selected geometric data...
data is converted to VRML format and offered by the CAD system as ".wrl" files. Using the CAD-connection, the selected geometric data is then sent as ".wrl" files to BUILD-IT.

A VRML based connection offers the important advantage of data compression, allowing for reduced information flow and less model complexity. This feature is just as vital to model handling in the Web as with the BUILD-IT system. Without data reduction, only high performance CAD systems would be able to deliver multiple 3D model within acceptable time.

For the following CAD representations, conversion to VRML-DATA is supported:
1. CATIA native (Version 3.x-4.1.7)
2. Unigraphics (Version 11.1 - 13.0)
3. Auto CAD (Version 13.0)
4. STEP PART 203
5. IGES 4.0 (depends on the CAD system)
6. MINICAD Macintosh
7. LOGOCAD

Fig. 1: BUILD-IT may communication with a wide range of CAD systems.
Appendix B: Software Aspects

Although software (SW) can be highly complex, it is mostly invisible. There is hardly any relation between the SW complexity and the end-user complexity of a product. For instance, a telephone is simple to operate, but there are thousands of line of codes behind it.

Even industries that are hardware (HW) oriented (cars, aeroplanes), invest most of their research and development in SW. Therefore, the structure, portability, transparency, modularity and maintainability are decisive for the life-time and growth/decay of a SW program.

BUILD-IT and software

The enclosed SW (Appendix 6) concerns a small part of BUILD-IT, namely the GroundCatcher function. The following case should illustrate some relevant questions that arise when working with SW-development:

One or a few people will take over the maintenance and development of the BUILD-IT SW. It is not known where and when that will take place. Measures required for SW-documentation are:

- Naming strategy (classes, types, variables, constants),
- Commenting code,
- Choice and use of algorithms, and
- Data and control flow diagrams

How much effort (in hours) will it take to implement the required documentation? Is it worth it?

If the person who developed the code wants to extend the SW one year later, what would be the answers to the same questions?

One cannot make assumptions about the programming skills of future developers. Novices won’t be able to use badly documented code, experts won’t touch it, but start from scratch.

From SGI to PC

The first prototype was implemented on a SGI O2, since this system offered image acquisition as well as 3D computer graphics, thus making a short development time possible. This system, however, only supported a single monitor interface, leading to a number of technical restrictions and compromises for the table and side views. Additionally, user studies revealed that most industrial companies did not have sufficient know how to support a UNIX workstation and therefore strongly preferred a PC version. Therefore, the complete software was ported to PC under Windows NT. The porting was started at the beginning of 1998 when a fast and inexpensive frame
grabber board as well as a dual-monitor graphics card supporting Open GL became available. It included rewriting the software for a true two-monitors solution, rewriting the image processing library, and a number of system specific adaptations concerning the BUILD-IT software as well as the graphics library MET++. This lead to a cheaper system with clean table and side views and a noticeable increase in speed. Additionally, the system can now profit from the fast innovation cycles of graphics cards and CPS on the PC.

**From C++ to Java 3D programming language**
The first prototype was implemented using C++ programming language. Towards the end of the project, the whole code was ported to Java 3D. The expectations were to make the system independent of MET++ and to make importation of different versions of Virtual Reality Modelling Language (VRML) data more fluid. These aims could be reached, however, the execution of the code took considerably more time. This loss in performance has later been documented in a representative study, where the three programming languages Java, C++, and Lisp/Scheme were compared (Gatt, 2000). The outcome of this study was that the Lisp/Scheme was, by far, superior to C++ and Java. Second, it was shown that C++ was clearly superior to Java.

**Modular Software**
The software is designed in a modular way. New object types with specific individual behaviour or interaction among objects with different types can be implemented quickly. Based on user requests and user studies a number of basic functionalities were implemented: stacking of objects with automatic grouping of stacked objects leads to intuitive manipulation of object groups. Predifinable snapping relations among objects lead to fast and exact object placement. Objects can be assigned to specific layers, allowing for a quick change of the visibility and mobility of groups of similar objects.

**CAD-connection**
A number of tools let the users accomplish their needs quickly and intuitively. CAD interfaces from a large number of 3D CAD systems were realized. The conversion based on VRML, is a widely supported standard for 3D models. Examples of supported systems and file formats are:

**Multibrick interaction**
The original prototype supported tracking of a single brick only. This was extended to multiple brick tracking, leading to a real multi-user systems and leading to new interface concepts based on two-handed interaction. The current version supports up to 30 bricks. BUILD-IT supports parallel interaction.
Appendix C: Collaborative Design with New Interface Technologies: a Competition

Team ETH Zurich: Maia Engeli and Kerstin Höger, Gudela Grote, Kristina Lauche, Fabian Seckler, Martin Bichsel, Morten Fjeld.

Keywords: Haptic Spatial Input, Intuitive Interaction, Interaction Paradigms, Virtual Architecture, Design Process, Co-located Multi-User Interaction

Jury: The jury consisted of two academics from the field of architecture, one from a company offering advanced office furniture, and one from a company in 3D graphical visualisation.

Participant Groups: There were five teams of graduate students in architecture, three consisting of three students, two consisting of four students.

Sponsors: Birkhäuser Publishers, Basle, Boston, Berlin; USM Modular Furniture, Switzerland; Tellware, Zurich.

Collaborative Design: Exploring New Interface Technologies
Gathered around a table, on which a computer-simulated environment is projected, a group of people is engaged in a lively conversation. By means of physical devices, called bricks, they interact with design elements in a three-dimensional virtual scene. In a flurry of activity, they build spatial compositions and imaginary landscapes. The group is taking part in a competition set up to test BUILD-IT, an innovative interface system, as a collaborative design tool. BUILD-IT realises a highly intuitive Human-Machine Interaction, enriching natural actions and communication with virtual features. Tangible objects, shaped as rectangular bricks, serve as the interaction-handles and mediate between the physical and virtual realm. By manipulating both a two- and a three-dimensional projection, team members can visualise and test their design ideas. The externalisation of individual thoughts through the BUILD-IT interface makes them accessible to others and thus allows for mutual understanding in the collective design process. Thanks to BUILD-IT, it is possible to exchange ideas and to create virtual three-dimensional compositions in a spontaneous yet direct collaborative manner.

From Indirect to Direct Interaction: Beyond Keyboard and Mouse
We have observe how the functionality of computer programs increases, but for interacting with the computer, we still use keyboard and mouse. By using physical bricks as interaction-handles, BUILD-IT replaces and extends the function of traditional input devices. It represents a significantly easier and more intuitive way of interacting than the mouse-keyboard-screen interface.

The BUILD-IT system is the result of an interdisciplinary research project, integrating the expertise of people from several institutes of the ETH Zurich with the know-how and support of industrial partners. It was developed with the idea of overcoming the communication constraints of current CAD applications and allowing for cooperation
and communication among stakeholders in a design and planning task. These stakeholders can either be a team of experts or a mixed team of experts and clients. The BUILD-IT competition, which took place in spring 2000 at the ETH Zurich, served to evaluate the feasibility of using the system in the early stages of a design process and to search for new fields of use.

The System: Coincidence of Action and Perception

By employing up to 30 tangible physical bricks, which are recognised by an infrared-sensible camera through its retro-reflective surface, Built-It enables its users to select and manipulate virtual elements. They can place, scale, rotate, fix and reposition them within the scene which they are designing. The users have two up-to-date and simultaneous views of their scene: a plan view projected onto the table and a perspective view projected onto a screen near the table. A virtual camera allows the users to choose the position from which the views are to be displayed. Next to the common workspace, two virtual libraries are projected: one for choosing the composition elements and one for selecting the navigation and manipulation tools.

With this system, the space of the user-interaction coincides with the space of the graphical display. This means the users live and act in the same space in which they receive visual feedback. The users get the impression that they touch and manipulate the virtual models directly. Thus, suitable positions for the elements within the created scene can be found and conflicts with other contributions can be recognised and solved.

The participants of a BUILD-IT session can engage and intervene in the scene at any time. Each change of the scene is immediately visible. In a highly interactive process, many design alternatives can be explored, analysed, and discussed. The design process is enriched through the wealth of ideas and can eventually cultivate the best solution from the field of possibilities and constraints.

Procedure: Constructing models (step 1) and Composition (Steps 2)

Step 1 - Constructing 3D virtual models: Each group had to design three VRLM models for use in the BUILD-IT system. Enrolment for the competition was accepted at the moment these three models were received by the arranging team. Since these models will be used in a composition, they have to be well prepared for building of complex models. This means, that must be able to align to other models, they have to make sense, and they must be easily manipulated in the BUILD-IT system. The following specific rules apply to each of the Virtual Reality Modeling Language (VRML) models:
- All static 3D geometries for VRML allowed.
- Since the models must be static, no textures, light, animation, and sensors allowed.
- Must no exceed 200 Kbytes and have to be in VRML1 Format.
- Unit is meter.
- Must be successfully tested with the BUILD-IT system before the next step.

Step 2 - Composition: The groups met for about three hours in the VISDOME hall of ETH where they worked with the BUILD-IT system to create a composition. There was no prescribed program and the virtual models of all groups were at the disposal of each
group. First, each group received approx. half an hour of instructions, followed by two hours for task completion.

**The Competition Task: Creating Compositions**

The two-phase competition was directed at architecture students. In the first phase, the participants constructed three-dimensional virtual models for BUILD-IT in the VRML format. To model them, they could use any conventional CAD program outside of the BUILD-IT system. The task was to design objects that could become components of an abstract composition to be assembled, stacked, and transformed in interesting ways, to be used for manifold purposes, or to convey a specific quality that would make sense within a composition. In the second phase, the participants formed teams of three to five people and together created the composition during a two-hour BUILD-IT session. For their designs, the teams could use all the elements submitted by the participants, which were made accessible through the virtual library of the BUILD-IT system. The competition was set up as a pure design and composition task. The teams were asked to work out their own theme and to elaborate it within their composition.

**The Design Process: Generating Coherence**

To evaluate the use of a common design platform, the actions and interactions of the participants were observed and recorded during the BUILD-IT sessions. At the beginning of the sessions, each team got a short introduction to the system. All participants learned to handle the interface within minutes. The easy, enjoyable and intuitive usage of BUILD-IT stimulated the team members to work closely together and thereby enhance their mutual exchange of creative ideas and knowledge. Most teams started by exploring the system and the objects submitted by the other groups. While adapting to the system, they sketched and negotiated potential design ideas in a playful and exploratory manner. Towards the end, the teams worked in a more goal-oriented and structured fashion, trying to realise their ideas and to customise the system to their specific needs.

The challenge of the design process was to integrate the diverse, often conflicting, attitudes and perspectives of the team members into a coherent composition. These negotiations, as well as the sharing of resources, revealed interference and conflicts in the design process which had to be resolved and turned into a common design concept and work strategy. BUILD-IT served as mediator between the motives of the individual team members and the scene of their common creation. Through the graspable interface and the unifying workspace, a collective learning and design process was triggered. The sessions demonstrated that BUILD-IT demands dialogue among the members of a team and thus facilitates the emergence of coherence from a multiplicity of attitudes and contributions.

**The Interacting Jury: Clarity versus Consensus**

The jury committee consisted of four members whose expertise spanned the fields of architecture, design, multimedia, and interactive visual computing. The jury members, like the participants, gathered around the BUILD-IT table, interacting simultaneously which each other and the works they were judging. While navigating through and viewing the submitted compositions in many ways, the jury acquired an understanding of the possibilities and constraints of the BUILD-IT system and could thus differentiate
the criteria for judging the works. All results were compared on the level of abstract composition. The implementation or development of design principles, their readability and interplay in the overall composition as well as their adequacy in regard to the requirements of the BUILD-IT system were decisive in the ranking.

The jury categorised the results in two groups, according to whether the composition focused on only one or two elements or covered the full range. The works based on a small number of elements were more coherent in nature, often resulting from the dominance of one team member. The teams that combined various elements were in general more playful and consensus driven. Here, the difficulty was to find a coherent design strategy. The winning design convinced the jury through its complex and open-ended interlocking concept which allowed the integration of multiple design ideas and was enlivened by the variety of elements designed by the competition participants.

**Conclusions: New Forms of Creative Collaboration**

The BUILD-IT competition provided an opportunity to rethink the ways in which teams have traditionally collaborated. By allowing participants to transform and build on each others insights and contributions, it explored new mediations to augment reality for group negotiation and fostered collective understanding in the design process. As in a game, the participants created their own language for communication. Its rules were defined by the team players, who were interacting simultaneously with each other and the environment that they were building. By establishing this level playing field, the competition broke down the hierarchical or manipulative tendencies of conventional means of design collaboration and led to the conception of new forms in which cohesion and consensus can be generated. The adjacent study investigated how BUILD-IT can enhance group design and in return how the demands of collaborative design help to shape the requirements of these new means. For us, the challenge is to bring computer-mediated design closer to the physical realm, a simultaneous habitation of two worlds, each taking advantage of its own medium and its connections to the other. BUILD-IT was implemented to support co-located interaction. The next steps are to enable the sharing and shaping of multiple views across time and space. In the near future, BUILD-IT could simultaneously be used as a co-located, multi-user interface and as a distributed, multi-site workspace that fosters a society of ideas all gaining strength from other ideas.

This composition took first place because of its synergistic manifestation. It exploits the competition task by using a container as a flexible frame in which manifold dreamlike volumes float and interlock with each other. Composed from the variety of elements designed by the competition participants, its spaces offer prospects and surprises which are constantly changing and unexpected and encourage us to wander through them.

This design was chosen for its reduction in the use of one curved surface element. The composition underpins the quality of redundant architecture and refers poetically to its title. Its delaminated properties allow for endless layers of perception and appearance. Light and shadow turn into design elements and form the topography.
Figure 1: First Prize: Dreambox by Olli Bertram, Martina von Tippelskirch, Thomas von Pufendorf, Odilo Schoch.

Figure 2: Second Prize: Delaminated Field by Marcelyn Gow, Andre Rethmeier, Benedicte Vanwanseele.

Figure 3: ScarpaLynn Subway Entrance: Michael Fox, Pau Sola-Morales, Miriam Zehnder.

The two compositions Flux by Ulrike Bahr, Remo Aslak Burkhard, Dimitri Kaden and Turbulence by Andrew Vande Moere, Maria Papanikolaou, Malgorzata Miskiewicz-Bugajski are both not currently documented by photos.
Appendix D: Other Augmented Reality (AR) Research Platforms

DigitalDesk

The DigitalDesk is a development of Wellner (1993), characterized by a projector and a camera sitting above the desk (Fig. 1). In this system, users can write on a paper document and see a projected image of the same document elsewhere on the desk. The projected document may also come from a user of a second DigitalDesk system. Hence, it is possible to work together on a shared information, merging physical and a virtual documents. Stand-alone use of a DigitalDesk is also possible.

By augmenting users’ every-day surrounding and allowing interaction through familiar elements, this was one of the first Augmented Reality (AR) systems. Data input and output is taken care of by the hardware layer, consisting of a combined projector-camera unit (Fig. 1).

![DigitalDesk prototype (Fitzmaurice, 1996).](image)

In Wellner’s groundbreaking work (1993), three further applications supporting human activities were described:

- **Calculator**: Instead of typing digits and numbers with a conventional typing interface, the user points to numbers on real sheets and performs calculations.
- **PaperPaint**: This application allows its users to merge virtual drawings with real drawings on a sheet. Hence, real drawing can be “selected and pasted”.
- **DoubleDigital**: This is an application based on the features of DigitalDesk and allows collaborative work by merging virtual and real elements.

MetaDESK

The metaDESK (Ishii and Ullmer, 1997) is based on the user interface concept called Tangible User Interface (TUI), being part of MIT Media Lab’s vision of Tangible Bits. A TUI is a physical instance of an element used in a Graphical User Interface (GUI), such as window, icon, and menu (Fig. 2). Hence, a TUI allows users to grasp and manipulate digital information in the users’ attention by coupling the digital information with everyday physical objects and architectural surfaces (Fig. 3).

![MetaDESK diagram](image)

In a conventional computer - using screen, mouse, and keyboard - there is a lack of direct physical access to the Graphical User Interface (GUI) elements. Designers have
tried to reduce this distance by implementing metaphors such as the desktop, but still a real desktop and a virtual desktop have few physical properties in common.

Hence, the Media Lab has developed AR systems like the ambientROOM and the metaDESK (fig. 4) in order to reduce the distance between the virtual world and the real world by adding a physical dimension to the GUI elements. The key concepts behind the metaDESK are:

- Interactive Surfaces: Transformation of each surface within an architectural space (e.g. walls, desktops, ceilings, doors, windows) into an active interface between the physical and virtual worlds.
- Coupling of Bits and Atoms: Seamless coupling of everyday graspable objects (e.g. cards, books, models) with the digital information that pertain to them.

This system represents the key ideas of AR: augmenting the user's surroundings with user interfaces, offering familiar interface elements as in a GUI. Thus, they developed graspable windows, icons and the more. They have taken conventional instruments' and passive objects' functionality (such as handles) and enhanced their functionality. This was inspired by the work of Fitzmaurice et al. (1995). These instruments and passive objects -for example phandles (acronym of "physical handles")- allow the user to give the system an input, the same way the user has given an input through keyboard and mouse. Hence, the passive objects and instruments mainly cover the physical aspects of the metaDESK, while the TUI elements, such as graspable windows, cover the handling aspects of the metaDESK.
The metaDESK system’s interface elements consist of lenses, phicons (acronym of "physical icons"), trays, phandles, and instruments. In Fig. 3 most of these elements are visible. The metaDESK also consists of the desk, a back-projected graphical surface, the active lens, an arm-mounted flat panel display, and the passive lens, an optically transparent surface through which images from the desk can be projected. The position and angle of rotation of the metaDESK’s elements are optically, electromagnetically and mechanically sensed.

The MIT Media Lab describe a prototype application for the metaDESK named "Tangible Geospace". In this chapter, only the most relevant features of the metaDESK and "Tangible Geospace" are mentioned. A phicon of the metaDESK can be the MIT Great Dome (Fig. 4).

As soon as the user places the MIT Great Dome phicon on the desk, the computer reads its position and rotation, computes and projects the two-dimensional map of the MIT campus on the desk, making the position, size and angle of rotation of the MIT Great Dome on the two-dimensional map fit the position, size and angle of rotation of the MIT Great Dome phicon. Thus, the MIT campus two-dimensional map can be manipulated by moving and rotating the MIT Great Dome phicon. The active lens and the passive lens allow users respectively to navigate through the two-dimensional map with a perspective view (Fig. 5) and with a magnified two-dimensional view respectively. It has to be highlighted that manipulating the two-dimensional map with the MIT Great Dome phicon (in the case of the Tangible Geospace application) influences the views through the active and passive lenses.

![Figure 4: The MIT Great Dome phicon during map manipulation (l.), the activeLENS (m.), and the rotation constraint instrument (r.).](image)

Manipulation of the two-dimensional map on the desk is also possible with the rotation constraint instrument (Fig. 4). By placing the rotation constraint instrument, which consists of two cylinders mechanically coupled by a sliding bar, on the desk, the two-dimensional map of the MIT campus can be rotated and scaled.

It is worth pointing out that although i) rotation is already available through phicons’ angle of rotation and ii) scaling could be possible using two-phicon interaction, a special scaling instruments was implemented. The reason is to avoid ambiguity that can arise when two phicons are placed on the desk. A situation of ambiguity is created as follows: since every phicon carries plan coordinate data and z-axis angle of rotation data with it, if two phicons are placed on the desk, the system will receive ambiguous user input. Scaling in order to make the phicon positions fit to the two-dimensional map is possible,
but thus the angle of rotation of the two-dimensional map is already determined. Several questions can be asked based on this result: Shall the metaDESK ignore the angle of rotation data given by the phicons? Shall in contrast the metaDESK consider this angle of rotation data too by warping the two-dimensional map? And, even if these questions are answered, shall the physical size of the phicons, which implicitly determines the two-dimensional scaling, also be ignored? If not, how can this data be considered during computation and display of the two-dimensional map? Ullmer and Ishii (1997) have chosen the “ignore phicon angle of rotation data” alternative. Another situation of ambiguity is given by three phicons placed on the desk. Ullmer and Ishii suggest four alternatives to handle this three-phicon ambiguity:

a) "Warp to fit
b) Flag and ignore outlier
c) Best fit with error display
d) Discrete per-phicon views"

The metaDESK can be used for visualisation of existing or planned urban settings, as already implemented with the “Tangible Geospace” application. We think that other possible activities in which the metaDESK can be used are urban planning or production plant planning and design. What is not possible with metaDESK is to simulate a designed production plant, since metaDESK is not equipped with an appropriate software interface.

At the level of actions many alternatives exist. The active and the passive lens give the user freedom of choice and generate redundancy within the metaDESK. The phicons possess a facility of cueing because of their tool-oriented and user-oriented differentiation, such that users can distinguish phicons and thus better understand the output of the metaDESK. We define user-oriented differentiation of (physical) handling elements as the property of the handling elements to be distinguished by users, for instance through shape, colour, and material. Tool-oriented differentiation of handling elements is analogous: the differentiation characteristics are implemented according to the sensing technology of the tools.

Operation is facilitated through the physical presence of the handling elements, thus, virtual objects can be manipulated through familiar handling operations. Operationalization is facilitated through the intuitive concepts of handling: the active lens is easy to understand and is based on interaction with familiar physical elements, such as a window or a photo camera. Another factor for intuitive interaction with metaDESK is the coincidence of action space and perception space.

**Illuminating Light**

Illuminating Light (Underkoffler and Ishii, 1998) is part of a larger application called Urp which was designed for urban planning. We will shortly describe the properties of Illuminating Light. Illuminating Light is a system composed of a sensory system and a projector where physical handles used for interaction are physical representations of optical tools, such as mirrors or lenses. Some physical handles symbolise laser pointers (Fig. 5). A physical laser beam source icon (or phicon if we use the metaDESK terminology) is associated with a virtual, one-line simulated laser beam. The laser beam
can be reflected by a mirror phicon. Thereby, the user can model optical systems by using a joint virtual and real representation of the optical system.

Figure 5: The user interacting with Illuminating Light by manipulating physical icons; from left to right: a laser beam source phicon, a beamsplitter phicon and a mirror phicon.

The activity connected to Illuminating Light can be optical system design, but in general this system is used in connection with Urp. Urp delivers support for urban planning activities. It is possible with Urp to model the shadows cast by a building during daytime or to measure the real distance between two physical representations of buildings.

**Envisionment and Discovery Collaboratory (EDC)**

The Envisionment and Discovery Collaboratory (EDC) is a tool designed at the University of Colorado, Boulder (Arias et al., 2000). The EDC consists of a horizontal electronic whiteboard called action space. (In our terminology, the whiteboard is a coincident action-perception space). This whiteboard is touch-sensitive, so that placing and moving physical objects, such as a tree-shaped physical handle, allow interaction with the EDC. A second, vertical electronic whiteboard, named reflection space offers information concerning the object currently being manipulated and shows other, related objects. The reflection space represents an interface to a dynamic information space and to a simulation program (Fig. 6).

Using a touch-sensitive action space, it is possible to create virtual scenes and draw lines with specific physical handles (Fig. 7). It is also possible to run simulations of the created objects. Another feature supporting Computer-Supported Cooperative Work (CSCW) is the WWW-interface. With this interface, it is possible to place relevant information on the World Wide Web and allow distant interaction regarding, for example, an opinion poll.

With the EDC the mentioned authors are striving towards a co-evolutionary system, supporting distributed cooperative work. The purpose of the EDC is to allow even very large groups of stakeholders within a certain field to participate in design and development processes, improving thereby a collective involvement in issues of interest of a community.
The activity described is for urban planning. Within a given urban setting, appropriate bus routes are to be found (Fig. 8), considering the needs and habitudes of the population.

Cooperative work is possible
- at the action and reflection space, and
- through the World Wide Web.

In the first case, the user interfaces of the EDC are such that co-located work is possible. In the second case, people can deliver information about their needs for travelling by
bus, lower traffic in their residence zone, and the more. Based on these interfaces, different variants of bus stop locations can be simulated.

It seems that the background of the EDC development is politically influenced, since this tool offers new manners for the citizen to influence projects of common interest.

**i-LAND**

i-LAND is a vision and a realisation of workspaces of the future (Streitz et al., 1999). According to the GMD Darmstadt, IPSI, users shall be able to use as many architectural spaces as possible for single-user and cooperative work purposes. The environment around people shall become user interface in a more intensive way than it is designed currently. They define i-LAND as a modular composition of roomware components, combining architectural spaces with information technology.

![Fig. 9: The DynaWall(l.), the CommChair (m.), and the chair in use (r.).](image)

i-LAND consists of the following components: the DynaWall (Fig. 9), the CommChair (Fig. 9) and the InteracTable (Fig. 10). DynaWall is an electronically augmented wall on which virtual information elements can be fixed and moved. The CommChairs are seats with either a docking facility for laptops or a built-in laptop. The CommChairs can transfer data to and from the other components of i-LAND. The CommChair can be used in stand-alone mode as well. The InteracTable is supposed to make co-located work possible. It is a mobile interactive table for creation, display, discussion and annotation of information objects. Up to six co-located users can share the InteracTable.

Another component of i-LAND are the so-called Passengers (Fig. 10). A Passenger is not really a special object, but a specially marked physical object that can be attached to digital information. Since the information structures are not stored on the Passenger itself but just linked to it, people can either turn personal objects into a Passenger, like a watch, ring, or glasses, or they can use objects that are neutral in terms of personal value. The link between digital information and mark is operationalized with the "Bridge", a device that detects the mark and displays the associated information on an InteracTable, for example. This concept, named Passage, allows one to carry data using real objects.
As described previously in Chapter 5.7 with the InteracTable, any kind of information objects creation, display, annotation and discussion is supported by i-LAND. Collective creative processes are thereby supported, for example the design of a new product line for new market segments.

**InfoTable / InfoWall**

The InfoTable/InfoWall system (Rekimoto and Saitoh, 1999) consists of two image projections, one on the desk and the other one on a screen (Fig. 11). The idea behind the InfoTable/InfoWall was to enhance computer screens, especially laptops. One of the many functions (in other terms, tools) that InfoTable/InfoWall offer is similar to drag-and-drop, but not limited to the laptop screen. In other words, a text file can be virtually dragged from the laptop screen and dropped on the desk. The other user sitting at the desk can now drag the text file from the desk to his laptop. Other features are, the augmentation of physical objects with data. It is possible to associate a text file to a marked video tape on the InfoTable.

Every time the video tape is placed on the InfoTable, the text file appears together with it. The domains of application for the InfoTable/InfoWall can be various. InfoTable/InfoWall was not designed for a particular activity. It supports familiar GUIs. Rekimoto and Saitoh try to combine well-known GUI features with concepts from graspable user-interfaces, such as augmenting physical objects with digital information. This is also similar to what we described above in the Passenger concept of i-LAND.

Activities performed with support of InfoTable/InfoWall could be for example financial meetings, creative meetings, stand-alone design work or data classification work, and archive work or collaborative work on a shared digital document.
Appendix E: Acronyms, Abbreviations, and Vocabulary

2D Two-dimensional
3D i) Three-dimensional (default), or ii) a file format for virtual models
AR Augmented Reality
brick a physical handle with a reflective surface
BUILD-IT a multi-user computer-based tangible user interface for planning
C++ an object-oriented 3rd generation programming language
CAD Computer-Aided Design
CSCW Computer-Supported Cooperative Work
etc.dat a standard PC file defining parameters and values at a system level
gif a graphical two-dimensional image, a standard file format
GUI Graphical User Interface
HCI Human-Computer Interaction
HW Hardware
Inline a virtual model based on VRML model
MET++ a multimedia framework written in C++
PC Personal Computer
PDM Product Data Management, data used to describe virtual models
phandle a physical handle, integrating virtual and physical features
pre-CAD Design taking place prior to the use of a CAD system
roll a camera rotation around the line of sight
rot a camera rotation around the vertical axis
shift a horizontal camera movement
target actions actions required to carry out a specific activity
target technology a specific technology serving as tool to carry out target actions
task analysis a detailed description of each step of an expert user’s interaction with a computer application, e.g. as inputs and outputs
tilt a camera rotation around horizontal ort. axis to the line of sight
SGI O2 Silicon Graphics Workstation O2
TexturedBox a virtual model being a cube with one or more gif’s on the sides
TUI Tangible User Interface
UNIX an operating system for Workstations
VISDOME a visualization cave for the ETHZ
VR Virtual Reality
VRML Virtual Reality Modeling Language (versions: VRML1, VRML2)
<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows NT</td>
<td>an operating system for Personal Computers</td>
</tr>
<tr>
<td>wrl</td>
<td>a standard extension for VRML files</td>
</tr>
<tr>
<td>zoom</td>
<td>zoom corresponds to the field of view (degrees) of a camera</td>
</tr>
</tbody>
</table>
References

Ackermann, P. (1996): Developing Object-Oriented Multimedia Software. dpunkt
Verlag für digitale Technologie, Heidelberg.


Wiley-veh.

with tactile and force feedback. International Journal of Human-Computer Studies,
vol. 45, pp. 483-493.

Alty, J.L. (1998): Designing better interfaces: can we learn from other sectors?
Proceedings of the 7th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design and

Individual Human Mind - Creating Shared Understanding through Collaborative
Design. TOCHI, 7(1), pp 84-113, ACM.


Balakrishnan, R. and Hinckley, K. (1999). The Role of Kinesthetic Reference Frames in

object manipulation in 3D graphics interfaces. Proceedings of the CHI’99. New York,
ACM.

Use. In J. Carroll (ed.) Designing Interaction: Psychology at the human-computer
interface, New York: Cambridge U.P.


Bichsel, M. Segmenting Simply Connected Moving Objects in a Static Scene.
Transactions on Pattern Recognition and Machine Intelligence (PAMI), Vol. 16, No.


Virtual Buildings. In F. Crow, S.M. Pizer (eds.): Proceedings of 1986 Workshop on


Bødker, S. (1996): Applying Activity Theory to Video Analysis: How to Make Sense of
Video Data in Human-Computer Interaction. In B. Nardi (ed.) Context and

Campbell, J. (1988): Training design for performance improvement. In Campbell, J.,
Campbell, R. (eds.): Productivity in organizations. San Francisco: Jossey-Bass, 177-
216.

de Greef, P. and IJsselsteijn, W.A. (2000): Social Presence in the PhotoShare Tele-

Sciences, des Arts et des Métiers, par une Société de gens de lettres. Mis en ordre &
publié par M. Diderot; & quant à la Partie Mathématique, par M. D’Alembert.


Available at:
http://www.acm.org/sigchi/chi96/proceedings/panels/Eisenberg/EM_txt.htm


Available at:
http://www.dgp.utoronto.ca/people/GeorgeFitzmaurice/thesis/Thesis.gf.html


Curriculum Vitae

Name        Morten Fjeld
Marital status    Single
Date of birth   April 22, 1965
Citizenship    Norwegian
Languages       Norwegian (mother tongue)
                English, German, French, Danish, Swedish (fluent)

Education and Work Experiences

1997 -        Research assistant at ETH in Zurich
1990 - 1997   Software engineer, Oerlikon-Contraves AG in Zurich (CH)
1990          MSc in Mathematics at Norwegian Institute of Technology (NTNU)
1989 - 1990   Third year at Institut National Polytechnique in Grenoble (F)
1985 - 1989   Student of Mathematics at NTNU in Trondheim (N)
1984 - 1985   Military service as instructor at KNM Tordenskjold (N)
1981 - 1984   Senior high school at Fana Gymnas (N)
1972 - 1981   Primary- and secondary school in Bergen (N)