Investigations on Design Tools: Towards a Participatory Tangible Board

Jonathan Benn

A Thesis
in
the Department
of
Computer Science and Software Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of Master of Computer Science at
Concordia University
Montréal, Québec, Canada

May 2006

Your file Votre référence
ISBN: 978-0-494-20772-7
Our file Notre référence
ISBN: 978-0-494-20772-7

Abstract

Investigations on Design Tools: Towards a Participatory Tangible Board

Jonathan Benn

Today's computerized software design tools fail to support designers' creativity and collaboration. Investigations of designers show that paper and pencil is still the tool of choice for early design. Contemporary computerized tools allow designers to better communicate their designs. However, if tools lack adequate support for creative sketching and collaboration then they are not useful for supporting the design process, which is by nature creative and collaborative.

The Participatory Tangible Board (P-Tab) project aims to improve this situation by creating a new sensing-based interactive and tangible environment that will support a group of designers in collaboratively designing and communicating. Our goal is to elicit the requirements for the P-Tab and offer guidelines for its implementation. The empirical studies we conducted show that the P-Tab must support the following requirements: sketching, rapid prototyping, model evaluation, a horizontal display surface for small groups and a vertical display surface for large groups.

Acknowledgements

The author would like to thank his supervisor, Dr. Ahmed Seffah, for providing him with so many opportunities and for making his stay in the Concordia graduate program comfortable and productive. Without Dr. Seffah it would have been a much poorer experience.

The author thanks Dr. Ahmed Seffah and Rozita Naghshin for their help during the writing of this thesis. They provided invaluable advice on what to write, acted as sounding boards for generating ideas, and found many good references.

The author thanks the Participatory Tangible Board (P-Tab) research team for their thoughtful suggestions, helpful research and illustrations. Thanks go to: James Maciukenas, Rozita Naghshin, Dr. Ahmed Seffah, Professor Jason Lewis and Antoine Morris.

The author would also like to thank everyone that participated in the setup and execution of the Osmose usability tests: Dr. Ahmed Seffah, Dr. Sudhir Mudur, Ashraf Gaffar, Fanglei Wang, Harkirat Padda, Homa Javahery, Muhammad Faraz Anwar, Naouel Moha, Qing Li, Rozita Naghshin and Yojana Joshi. In particular, the author thanks Qing Li for helping to set up the test, replacing him when he was recovering from surgery, and helping him develop **Figure 6**.

The author thanks everyone that participated in making the P-Tab design studies a reality. For her equal contribution to the workload: Rozita Naghshin. For their help during the first study: Dr. Ahmed Seffah, Frank Radeke, Mohamed Taleb and Qing Li. The author also thanks the study participants, who cannot be named, for being so helpful and cooperative and for sacrificing one to two days of their lives to help in our research.

Finally, the author thanks Elena Alonso for her endless love and support.

Table of Contents

List of Tables

List of Figures

1 Introduction

Design in general is concerned with "the planning, development and production of products... Product designers tend to concentrate on the needs of people and the ways in which products can be made safe, easy to use, and a comfortable fit with the way people live [Owen, 2004]."

Software design, on the other hand, is concerned with the iterative development of a correct and fit software product, the description of software components, interfaces and algorithms, and the empirical measurement of design models and prototypes.

Concrete design tools are physical (or software-based) tools that designers can make use of to create designs. These range from simple but time-honoured tools like pencil and paper to sophisticated, computerized round-trip engineering software tools. Many computerized design tools promise great benefits in terms of increased productivity and improved designs. However, in order to achieve these purported benefits, tools need to support designers' genuine needs and comfortably fit into their methods and workplaces.

It is our belief that today's computerized software design tools are inadequate, with regard to the early stages of design and especially in terms of supporting creativity and collaboration. Despite the large number of tools available, very few seem to meet the basic needs of software developers. This can be seen by the fact that many, if not most, software designers still use pencil and paper, or marker and whiteboard, when creating their designs. Most computerized tools available today are useful after this pencil and paper stage, when the important design decisions have already been made. Often, the computerized tools serve simply as a means to create a clean copy of quick and dirty diagrams drawn and verified by hand. Very few tools, however, aid designers in sketching those early diagrams or help them make those design decisions.

The Participatory Tangible Board (P-Tab) is a project, spearheaded by the Human-Centered Software Engineering (HCSE) research group, whose purpose is to fill this gap. The P-Tab is envisioned as a software design environment that software designers, user interface designers and graphic designers can use to perform their early- and late-stage design work. The P-Tab should use smart boards, tangible metaphors, virtual reality, mixed reality, software agents, and any other techniques necessary to dispense with pencil and paper and make the P-Tab the tool of choice for designers. The P-Tab should help designers produce better designs, faster, by providing sketching capability, rapid prototyping techniques, and feedback on design quality.

The P-Tab's nature and requirements at this point are very vague. It is an immensely ambitious project. At its conception it was envisioned as a horizontal smart board that a team of multi-disciplinary designers could sit around and use to facilitate design sessions. However, theoretical and practical investigations were required to bear out this idea, or show that it would not accomplish the P-Tab's goal of transparently helping designers work.

In order to uncover the P-Tab's requirements, we must first investigate design literature to better understand the nature of design and how the P-Tab can fit into it. We must also identify and address the drawbacks of modern computerized design tools. This will allow us to know what is missing and what the P-Tab needs to do better. We must observe other unconventional design environments in order to understand how their techniques and technologies can be applied to the P-Tab. Finally, we must observe software designers in action in order to see what their needs are and how the P-Tab must be designed in order to meet those needs.

1.1 Research Objectives

Our research goals are as follows:

- To understand the requirements of the P-Tab environment.
- To offer guidelines for implementing the P-Tab.

As an innovative design environment, the P-Tab exposes sensing-based metaphors for human-to-human interaction as well as for interaction between the design artefacts (represented by tangible objects) and one or more designers. By using the sensing-based interactions paradigm to foster communication and creativity, the P-Tab will be superior to current design environments, which provide a Window-Icon-Menu-Pointer (WIMP)-based interface responsible for the current gap between how designers think and create a design and how tools work. We will attempt to elicit requirements for the P-Tab environment in pursuance with this overall vision.

In the original research proposal, the P-Tab environment is built around a physical table that embeds a large interactive whiteboard and a second monitor to create a tangible workspace. Design artefacts (*e.g.* software classes, user interface widgets, or multimedia documents) are represented by physical forms that can be directly manipulated and perceived by human sensors. Sensing-based interaction mechanisms allow designers to interact directly with the tangible forms on the large whiteboard. While prototyping using basic design objects and patterns, designers can also annotate their designs using, for example, real-time audio/video. Our upcoming research will either confirm or deny the effectiveness of this proposal. If necessary, we will offer guidelines for a different physical environment.

1.2 Research Scope

In this thesis, we will be using a variety of techniques for eliciting the requirements for the P-Tab project. This includes research into design theories, evaluation of existing tools, investigations into non-conventional design tools as well as empirical studies. In addition, we will also offer a preliminary look at what a finished P-Tab might look like in order to meet these requirements. However, a detailed design and implementation of the P-Tab remains the topic of future research.

1.3 Research Methodology

In order to accomplish our goals, our main investigations will be as follows:

- To research the state of the art in design theory and learn more about the nature of software design.
- To investigate and critique existing design tools.

- To explore how designers might benefit from the new immersive and tangible interface technologies that are being developed.
- To design, conduct and analyze two empirical studies of software designers. By studying designers performing design tasks in a multi-disciplinary environment, we aim to better understand the social and cognitive processes that underlie design activities.

1.4 Thesis Organisation

This thesis is divided into 11 sections:

1. **Introduction:** describes the research context and overall concepts. Offers information on our research objectives, scope, methodology and the contents of this thesis.
2. **Design survey:** shows the results of research into design theory and software design theory. Describes relevant concrete design tools and design innovations.
3. **Osmose experience report:** provides a detailed description of the usability test setup performed on a portable virtual reality installation, and lessons learned.
4. **VRAD experience report:** contains observations of a fixed virtual reality installation and its design software, and discussion of the impact on creativity.
5. **December field study:** provides a description of our first empirical study and what we learned from it.
6. **February field study:** describes our second empirical study and what we learned.
7. **Conclusion:** contains a summary of our findings, as well as requirements and guidelines for the P-Tab.
8. **References:** the collected references for this thesis.
9. **Osmose appendix:** contains ancillary documentation used to prepare the test.
10. **December study appendix:** has auxiliary information such as documentation used during preparation and execution, and a summary of the data results.
11. **February study appendix:** provides subsidiary data (documentation and results).

2 A Brief Survey of Software Design

We will now perform a survey of design literature, design tools and new ideas and technologies. This survey will not be exhaustive, but will nonetheless serve to provide an overview of the scientific context in which our research was conducted. It will also serve to elicit some requirements for the Participatory Tangible Board (P-Tab). Our research was not carried out in isolation, but instead was an elaboration of and contribution to pre-existing thoughts, ideas and research.

2.1 Design

The question, "What is Design?" is a surprisingly difficult one to answer.

2.1.1 Competing Definitions of Design

According to [Owen, 2004], design is concerned with "the planning, development and production of products — simple ones like can openers and others that are actually complex systems of products such as cars, airplanes and whole transportation systems as well as complex software systems. Product designers tend to concentrate on the needs of people and the ways in which products can be made safe, easy to use, and a comfortable fit with the way people live."

According to [Budgen, 2003], design is a process that has behaviour, as well as structure. Budgen distinguishes between design/engineering and science. Design is creating the future, whereas science is observing the present. Budgen makes the following quote to explain why design is difficult: "Designers have to work backwards in time from an assumed effect upon the world to the beginning of a chain of events that will bring the effect about [Jones, 1970]."

According to Budgen, design involves four steps: (1) postulating a solution, (2) modeling the solution, (3) validating the model and (4) elaborating the model to produce a solution plan. Design is hampered by constraints such as performance or manufacturing cost.

Design requires abstraction, which "is concerned with the removal of detail from a description of a problem, while still retaining the essential properties of its structure."

The definition from [Shedroff, 2001] combines the above points of view: "design is a set of fields for problem-solving that uses user-centric approaches to understand user needs (as well as business, economic, environmental, social, and other requirements) to create successful solutions that solve real problems. Design is often used as a process to create real change within a system or market. Too often, Design is defined only as visual problem solving or communication because of the predominance of graphic designers… a recognition of the similarities between all design disciplines shows that the larger definition for Design operates at a higher level and across many media."

According to [IEEE, 1990], design is "the process of defining the architecture, components, interfaces, and other characteristics of a system or component."

2.1.2 Design versus Art

When attempting to learn what something is, it can help to learn what it is not in order to gain a more complete understanding. According to [Owen, 2004]:

"Designers generally work objectively on teams… to create the products, systems, communications and services needed by society. They are outer-directed, work for others, and use a wide range of design and planning tools to collect and organize information in the process of developing the things people need and want to improve their quality of life... [They] frequently work with professionals from other disciplines in the development of complex systems requiring broad expertise."

"Artists work mostly subjectively with the motivation of self-expression to produce works fulfilling aesthetic and intellectually stimulating objectives. They are inner-directed, usually work for themselves (or individual clients), and use primarily intuitive and personally developed skills. Some techniques and processes overlap (the perceived similarity between designers and artists stems from a common use of visual media to communicate ideas), but the fundamental methods, results and—most important—goals are quite different."

2.1.3 Conclusion on the Nature of Design

From the definitions of design and its comparison with art, we can conclude that design is concerned with:

- Making a product that meets the needs of others
- Making the product safe, usable and comfortable
- Creating a quality human-machine interface
- Problem-solving
- Architecture, components, interfaces and relationships

Therefore, a good design environment must support each of these components of design.

2.2 Software Design

Most software engineering definitions of design take a process-oriented, component-based view, such as that seen in [IEEE, 1990]. For example, the Software Engineering Body of Knowledge (SWEBOK) [IEEE, 2005] states that software design "must describe the software architecture—that is, how software is decomposed and organized into components—and the interfaces between those components. It must also describe the components at a level of detail that enable their construction."

According to [Sommerville, 2001], "a software design is a description of the structure of the software to be implemented, the data which is part of the system, the interfaces between system components and, sometimes, the algorithms used. Designers do not arrive at a finished design immediately but develop the design iteratively through a number of different versions."

[Gould, 1985] approaches software design from the perspective of usability. The authors define three principles of design: (1) "Early focus on users and tasks... designers must understand who the users will be." (2) "Empirical measurement... early in the development process, intended users should actually use simulations and prototypes to carry out real work…" (3) "Iterative design... when problems are found in user testing, as they will be, they must be fixed."

Note the similarity between the SWEBOK and Sommerville definitions of software design, and how different they are from the Gould definition. This is not necessarily a contradiction. Verifying is designing the thing right (correctness), whereas validating is designing the right thing (fitness for a purpose) [Boehm, 1979]. The SWEBOK and the Sommerville design definitions can be seen as verifying approaches. Gould's design definition can be seen as a validating approach. When seen in this way, the competing definitions are not contradictory, but complementary.

From these definitions we can conclude that software design is concerned with:

- The construction of a correct and fit software product
- Iterative development
- The description of software components, interfaces and algorithms
- Empirical measurement of design models and prototypes

2.3 A Brief Survey of Design Tools

Software design practitioners often seek tools to help them reduce the time and effort required for their work, and to improve the quality of their results. Here we are using the term *tool* to refer to any technique or technology that aids a designer.

The SWEBOK [IEEE, 2005] describes design tools in terms of *general strategies* and *methods*. General strategies "help guide the design process." In the other hand, methods are more specific. They tend to "suggest and provide a set of notations," describe "the process to be used," and offer a "set of guidelines on using the method." *Concrete tools*, such as computer programs that assist the design process, are not mentioned in that part of the SWEBOK, but are another important kind of design tool. Examples from each category of design tools will be surveyed below.

2.3.1 General Strategies for Software Design

One of the best known general strategies is Design for Change (DFC). This concept has had many different names over time, including *information hiding* [Parnas, 1972], the *Bridge pattern* [Gamma, 1995], *commonality and variability analysis* [Coplien, 1998], and *protected variations* [Larman, 2005]. The essential idea is to (1) collect all of the major design concepts into chunks, and hide each chunk behind an interface, and (2) determine which concepts are likely to change over time and apply additional effort in designing those components. DFC's most important impact is on software maintenance, which for successful software amounts to the longest and most costly part of the development process.

User-Centered Design (UCD) is another general strategy. First introduced as a form of usability engineering, UCD embodies three principles: early focus on users and tasks, empirical measurement and iterative design [Gould, 1985]. In UCD designers create simulations and prototypes of the final product and then test these with representative end users. The feedback from these empirical tests is used to drive the iterative development of the next prototype, and so on until the next prototype becomes the final product. While it is not required, design teams in UCD are frequently multi-disciplinary (with each person having a different background and problem-solving approaches).

Using *design patterns* is a general strategy that originates from architecture. "Each pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice [Alexander, 1977]." The seminal work on *software* design patterns is [Gamma, 1995]. Design patterns also provide designers with a vocabulary for conceptualizing problem-solving and communicating with stakeholders [Shalloway, 2001].

Design activities form a general strategy that contributes to designing a product. Their inspiration comes from aerospace engineering. There are six design activities: sketching, annotating, storyboarding, animating, simulating and prototyping [Seffah, 2005].

- **Sketching:** a quick outline or drawing showing the overall theme and main elements of a design.
- **Annotating:** adding additional information to an existing sketch; appending a critical or explanatory note.
- **Storyboarding:** stringing together a series of sketches in order to model the passage of time.
- **Prototyping:** creating a working model suitable for evaluating a design.
- **Animating:** adding motion to a design model.
- **Simulating:** adding the environment's effects to a model animation.

2.3.2 Software Design Methods

The Unified Process (UP) is a typical example of a software design method. In UP, a designer or team of designers directs the design in a hierarchical way. First, large systems are defined with the Unified Modeling Language (UML). Then these systems are broken down into smaller pieces, which are themselves defined and then broken down further. This process is repeated until the design is sufficiently detailed to allow the reasonably predictable implementation of the product. UP is iterative, and tends to overlap the different stages of software development [Larman, 2005].

Another example of a software design method is Design Space Analysis (DSA) [MacLean, 1993], which uses the Questions-Options-Criteria notation. The "design space is an explicit representation of alternative design options, and the reasons for choosing among those options… Questions highlight key issues in the design space. Options can be thought of as 'answers' to Questions. Criteria are the reasons that argue for or against possible Options." DSA serves three complementary roles: to help designers make design decisions, to communicate design decisions to other stakeholders, and to explicitly document design rationale as an aid to maintainers.

Test-Driven Development (TDD) is a design method. When writing software using TDD, software test-cases are written first and then source code is written in order to pass those tests. Since in TDD the primary design activity is writing and passing tests, then the resulting software can be seen as an emergent property of that activity. In this way TDD simulates the natural selection of source code with the test cases as selection criteria, a potentially unpredictable and leaderless system. TDD combines design and implementation [Beck, 2002].

Another example of a design method is Responsibility-Driven Design (RDD) [Wirfs-Brock, 2002]. In RDD, classes have responsibilities and communicate with other classes. The first thing a designer does is to assign each class responsibilities, ideally, one responsibility per object. Then the designer can determine other class characteristics (*e.g.* methods, attributes and collaborators) based on these responsibilities. RDD helps designers make better low-level design decisions. According to [IEEE, 2005], RDD is an alternative to the Object-Oriented Design (OOD) method, since RDD uses responsibilities to create components instead of using noun-verb-adjective analysis of the problem domain. However, RDD can be seen as a complement to OOD, rather than as a replacement, since both methods can be used together during analysis as seen in [Larman, 2005].

2.3.3 Concrete Design Tools

Concrete design tools are physical (or software-based) tools that designers can make use of to create designs. **Table 1** describes a variety of existing tools.

Table 1: Concrete Design Tools

Design Tool	Description	Benefits & Drawbacks
Computerized tool	Typical features include: cut/copy/paste, click and drag, save/load	Saves time and effort Requires a computer, which generally makes group work harder

Design Tool	Description	Benefits & Drawbacks
Pencil and paper	The basic design tool	Full support for sketching Infinite creative scope, in terms of what can be conceptualized No computerized features Ineffective for more than a few people
Rational Rose	Software for designing software systems using the UML	Cannot sketch diagrams UML records design results, not design rationale Strict syntax makes brainstorming difficult Largely ignores user-interface and other non-functional aspects of the system under construction
SMART Ideas	Concept-mapping software for brainstorming, teaching or explaining	Quick and easy No sketching capability Zoom levels exist, but are not usable

Design Tool	Description	Benefits & Drawbacks
Microsoft Visio	Software for creating diagrams representing any system	Quick and reasonably easy to use Infinite creative scope Zooming supported Sketching not supported No layers No automated rapid prototype generation
Whiteboard	A large, vertical surface upon which dry-erase markers are used to write	Full support for sketching Infinite creative scope No computerized features, although erasure is relatively painless Useful for large groups due to high visibility of anything written on its surface
Sticky notes	Small pieces of paper, usually yellow, that have a special kind of glue on the back that leaves no residue and can be reused.	Full support for sketching Infinite creative scope Notes can be moved around, simulating computerized click & drag Otherwise lacks computerized features

2.4 A Brief Survey of Design Innovations

Table 2 describes a sampling of some new ideas, techniques and interactive environments that are relevant to cutting-edge design today. Some of these ideas have existed for a long time, but they have yet to become mainstream ideas and hence are listed here as innovations.

Table 2: Innovations Relevant to Design

Innovation	**Description**	**Benefits & Drawbacks**
Fisheye views [Turetken 2004]	A method for zooming into a portion of a diagram, while retaining the context in smaller form.	Allows better communication of design ideas
Layers	A software-based technique similar to drawing on multiple transparent sheets of plastic that are stacked one on top of the other, and then being able to lift a particular sheet to not show the drawings on it.	Easy and intuitive to show changes over time—no need to copy elements that stay the same. Allows conceptually related elements of a diagram to be shown or not shown, depending on what needs to be communicated.
Software agents [Repenning, 2004]	Small, simple computer programs that when combined form a more sophisticated and intelligent whole.	Can provide designers with rapid feedback on the quality of a design Can offer advice on directions to take

Innovation	**Description**	**Benefits & Drawbacks**
Rapid prototyping	Software-based technique for allowing the easy creation of a working prototype	Greatly helps creativity by allowing designers to create and evaluate more prototypes—more prototypes mean a higher likelihood of finding the best.
DENIM [Newman, 2003]	Sketch-based software for designing web pages	Zoom levels Rapid prototyping Sketching supported Input recognition imperfect No layers
DEMAIS [Bailey, 2002]	Behaviour-sketching tool for early multimedia design	Full support for sketching Uses layers and rapid prototyping
IBM Dreamspace [Lucente, 1998]	Smart-room that combines speech recognition and machine-vision tracking, and requires no hand-held or worn devices	Natural method of interacting with a computer. Resolution of sensors limits utility How does the computer differentiate between verbal commands and conversation?

Innovation	Description	Benefits & Drawbacks
Augmented reality	Overlaying virtual reality on top of reality	Allows Heads-Up Display (HUD) for showing information while a designer works Allows richer information to be overlaid on top of everyday objects
Two-hands GUI paradigm [Kurtenbach, 1996]	Using two tablets and a software sketching tool to allow designers to use both hands during sketching	Allows off-hand to be used unconsciously Enables two-handed tasks
DiamondSpin and DiamondTouch [Shen, 2004]	Smart table that allows multiple simultaneous inputs from hands and fingers, and a toolkit for creating applications for the table.	Table allows groups to work together in a natural way Toolkit allows prototyping and evaluation of ideas
SmartSkin [Rekimoto, 2002]	Like DiamondTouch, allows multiple inputs from hands and fingers.	Allows groups to work together at a table in a natural way
Haptic sensor	A 3D joystick that allows users to physically interact with a 3D world and feel resistance.	Provides a method for tangible interaction with a virtual world
Ambient ROOM [Ishii, 1998]	A special room that allows ambient sensory input (*e.g.* sound, light, movement) to convey information.	Takes advantage of subconscious processes in the brain to convey information without distracting

Innovation	Description	Benefits & Drawbacks
Natural programming [Myers, 2004]	A technique for designing programming languages that more closely resembles natural human problem-solving methods.	HANDS: Event-based and set-based programming language that avoids iteration and Boolean logic—easier to use Why Line: debugger allows programmers to make and test hypotheses
SMART Board interactive whiteboard	Touch-sensitive whiteboard that also displays the contents of a computer screen.	Full support for sketching Useful for large groups, since visibility of information is high
UML Sketch Recognition [Hammond, 2002]	Allows hand-drawn UML sketches to be scanned into a computer and converted into formal diagrams	Allows designers to sketch without compromising convenience

2.5 Concluding Remarks

From what we learned about software design, we can make the following recommendations for the P-Tab:

- **The description of software components, interfaces and algorithms:** the P-Tab needs to support sketching, so that early designs can be brainstormed. It also needs to support more sophisticated, later-design-stage tools so that designs can be formalized.
- **Iterative development:** the P-Tab needs to support automated sketch-cleanup and annotation. This way, early designs can be cleaned up for communication

with stakeholders and then annotated with additional information as the design evolves.

- **Empirical measurement of design models and prototypes:** the P-Tab needs to support rapid prototyping so that prototypes can be measured and evaluated. This could be implemented with a natural programming language.
- **The construction of a correct and fit software product:** the P-Tab needs mechanisms for providing feedback on design quality and for providing design advice. These mechanisms could be implemented with software agents.

Based on our research, we also recommend the following features because we consider them to be potentially useful:

- Time-saving computerized features, *e.g.* cut/copy/paste, load/save, etc.
- Ability to run familiar, contemporary software, *e.g.* Rational Rose, Adobe Photoshop, etc.
- Ability to receive multiple interactions simultaneously, by different people or by both of a single person's hands
- Support for zoom levels, so that parts of sketches or diagrams can be compressed or expanded. If possible, this could implement fisheye views.
- Support for layers in sketches and diagrams

3 Osmose Experience Report

In this experience report we will describe how we developed a new testing environment and methodology for performing usability tests on an unconventional computerized interactive system.

3.1 *Motivation*

3.1.1 Overall Motivation

Our overall research goals were to elicit requirements for the Participatory Tangible Board (P-Tab) and uncover some possible design directions. We needed to observe existing tangible and other unconventional computerized systems in order to better understand these requirements.

Through one of the Human-Centered Software Engineering (HCSE) group's research projects [McRae, 2004] we gained access to *Osmose* [Davies, 1996; Davies, 2004 and Immersence, 2006], an unconventional interactive system. Osmose was an ideal candidate for our experiment because its Head-Mounted Display (HMD) interface, motion tracking and breathing mechanism were very unconventional relative to a desktop computer.

We joined the beginning of this new research project so that we could (1) get into contact with and learn more about a real unconventional interactive system, and (2) help our research group with its goals.

As part of the Osmose project we developed a test environment and methodology for performing usability tests. Our test environment would be subsequently used to perform a usability study. This study would be the first step of a three year project that aimed to design and develop a new Osmose prototype that would optimally meet its quality-in-use requirements.

3.1.2 Motivation Within the Osmose Project

Empirically testing a software product's usability is an important aspect of validating it [Gould, 1985]. Without empirical usability testing, it is difficult to know if a product will actually be useful to its intended audience, and if their use of the product will be effective, efficient and enjoyable. Direct observation of users as they use the product can help evaluate the product's usability in an unbiased way. This direct observation is called *usability testing*. When usability testing is combined with other techniques, like user questionnaires and expert heuristic evaluation, it can help provide a complete picture of the product's usability.

In the field of software usability testing, it is known today [Dumas, 1999 and Mayhew, 1999] that many stationary and mobile usability testing labs suffer from a critical problem: there is an implicit assumption that the *participant* (*i.e.* the representative end users) will be sitting at a desktop computer with a mouse and keyboard for input and a monitor for output. However, the range of computerized systems is far greater than just desktop applications. In particular, unconventional input/output systems (such as augmented reality and virtual reality) and mobile applications (such as pagers, cellular phones and personal digital assistants) are difficult or impossible to test using such labs [Dumas, 2004]. For Virtual Reality (VR) many difficult design questions arise [Sherman, 2003], such as: how do we record what the participant is seeing when the computer's monitor is really a helmet? How do we record the participant's interaction with the system when the participant is able to freely move about the entire world?

The importance of usability testing is just as great for unconventional non-desktop systems as it is for conventional desktop ones. Since the usability testing of non-desktop applications is difficult using traditional approaches, there is a need today to develop new usability testing environments and methodologies.

To help meet this need we aimed to develop a new test environment and test methodology for performing empirical usability tests on an unconventional interactive system called Osmose. Osmose was a public exhibit that had to serve dozens of different people each day. This gave it particularly stringent usability requirements and made it an excellent candidate for usability testing. While the new usability test environment would not be generally applicable to all VR systems, it would nonetheless serve as an example of how to bridge the gap between new interactive technologies and the old usability testing techniques.

3.2 Goals

In creating the new test environment and methodology, our goals were to:

- Observe first-hand an unconventional interactive system.
- Help our research group perform its three-year study.
- Create a testing environment that was low-cost, using only off-the-shelf components in new ways.
- Describe the new technique, and the process used to create it, at a level of detail sufficient to allow others to repeat our experiment.
- Develop a practical expertise in the usability testing of unconventional computerized systems.
- Customize pre-existing usability testing methods to an HMD.

3.3 Method, Test Plan and Infrastructure

Our test plan was to take two weeks in late January 2005 to set up the test environment and methodology for a usability test of Osmose. We planned to use the infrastructure to perform the usability test between February and April 2005.

3.3.1 Osmose Description

Osmose is "an immersive interactive virtual-reality environment installation with 3D computer graphics and interactive 3D sound, a head-mounted display and real-time motion tracking based on breathing and balance [Immersence, 2006]."

Osmose is an HMD-based VR system that allows its user to observe and move through the virtual world using real-time motion tracking and breathing measurements. It can usually be found on display in art galleries and museums. Osmose was created by Char Davies, a pioneer of VR and one of Canada's most awarded artists. Osmose has been widely exhibited across the world, in places such as: the Museum of Modern Art (San Francisco), the Australian Center for the Moving Image (Melbourne), the Barbican Centre (London), the Museum of Monterrey (Mexico), the National Gallery of Canada (Ottawa) and the Museum of Contemporary Art (Montreal).

Osmose is composed of the following components:

- A server (see **Figure 1**), made up of two storage racks containing four Personal Computers (PCs): two with the Windows operating system, one with Macintosh and one with Linux. The racks also contain sophisticated multimedia processing hardware that provides a variety of audio-visual outputs in various formats.
- A helmet (see **Figure 2**), also known as an HMD, containing two low-resolution stereoscopic monitors offering a wide 180 degree field of view, speakers, and a magnetic sensor that detects the helmet's position and orientation. See **Figure 5** for an example of what a participant might see while wearing the helmet.
- A vest (see **Figure 3** and **Figure 4**), which measures participants' breathing (by measuring chest expansion via a pressure gauge) and the orientation of their upper body.

Most of the components of the vest and helmet are connected to the server by wire. The exception is the helmet's magnetic sensor, which is wireless but short-ranged. Its signal is captured by the magnetic sensor controller that is placed above the participant's head (see **Figure 4**). Hence, the participant does not need to be at the same location as the server. However, the participant cannot stray very far from the magnetic sensor controller.

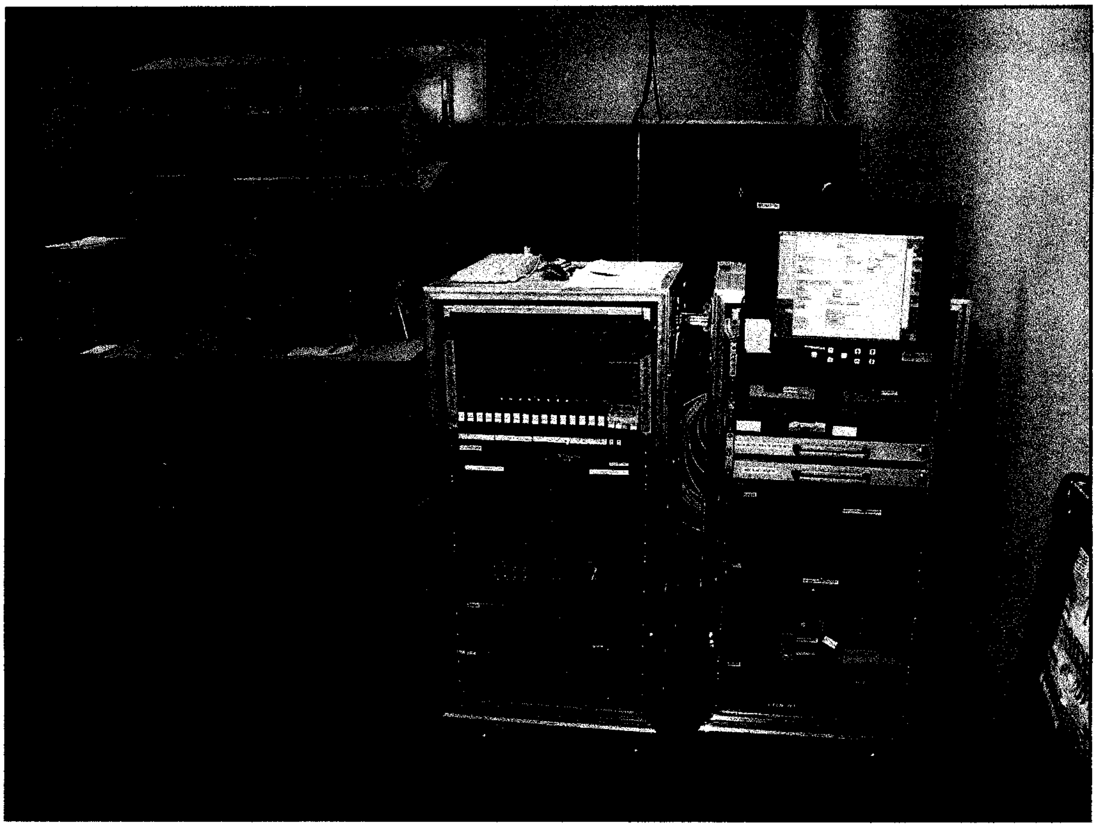


Figure 1: Test Back-End (left) and Osmose Server (right)

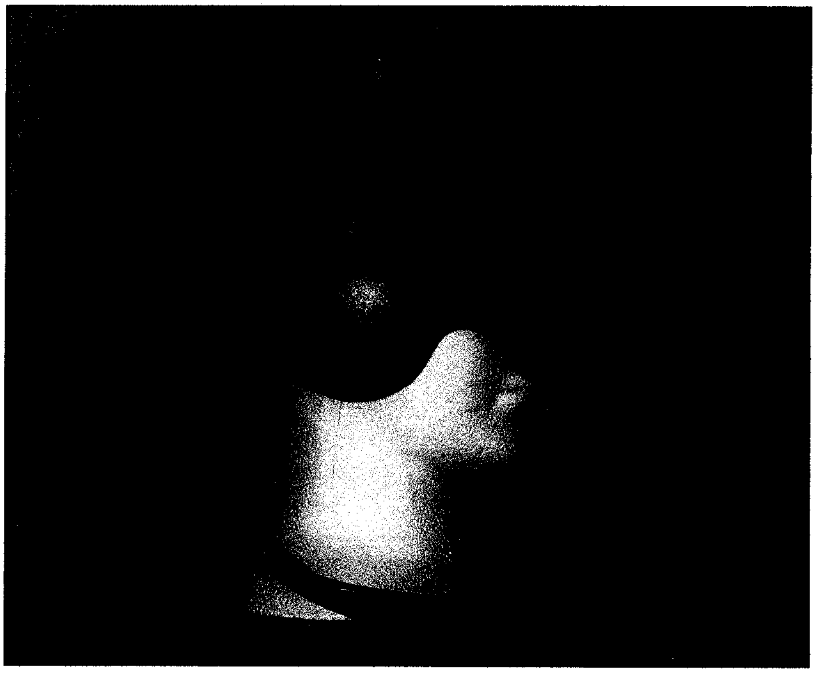


Figure 2: Osmose Helmet

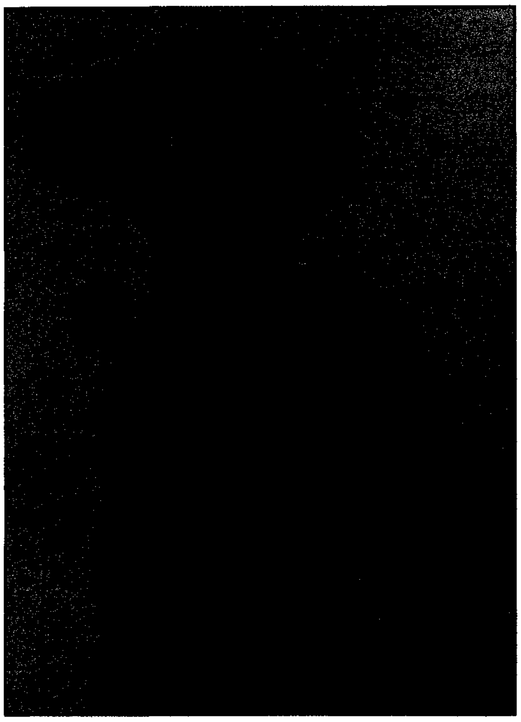


Figure 3: Osmose Helmet and Vest

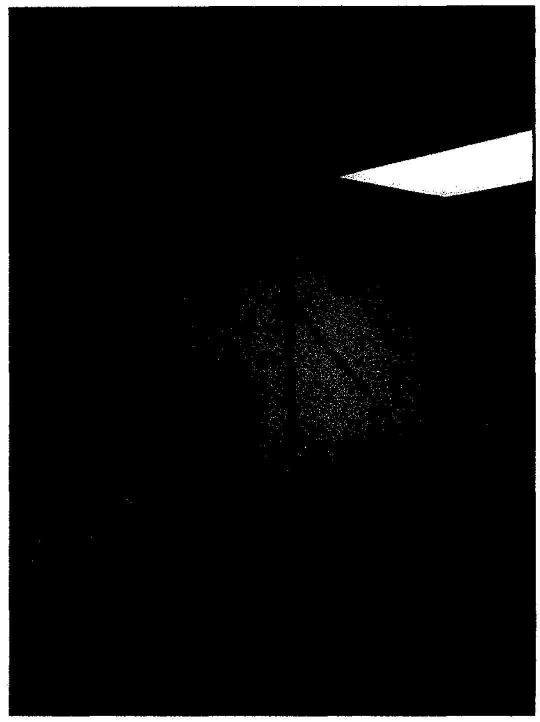


Figure 4: Osmose Magnetic Sensor Controller

[Immersence, 2006]

Figure 5: Subterranean Earth Environment from Osmose

Participants interact with the virtual reality in the following ways:

- To look in any direction, the participants simply turn their head. They can look down, through their own body, up and all around.
- To move forward, backward or to the side, participants bend their upper body at the waist. The direction leaned in is the direction that the participants move. The greater the angle of the lean, the greater the speed they will move in the virtual world.
- To move up participants inhale. To move down participants exhale. As a result, deep breathing can cause them to bob up and down in the virtual world.

While it's not necessary for participants to move their feet in order to move in the virtual world, they have the tendency to shuffle their feet or move around anyway. If participants move too far from that magnetic sensor controller, it will degrade the accuracy of the magnetic sensor and cause the virtual world to jitter.

Since Osmose was designed to be available for public display, it has a built-in mechanism to ensure that each participant's session lasts about 15 minutes.

Turning on Osmose is a step-by-step process that takes approximately 30 minutes. Turning it off is simpler and only takes about 5 minutes.

3.3.2 What Made Osmose Difficult To Test

During a typical usability test, testers ask participants to perform a series of tasks and then the testers record the participants' experiences [Dumas, 1999]. Testers record three types of data: (1) the participants' interaction with the system, (2) what the participants are seeing, and (3) the participants themselves. While testing these is normally a challenge for desktop applications, Osmose was even more difficult to test. Here are some reasons why:

1. The style of interaction, instead of being through a mouse and keyboard, is through unusual mechanisms: breathing and movement of the head and body.

Therefore, it is more difficult to record the participant's interaction with the system since we cannot simply record mouse clicks and keystrokes.

2. Participants are relatively mobile. Instead of being confined to a chair in front of a computer monitor, they have a certain degree of freedom (they may walk around within limits, and look in many directions). This makes participants harder to record.
3. Running Osmose requires darkness for participants to have an immersive experience. This makes observing them more problematic.
4. Recording what participants see is a challenge, since they are seeing two images originating from inside a helmet, rather than a single image coming from a standard computer monitor.
5. Obtaining all of the required information will result in multiple data streams (*e.g.* participants, what they are saying, and what they are seeing). The design challenge is how to *inexpensively* assemble these data streams so that they can be reviewed in a simultaneous and synchronized way.

3.3.3 Test Setup

In creating the test environment, the equipment we used included: two computers, a digital camera, a wireless microphone, a webcam, and software such as Morae (by TechSmith) and Premiere (by Adobe). For a full list of the equipment used see **Section 9.1**, located on page 87.

We began setting up Osmose's usability testing environment about two weeks before the start of our first test. One of the first things we did in preparation for the tests was to obtain off-the-shelf software called Morae. Morae is a piece of software designed to perform usability tests in a traditional testing environment. In other words, it expects a participant sitting at a desktop computer in a stationary setting. It records the participants' interaction with the system by recording the computer's screen output, as well as mouse clicks and keystrokes. It records participants themselves via a webcam and a microphone. Morae then combines all of the different data streams into one video file. It also offers important and useful features such as real-time remote viewing and annotation of a recording.

Figure 6 shows the test environment we set up in order to test Osmose. In the diagram, monitor (A) showed what participants could see with their right eye, while monitors (B) and (C) showed their left eye view. The speakers (D) played Osmose's musical accompaniment.

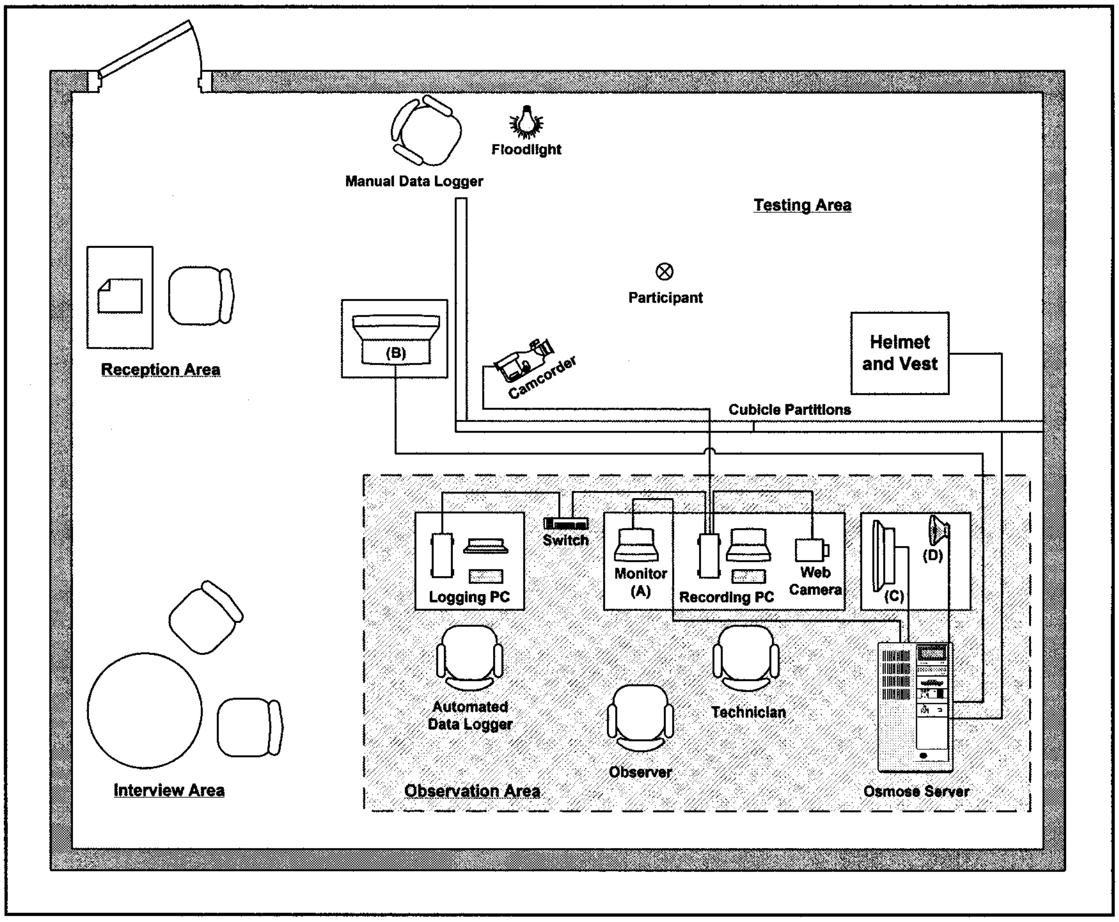


Figure 6: Osmose Test Environment

We could not install Morae on the Osmose server in the hope of recording the participant's interaction with it. This was because we were not allowed to disrupt Osmose's carefully custom-designed software by installing new software. Since Morae only functions on Windows, and it's the Linux PC that ran Osmose's graphics, we couldn't install Morae on the appropriate computer anyway. Hence, we were not sure how we could effectively use Morae in this situation.

We turned our attention to how to record the participant's interaction with Osmose. The fact that participants had to stay near the magnetic sensor controller (in **Figure 4**) made our work easier. We marked an X on the ground where participants were supposed to stay during the test (see the label *"Participant"* in **Figure 6**).

To record participants, we placed a digital camcorder facing the X on the ground (see **Figure 7**). After some pre-tests, putting ourselves in place of the participants, we could see that even though participants could move around during the test, they couldn't move enough to leave the field of view of the camera. What's more, since interaction with Osmose is limited to upper body motion, recording the participants' upper body also allowed us to record their interaction with the system. This solved problems 1 and 2 (as described in **Section 3.3.2**).

Even though darkness was required for a good Osmose experience (problem 3), we were able to use a floodlight and the camcorder's low light mode in order to overcome that limitation. Some pre-testing allowed us to adjust the lighting in order to produce optimal results. We had overcome the lighting problem.

One question that remained at this point was how to store information recorded by the camcorder. We could have recorded the data onto mini-DV tapes. The trouble with that option was that if we also separately recorded what the participant saw (problem 4), then we would have had two un-synchronized separate data sets (problem 5). We wondered how we could stitch together these two data streams. One technique would have been to buy or rent very expensive TV broadcasting hardware, which would have allowed us to merge multiple input signals and combine them into one stream. However, this was too expensive (on the order of $100 000 Canadian dollars to purchase). We had to think of other options. This was where Morae came in. We realized that we could use a computer to inexpensively assemble all of the different streams of data.

Figure 7: Camcorder for Observing Participants

We used the software Premiere to display the camcorder's output on the Recording PC's monitor—thus using the camcorder as an electronic eye rather than a recording device. Then we set up a webcam to observe monitor C (in **Figure 6**), which allowed us to see and record what the participant would be seeing (see **Figure 8** for a picture of the webcam setup). Pre-tests showed us that the recorded video quality was low, but that it was still good enough for our purposes. We didn't need to be able to see tiny details. We only needed to know what was happening in Osmose's virtual world at any given time.

Finally, by running Morae Recorder we were able to record what the Recording PC was showing on its monitor (thus capturing the camcorder's output) and simultaneously record from the webcam. This was all saved in a single synchronous video file, which was exactly what we needed. This technique solved our remaining problems (4 and 5).

There was one last thing we wanted to record, which was what participants would be saying during the test. We attached the wireless microphone transmitter to Osmose's vest, and then plugged the receiver into the Recording PC's sound card. Morae was able to take the input from the wireless microphone and integrate it into the single recorded file. We also connected a set of headphones to the wireless receiver so that the testers could hear the participant clearly, if necessary, even through the cubicle partitions that muffled the participant's voice. This way the data loggers could clearly hear the participant, without the participant being able to see them.

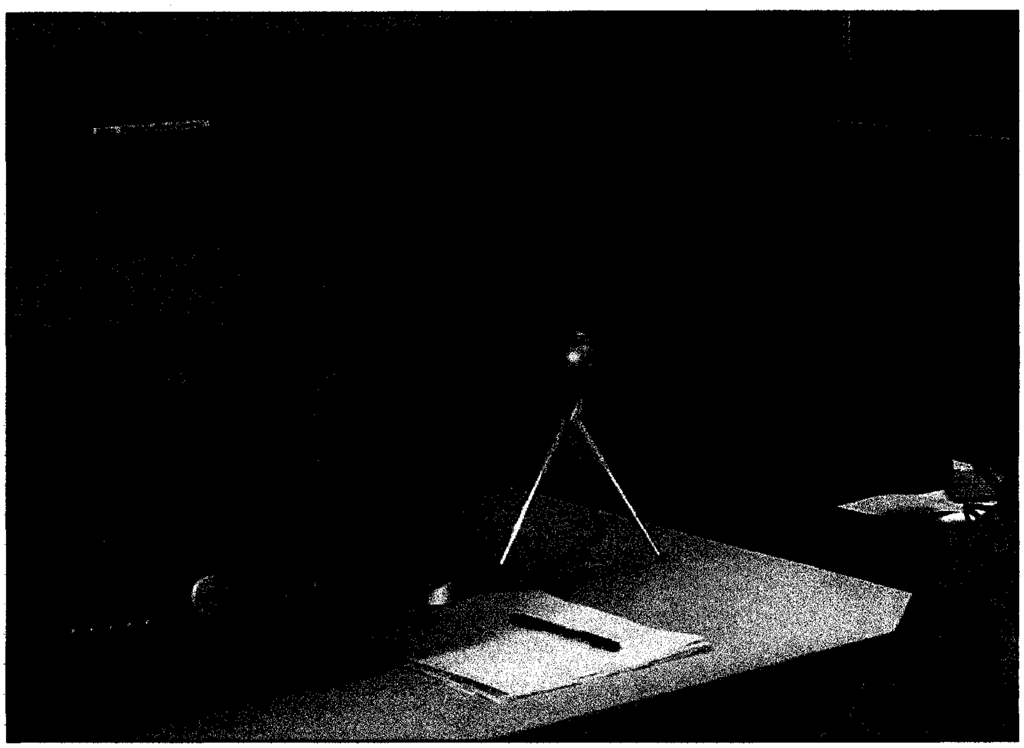


Figure 8: Recording PC and Webcam

We needed to know how large the Morae Recorder files would be. We wanted that size to be less than 650 MB, so that we could burn any recording onto a CD-ROM. We knew that a typical test session would normally last about 15 minutes, since Osmose automatically ended sessions after 15 minutes. However, we supposed that a session could potentially last up to 40 minutes (especially if we extended the recording to before and after the Osmose session was started, in order to record the process of putting on and taking off the helmet and vest). We ran through Morae's various recording quality options, and for each set of options recorded for 1 minute. Multiplying the resulting data file's size by 40 let us know approximately how large the file would be after 40 minutes. Every combination proved to be adequate, so we opted for the most portable one:

Morae's basic built-in full-motion video compression scheme at a 15 frame per second (fps) capture rate with 1024x768 resolution on the monitor, and 640x480 resolution on the webcam. We did find that we could get much better compression using other schemes, but we opted for portability since the better compression was not needed.

Another thing we wanted to enable was Morae's remote viewer capability. This would allow a data logger, located at the Logging PC, to embed annotations inside the Morae recordings being made on the Recording PC. To accomplish this we installed the software Morae Remote Viewer on the Logging PC, and then connected the Recording PC to the Logging PC via Ethernet (RJ-45) cables and a network switch.

As a final note, even though Osmose did play music, the music did not affect interaction with Osmose. Therefore, we judged that it was not necessary to record the music for the purpose of the usability test.

Throughout the process of setting up the usability test environment, we considered what sort of testing procedure would be required and how many people would be required to conduct each test. The details of the procedure and the roles and responsibilities of each tester evolved naturally as we built and tested the system. In addition, refinements continued during the first several live test sessions. The resulting roles and responsibilities can be found in **Section 9.2**, on page 88. The resulting test procedure can be found in **Section 9.3**, on page 90.

3.4 Analysis

During the usability test setup process, the major problem we encountered was getting the video footage of the participant, which was being displayed by Premiere, to be screen-captured by Morae. We originally tried to accomplish this on the Logging PC, but Morae did not function correctly. Morae could not capture Premiere's live video feed. In the video file, the feed would manifest itself as featureless grey or as a hole in the screen capture.

When we tried to do exactly the same thing on the Recording PC, that computer was able to record Premiere's live video feed—but only if we were logged in as the administrator. While this result was very mysterious at the time, a later investigation of TechSmith's customer support website revealed the probable source of this problem:

"Some systems have graphics hardware which can perform a 'hardware overlay' of video on the screen which bypasses the normal Windows display memory that [Morae] captures... The best solution is to disable hardware acceleration in the application that is playing the video, or disable hardware acceleration system wide [TechSmith, 2006]." Hence, most likely the Logging PC's graphics card used a hardware overlay to play the Premiere video, and when logged into the administrator account the Recording PC did not.

We ran into a minor problem with Morae Remote Viewer. By default, its *Include Audio and Picture-In-Picture* feature was turned on. Without this feature, Remote Viewer would only show the screen capture from the Recording PC. With this feature turned on, Remote Viewer also showed the webcam output (picture-in-picture) and outputted the participant's recorded audio. Leaving this feature turned on increased the network load and added a delay to the Remote Viewer connection with the Recording PC. This was very confusing for the data logger because it would desynchronize what was happening at one moment in the lab with the audio-video signal that the Logging PC was showing. Turning the feature off solved the problem. Since anyone sitting at the Logging PC could see monitor C (in ***Figure 6***) and hear the participant (this sometimes required the help of the headphones connected to the wireless microphone receiver), we did not need the *Include Audio and Picture-In-Picture* feature anyway.

3.5 *Findings and Recommendations*

3.5.1 Methodological Findings

Here are some of the important lessons on how to set up usability tests that we learned during our work:

- It is possible to set up a usability testing environment for an unconventional virtual reality system, using only low-cost, reusable, off-the-shelf components.
- It is important to begin preparing the testing environment in advance of the testing (two weeks was enough in our case). This gives time to work out bugs.
- Pre-testing, putting testers in place of the future participants, should be used to develop a testing procedure and to find faults in the testing environment. Don't assume that any component or, especially, any component combination is working correctly until it has been tested. This allowed us to catch problems, like Morae Recorder's failure to capture a live video feed and the synchronization problem with Morae Remote Viewer, before it was too late.
- Having clearly defined roles and responsibilities will help make sure that no element of the testing process is neglected.
- Have a defined procedure for conducting the usability test. This will ensure that the testers work smoothly together. However, the procedure should not be written in stone and should be open to opportunistic adaptation. Run through the procedure at least once in a pre-test to make sure that nothing has been forgotten and that the procedure runs smoothly.
- Once usability testing begins, regularly perform reviews of the research results. Doing so can help you optimize your test environment and methodology, and catch critical errors that were not apparent during pre-testing.

3.5.2 Conceptual Findings on P-Tab Requirements

Our experience with Osmose was sufficient to allow us to make a few preliminary conclusions regarding the P-Tab's requirements and design constraints. The requirements considered when building Osmose were fundamentally different from those we would need for the P-Tab. In spite of this, we were able to draw a few important requirements from our experience.

Osmose was designed to be portable, whereas this is not necessary for the P-Tab. As a result, rather than using bulky wearable devices (*e.g.* the HMD) we can use lightweight wearable devices and bulky *stationary* devices.

Osmose had the capacity to be immersive. This will probably be a useful trait for the P-Tab to have, although it is as-yet unclear how to accomplish this or what the benefits will be. It was clear that several elements of Osmose's construction made it less immersive. The breathing sensor constricted the participant's chest and made it hard to breathe. The helmet was very heavy and would typically hurt the participant's head or nose. Participants would often get tangled in wires. All of these elements contributed to making the experience less immersive, since each of these problems could remind participants that they were wearing devices and not truly inside the VR. Therefore, any implementation of the P-Tab will have to avoid using devices that might distract them from what they are doing.

3.6 Concluding Remarks

In this experience report we gave a detailed description of how we successfully set up a usability test environment and methodology for an HMD-based VR system. We customized existing techniques and technologies useful for desktop applications to our unconventional system under test. In so doing, we learned many important lessons on the practicalities of creating usability test environments for unconventional systems.

3.6.1 Methodological Conclusions

Today there is a need to develop novel usability testing environments and methodologies because computerized interactive systems are evolving. The new techniques will be necessary for testing a new generation of interactive systems, such as augmented reality, virtual reality and mobile devices. It is true that today's VR provides very limited tactile feedback, almost no proprioceptive feedback (as would be the case for walking on a sandy beach or on rough terrain), rare opportunities to smell, and little mobility [Jones, 2000]. However, in the coming years some or all of these limitations will disappear. As they disappear, usability labs will need to keep pace and continue to adapt their techniques. This experience report was a small contribution to this need.

It is our hope that our experiment will be useful to people performing usability tests on similar systems. The amount of detail supplied by this report should make it relatively easy to repeat our experiment. We provided the final result of our experimental procedure and our knowledge of the required roles and responsibilities. In particular, the low cost (perhaps $3 000 Canadian dollars, not including the cost of the computers) of our solution makes it more broadly applicable. Our materials were off-the-shelf and therefore reasonably easy to find.

We accomplished our goal of helping the HCSE research group to perform the first part of its three year Osmose study. After the testing environment was developed, it was successfully used over a period of three weeks to test 25 participants.

3.6.2 Conceptual Conclusions on P-Tab Requirements

With regard to our goal to elicit more requirements for the P-Tab, we did discover a few. The basic requirements for Osmose were fundamentally different from those for the P-Tab. We confirmed that the P-Tab would not need to be portable. We also learned the nature of the barriers that Osmose had that often prevented its users from becoming immersed in the VR, and thus if we wanted to implement the same thing we would know how to do it better. The main barriers to immersion were the heavy helmet, the breathing sensor that restricted breathing, and the wires that participants could tangle themselves in.

4 VRAD Experience Report

In this experience report we will describe how we observed a cave-based Virtual Reality (VR) installation developed by the VENISE research group at the University of Paris XI in Orsay, France.

4.1 Motivation

Our overall research goal was to better understand the Participatory Tangible Board (P-Tab) project's requirements. Since the P-Tab was only an abstract idea, we needed to observe concrete unconventional interactive systems in order to better understand the situation.

Through our research, we discovered the VENISE research group's cave-based VR installation. We accepted VENISE's invitation to visit in September 2005 and observe the installation. This visit would afford us the opportunity to see a cave-based VR installation (one taking place in a room with the VR displayed on the walls), thus complementing our previous experience with a Head-Mounted Display (HMD)-based VR system.

In addition, one of the Ph.D. students performing research on the VENISE VR installation had written a Computer-Aided Design (CAD) program for it, as well as a creative Sculpting program. In visiting Paris we would get the unparalleled opportunity to observe how design and creativity might be affected by VR.

4.2 Goals

We had the following research goals:

- To observe first-hand the operation of a cave-based VR installation.
- To observe first-hand how VR affects design and creativity.
- To learn what components went into the VR installation's construction.
- To describe our experience in detail for the benefit of others.

4.3 *Analysis*

Virtual Reality Aided Design (VRAD) [Convard, 2005] was a research project developed by Thomas Convard at the University of Paris XI in Orsay, France. The purpose of the project was to explore how to enhance CAD with VR. Two sub-goals included to see if an effective user-interface could be created without needing any wires attached to the user or user-interface widgets, and to see if virtual objects could be made to interact with each other in meaningful ways. Thomas wrote two programs as part of VRAD: a Design program and a Sculpting program.

4.3.1 The VENISE Virtual Reality Installation

The VENISE VR installation was composed of a pair of rear-projection screens forming a V-shape (see **Figure 9**), with a stereo Cathode-Ray Tube (CRT) projector (see **Figure 10**) putting an image on each of the screens. The images were generated by a network cluster of four Intel Pentium IV Xeon PCs with nVidia Quatro 4 graphics cards (see **Figure 11**).

The participant experiencing the virtual reality would stand inside the V and put on one or more pieces of equipment, including: active stereo shutterglass glasses, a microphone and a wireless mouse (see **Figure 12** and **Figure 13**). A series of infrared cameras, aided by small reflective globes attached to the wearable equipment, detected the position and orientation of the equipment. The glasses complemented the stereo projectors and allowed the wearer to see a perfect perspective of a virtual reality world when head-tracking was enabled. Wearers could move around, tilt their heads, look underneath objects, look all around objects, etc. The effect was so realistic that it was easy for participants to forget that the projection screens existed—they would often try to push through them with the mouse. The microphone allowed voice input, while the mouse allowed interaction with the virtual environment.

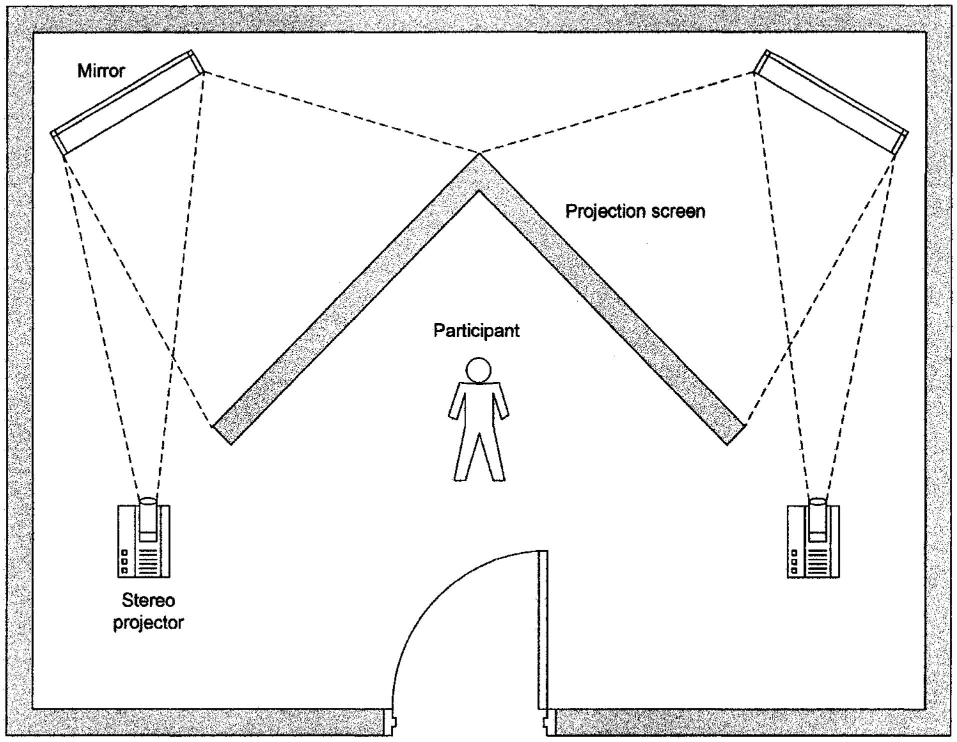


Figure 9: VENISE Virtual Reality Installation

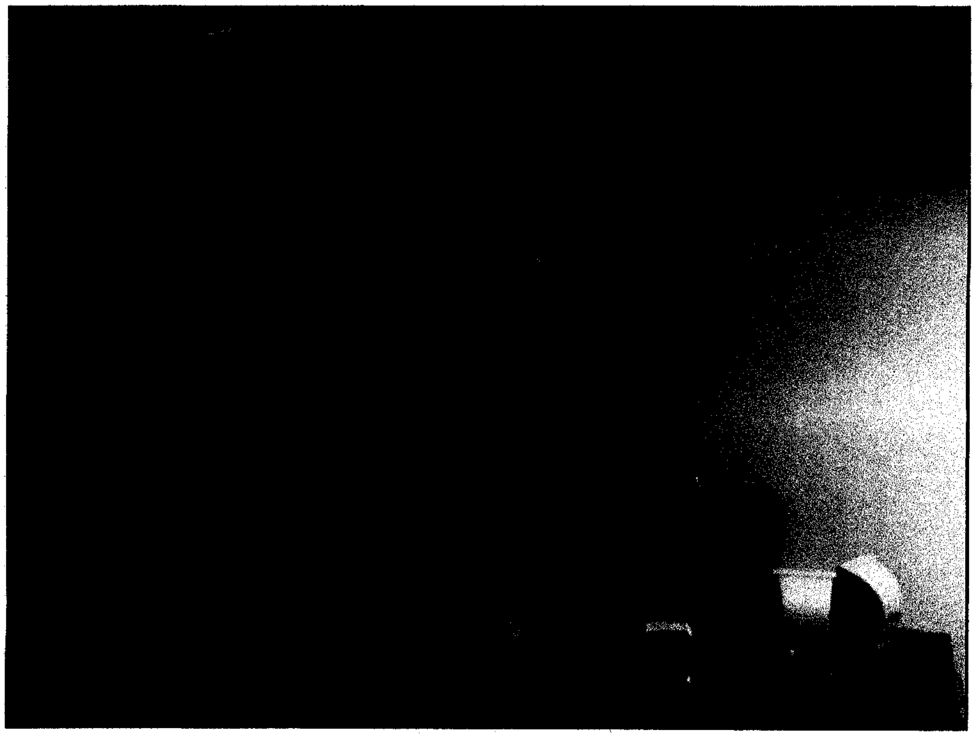


Figure 10: Stereo CRT projector

Figure 11: Graphics processing cluster

Figure 12: Equipment used for VRAD

Figure 13: Wireless mouse with infrared-reflective globes

4.3.2 The VRAD Design Program

The Design program created as part of the VRAD project was used to design geometric shapes. Using a wireless mouse to interact with the virtual environment, and voice input in order to select new shapes or tools, the participant would create new shapes in 3D. For example, the user could create points, lines, curves, spheres, cubes, etc.

By combining different shapes more complex designs could be created. One shape could even be subtracted from another one in order to create complex concave shapes more easily (see **Figure 14**).

The voice input could only recognize one person's voice, and even then users sometimes had to repeat themselves in order for the system to recognize a command. However, this was because the voice recognition software used for the VENISE prototype installation was fairly cheap. A commercial system would have to employ better software.

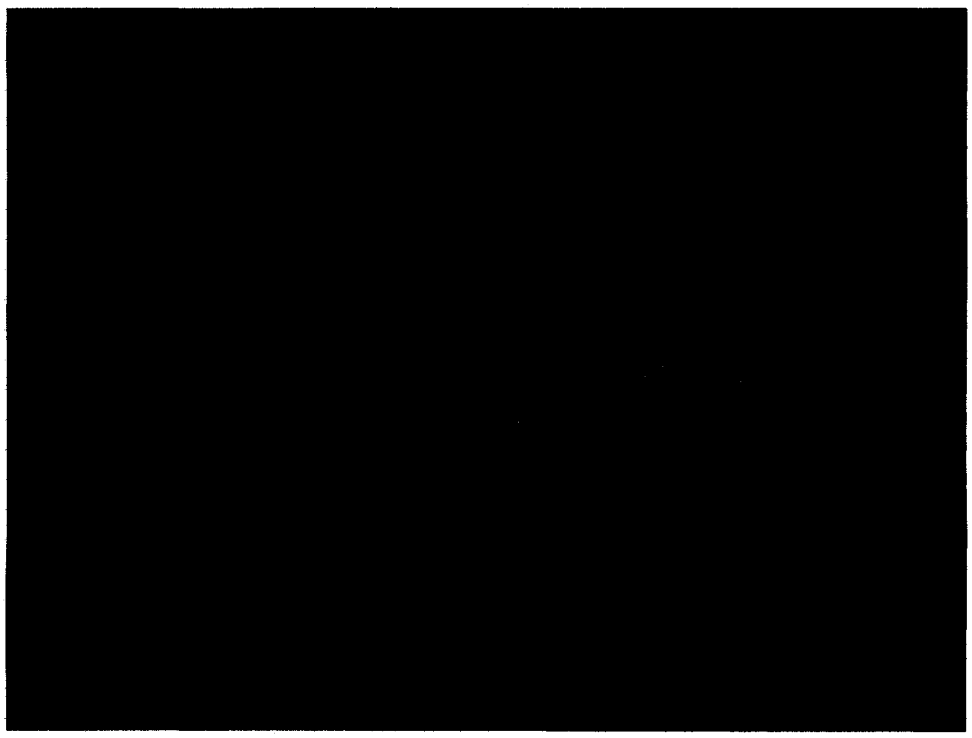


Figure 14: The VRAD Design program

4.3.3 The VRAD Sculpting Program

VRAD's Sculpting program was more creativity-oriented than design-oriented. Its purpose was to enable participants to sketch in 3D. The participant's mouse pointer was a 3D sphere. By clicking the wireless mouse, the user would fill in the area of the sphere with grey-coloured matter. Hence, the participant could create objects by clicking the mouse and moving it around. Surprisingly complex shapes could be created very quickly, although they might not look very beautiful (see **Figure 15**). By rolling the mouse wheel the user could change the diameter of the sphere. Clicking a secondary mouse button would erase matter instead of creating it. Using voice commands users could change the system into a colouring mode and select a colour. During colouring mode, clicking the mouse would colour any matter within the mouse pointer sphere instead of creating new matter.

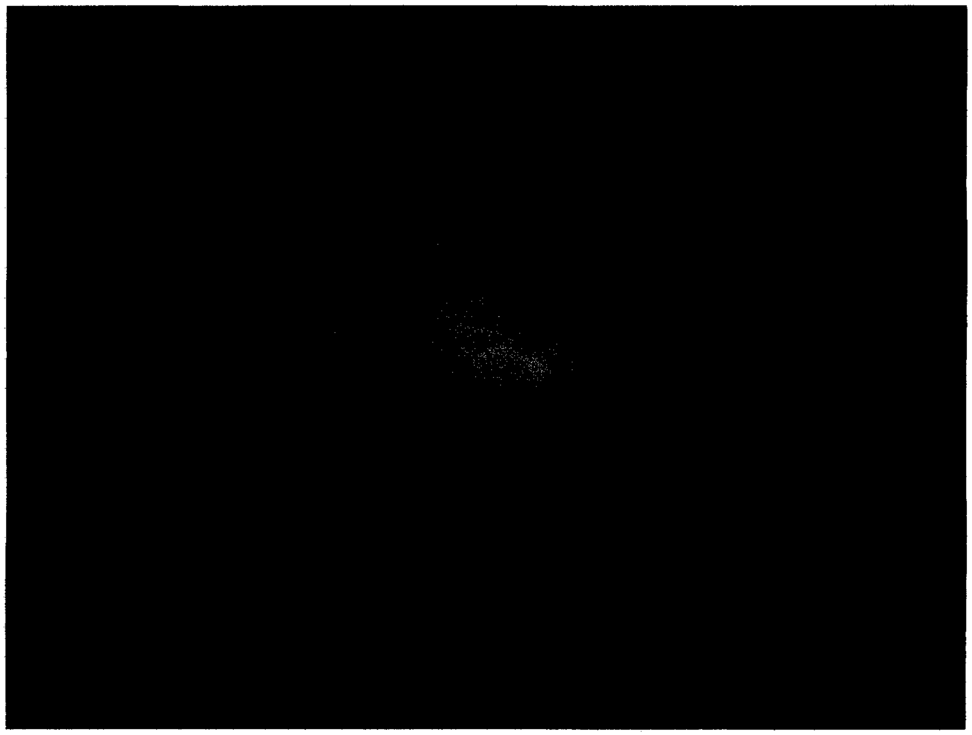


Figure 15: The VRAD Sculpting program

4.4 Findings and Recommendations

By using only light-weight, wireless equipment and voice input for interacting with the VR, the VRAD project freed its users from the encumbrance of consciously interacting with a system. The result was a very immersive environment, in which it was easy for users to forget that they were dealing with a *virtual* reality. In particular, the lack of any sort of visual user-interface widgets intruding in the VR made working with the VRAD programs a very immersive experience.

Removing user interface widgets has many advantages, but it has its drawbacks too. It's likely that only fairly simple and straightforward user interfaces can be modeled in this way. One could simply increase the number of voice commands in order to provide a richer user interface, however, this would come at the cost of the user having to remember all of these commands. Furthermore, it would be difficult for the user to be reminded of what commands exist without a user interface widget intruding into the VR to deliver this information.

In fact, this widget-less technique is reminiscent of command line interfaces, where users must be experts to know what all of the commands are. The more commands exist, the harder it is to know which commands are the most useful and which commands you need for a particular task. Hence, a complex VR CAD program might have a very high learning curve and require a lot of documentation review.

The advantages of designing a 3D object in a 3D VR environment are great. Users are able to see the object from all sides, and in a good enough installation such as the VENISE one, the simulation is able to trick the brain into thinking the object is real enough to touch. Such perfect visualization of a product under construction surely helps make designing it much easier.

For designing non-physical systems, such as software, the advantage of using a VR is less clear. Software is a mathematical construct that can't be directly visualized. Hence, one wonders what exactly a VR would display. It seems that VR would suffer from the same problem as contemporary Integrated Development Environments (IDEs)—it is very difficult to create visual interface metaphors for something that is purely abstract.

4.4.1 Regarding Creativity

An important question is whether or not a VR system like VRAD helps the designer's creativity. This question is important because it impacts a designer's performance and effectiveness. There is no doubt that designing objects in an immersive VR is very fun. This may be because of the relative novelty and rarity of such systems, or it may be intrinsic. Either way, the question remains: does this help creativity?

Immersive and fun environments engage users' interest and result in a longer and more intense experience. While periods of great creativity are often described as being immersive, the reverse is not necessarily true [Csikszentmihalyi, 1997]. Hence, we can say that an immersive experience has a positive impact on creativity. However, how much of an impact is open to debate.

The following quotation is written regarding how CAD tools help or harm creativity in architecture:

"Present user interfaces can make draughting and the creation of alternatives quicker and more effective in the final stages of designing… User interfaces are at present inflexible in sketching. Draughting and sketching are the basic methods of creative work for architects. When working with the mouse, keyboard and screen the natural communication channel is impaired, since there is only a weak connection between the hand and the line being drawn on the screen… and the important items cannot be emphasized by, for example, pressing the pencil more heavily than normally. In traditional sketching the pen is a natural extension of the hand, as sketching can sometimes be controlled entirely by the unconscious. Conscious efforts in using the computer shift the attention away from the actual design process. However, some architects have reached a sufficiently high level of skill in the use of computer applications in order to be able to use them effectively in designing without any harmful effect on the creative process [Haapasalo, 2000]."

We can adapt what Haapasalo wrote to draw the following preliminary conclusions regarding design tools:

- Allowing users to quickly create and evaluate design alternatives helps creativity.
- Allowing sketching helps creativity.
- Making sketching natural (*i.e.* requiring no additional skill) helps creativity.
- Allowing users to emphasize important elements of the design (*e.g.* by circling the best solution, underlining, writing harder, or writing annotations) helps creativity.

4.4.2 P-Tab Requirements

Based on our findings, we can make the following suggestions on how to build the P-Tab:

- The P-Tab must support natural sketching that requires no additional skills to use, and allows designers to emphasize important elements of the sketch.

- The P-Tab must help designers rapidly create and evaluate design alternatives.
- At least some of the P-Tab's interfaces should be widgetless, in order to improve the environment's immersiveness and designers' creativity.
- If the P-Tab uses wearable or hand-held equipment, that equipment should be lightweight and wireless.

4.5 Concluding Remarks

In this experience report we described in detail our experience with the VENISE VR installation located in Orsay, France. We gave a detailed account of our experience with the Design and Sculpting software written for VENISE as part of the VRAD project.

We learned how the VENISE installation was constructed, and observed first-hand its operation and the operation of the VRAD prototype's Design and Sculpting software. Drawing upon this experience, we were able to make some preliminary conclusions about VR's impact on design and creativity.

We determined that VRAD's use of lightweight, wireless, wearable devices coupled with voice-based interaction instead of widget-based interaction made it a very immersive experience. We concluded that VRAD improved a designer's creativity by supporting various requirements like providing natural sketching capability.

This learning experience helped us elicit more requirements for the P-Tab, and helped us understand some ways that these requirements could be met.

5 P-Tab December Field Study

In this experience report we will describe the first of two field studies that we performed in order to learn more about the Participatory Tangible Board (P-Tab)'s requirements. The first study took place during the month of December, 2005.

5.1 Motivation

The P-Tab was an immense project with vague requirements. We had many questions regarding how the P-Tab might be implemented, what its design constraints were, how broad its scope might be, who might use it, where and under what circumstances.

We planned to conduct a *field study* to help answer these questions. In a field study the researchers go to the "representative users' workplace and talk to them, observe them work, and ask them questions, to understand the user characteristics, the work flow, and the system features they need [Nielsen, 1993]."

However, we had many questions on how to conduct the field study. As a result of this uncertainty we planned two studies. The first December 2005 study would primarily serve the purpose of gaining a better understanding of how to perform the study well, what to look for and how to record that information. We hoped to also learn as much as we could about the P-Tab as well. Next, in the second February 2006 study, we planned to apply all of the lessons learned from the first study and consequently perform a more effective study that would provide us with more solid information on the P-Tab's requirements.

Here are some of the questions we asked ourselves on the nature of design and regarding how to conduct the December study:

- What design tools do designers need? Hence, what tools should we provide for the test?
- How does a multi-disciplinary team communicate design concepts?
- How do designers represent their designs?
- What tasks do designers perform in the earliest stages of design?

- What and how many disciplines are involved in a multi-disciplinary design session?
- How many participants are involved in a design session?
- How long should a session be?
- What is a good design problem to use as a test?
- Should we introduce the design problem to the participants *before* they arrive to the test session, or *after* they arrive?
- Should we add or remove participants to/from the test, in order to change the number and discipline type of participants in the working group? This would change the group's social dynamics.

5.2 Goals

Our goals during the December study were as follows:

- To gain practical experience at performing field studies, and in so doing to learn how best to gather the information we needed.
- To answer the questions we asked ourselves on the nature of design and how to conduct a field study.
- To elicit some requirements for the P-Tab, and gain a better understanding of the project scope, deployment context and design constraints.

5.3 Methods, Test Plan and Infrastructure

Since we did not know the answers to any of our questions, we planned a very broad study. We made some assumptions on many of the questions, and then planned to learn if our assumptions were correct through the study results.

It took 20 working days to prepare the study, spread over two and a half months.

5.3.1 Initial Test Plan

Here are the major decisions and assumptions that we made:

- We decided to write pre-test and post-test questionnaires for the test participants to answer. The questionnaires would inquire about the nature of design, and the correctness of some of our decisions and assumptions. We also decided to perform a *field observation*. A field observation is a sub-part of a field study in which the testers observe the participants as they work, to understand how the participants are using a system to accomplish their tasks and what kind of mental model they have about the system [Nielsen, 1993]. The "system" in our case was software design.
- We were not creating a custom device for a particular company. As a result, there was no workplace for us to go to in order to observe the P-Tab's representative users. Instead, we decided to create an artificial design environment (in Concordia's Usability and Empirical Studies lab) and invite representative users to participate in the study there.
- Since in the December study we wanted to observe the earliest stages of design, we decided not to provide the participants with the design problem before the study began.
- We chose a computer game as a design problem, because it encompassed all three of the P-Tab's fields of interest (user-interface, art and software design) and would be more fun for the participants and more likely to keep them awake. We chose to make it a cellphone game to keep the design scope limited, since cellphones have very limited capabilities.
- We arbitrarily set the number of participants at 6. This was more than two people (and hence a group of people), and near the maximum we could fit into the usability lab at one time.
- We wanted 2 people from each of 3 design disciplines: art design, software design and game design (the latter of which is a lot like user interface design, but more fun-oriented and managerial).

- We chose a time period of 4 hours, because this was the largest amount of time a group of people could be reasonably expected to work without needing to go to the washroom or eat a meal.
- We planned for the study to take place on 2005-Dec-17 from 13:00 to 17:30.
- We wanted the study to occur at Concordia University's Engineering, Computer Science and Visual Arts complex (the EV building) in Concordia's Usability and Empirical Studies lab, room EV-9.210. We would use our research group's lounge and guest room, room EV-9.189, to greet the participants upon arrival and to house any participants that were taking a break.
- We wanted to see the impact of mixing up the social dynamics of the group by splitting them up based on their design specialty.
- Participants were to arrive at 13:00 in room EV-9.189. We would give the participants a grace period of 30 minutes to arrive and fill out the pre-test questionnaire and sign their consent form. The testing would then take place from 13:30 to 17:00 in four phases. Phases 1 and 4 would take one hour each and involve all participants. Phases 2 and 3 would take 45 minutes each and would only include either an art team or a software team—with the game designers splitting between the two teams. This would leave 30 minutes leeway for transitions between phases, and for post-test questions. **Table 3** illustrates our timeline plan.

Table 3: Timing of December Study

<table>
<tr><td colspan="6">13:00 13:30 14:30 15:15 16:00 17:00 17:30</td></tr>
<tr><td>Consent & questionnaire</td><td>Art & software designers</td><td>Art designers only</td><td>Software designers only</td><td>Art & software designers</td><td>Post-test questions</td></tr>
</table>

5.3.2 Study Setup

We configured the Usability and Empirical Studies lab for the purpose of conducting the December study. The end result can be seen in **Figure 16**.

We set up a product called Morae Recorder (by the company TechSmith) on the test computer (labelled as "PC" in **Figure 16**), and attached a camcorder to it. Morae Recorder would allow us to record user actions on the computer and record the participants' actions and conversations using the camcorder. For sound, we connected a wireless microphone to the camcorder.

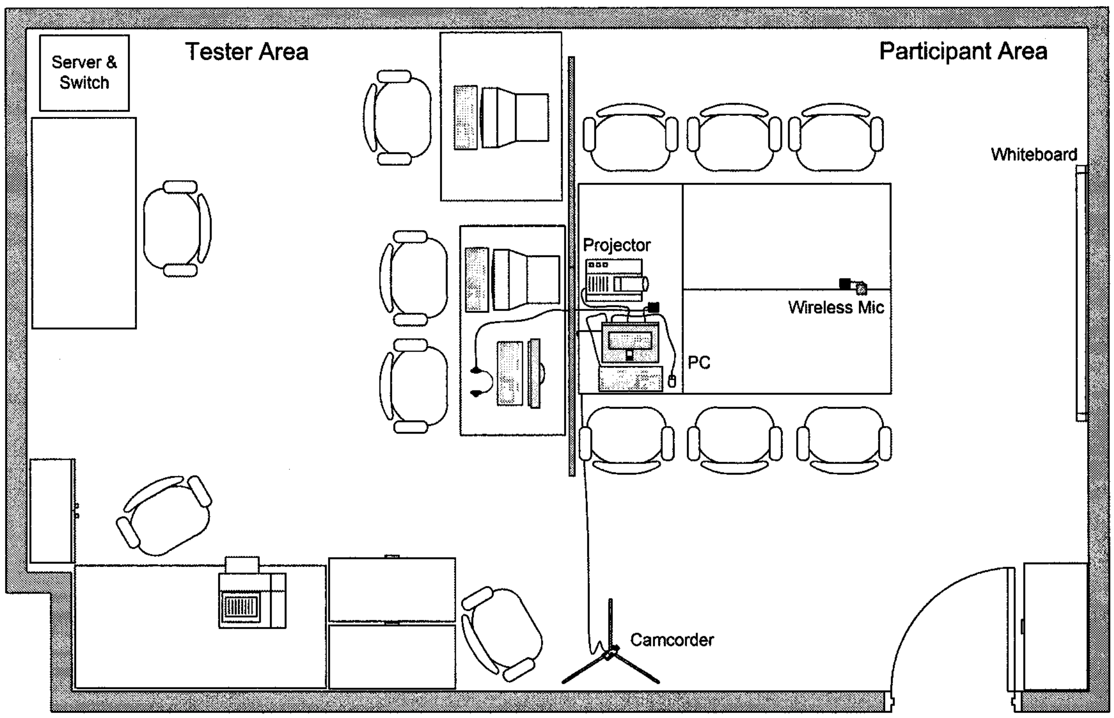


Figure 16: December Field Observation Setup

A 1.5 m tall partition separated the tester area from the participant area. This hid the testers from the participants' view. However, the camcorder was visible to the participants, as would be anyone operating the camcorder. In addition, since the room had only one door, any testers involved would need to come in and out that one door, in full view of the participants.

We wrote several pieces of documentation, including:

- A consent form for the participants to sign.
- Roles & responsibilities (**Section 10.1.1**, p.92) and procedure (**Section 10.1.2**, p.93) documents to help organize the testers during the testing.

- A design problem for the participants to attempt to solve (**Section 10.1.3**, p.95).
- Post-test questions to ask the participants (**Section 10.1.4**, p.96).

We made the following design tools available to the designers:

- A computer with a projector attached to it (to project what was on the screen).
- The computer had several pieces of useful software installed: Microsoft Word, Microsoft Visio, Adobe Photoshop, SMART Ideas, Calculator, and Notepad.
- A whiteboard with dry-erase markers and eraser.
- We originally wanted the whiteboard to be instrumented with a product called Mimio, which with the help of the projector would allow the whiteboard to act as an interactive smart board. However, the ultrasonic noise in the laboratory was too great for Mimio to function. Despite our best efforts to block the motion detectors that were creating the ultrasonic noise, we could not get Mimio to function. Hence, it was removed from the testing.
- A large set of conventional design tools, such as pens, pencils, paper, scissors, glue, rulers, etc.
- We attempted to enable Internet on the computer, but were not able to. The reason why was that we had decided to turn off the network switch and server (located in the same room) because they were creating excessive noise that was making it difficult to record conversations.
- We wanted to install Adobe Illustrator and Adobe InDesign on the computer, but our research group's installation CDs were missing at the time and we could not.

5.4 Population and Selection Method

During the time spent preparing the study, we were busy trying to find participants. We wrote a *call for participation* and posted it on the Human Centered Software Engineering (HCSE) group website. We offered to compensate participants with a $30 Canadian dollar reward. We sent the call to the International Game Developer's Association (IGDA) web forum, and the IGDA Montreal chapter's mailing list. We sent the call to several Concordia computer science and art student mailing lists. We printed out the call and posted it around Concordia's EV building. We also communicated with several personal contacts.

We received responses from 13 people: 8 art designers, 4 software designers and 1 game designer. Three respondents were female and ten were male. With only 1 game designer, we decided that we would recruit a third software designer to replace the missing game designer—because we believed that the software design aspect of the test project would be more difficult than the art design aspect and could benefit from an extra designer.

We non-randomly selected the participants for the study. We selected for the ability to make a firm commitment to the study date and time, and for maximum amount of design experience. We were able to get firm commitments to participate from 3 software designers, 2 art designers and 1 game designer. On the day of the study one of the software designers did not arrive. The remaining participants are listed below, with their names changed to protect their identities:

- Two art designers: Adrian and Brian.
- Two software designers: Ed and George.
- One game designer: Jerry.

5.5 Data Collection and Analysis

We collected data from our study by reviewing the recorded video footage and taking notes on relevant data. For the data resulting from the field observation, see **Section 10.2** on p.97. We focused on what the participants were doing and for how long and how they were arranging themselves as a group. For the data resulting from the post test questions, see **Section 10.3** on p.98. In our notes we summarized the participants' answers.

5.5.1 Review of Field Observation Results

During the test we were helped by some fellow HCSE group members (see the **Acknowledgements Section**, page *iv*). Their help was greatly appreciated, although as it turns out their help wasn't necessary—there wasn't enough work for more than two people. Furthermore, it seems like their help might actually have hindered the study thanks to the way that the lab was set up.

The usability lab was not equipped with a back door. Hence, every time one of the testers needed to leave the room, they had to pass in front of the test participants. The participants had only met and exchanged names with two of the testers, but not with the other testers. Having so many strangers moving around them might have made the participants nervous. They would often stop talking when strangers were nearby, and at one point Jerry even said "I'm nervous!"

One of the first things we noticed as we were reviewing the camera footage was that the camera did not record the events very well. An unfortunate choice in camera setting washed out the background in white, making it impossible to see what people were doing at the table or writing on the whiteboard. We were able to see the participants and where they were, but not much else. Fortunately, the sound quality was high.

Splitting the team during the different phases of the study varied the number of designers working at one time, and consequently allowed us to see the variety of ways that participants would sit together.

We noticed a marked difference in group behaviour between groups of 2 or 3 people, and groups of 4 people or more. Groups of 2 or 3 people were able to all work together on one task, while groups of 4 or more seemed much less effective at a single task.

Furthermore, we observed that the design team members seemed to communicate their design concepts in two ways: with words and with sketches. This made it clear that non-active participants had two basic requirements that allowed them to keep track of what was going on: being close enough to hear what active participants were saying, and being able to see what active participants were sketching.

We noticed that during or after a design session, transferring the data that had been placed on a non-computerized medium (*e.g.* paper or a whiteboard) required manual recopying to another medium in order for it to be reused. This manual recopying took additional time. For example, the designers re-wrote the essential points of their design into Microsoft Word.

We observed that the participants were able to do design work, but that they weren't very productive. The end result of their design session was a four-page point-form Microsoft Word document and a few sketches.

During the post-test questions, the participants mentioned several times the requirement for a design team to split up into a collection of individuals, and then join together again as a group to discuss results. They themselves never split up during the field observation unless we specified that they had to.

One other thing we noticed was that the participants complained about the lack of availability of certain tools that *were* in fact available (*e.g.* Photoshop and Visio). Also, they never used the brainstorming tool SMART Ideas.

Sketches seemed to be really important as they were used to help communicate concepts, but we never saw any formal notations or diagrams.

5.6 Findings and Recommendations

5.6.1 Methodological Findings

As it turned out, it was fairly easy to find art and software designers (because they were readily available in the EV building environment), but game designers were busy professionals and difficult to find.

An important element in attracting participants was the fact that we offered to pay them. We offered a gift of a USB memory key/pen worth $35 Canadian dollars, which was reasonable compensation for 4 hours of work.

With regard to our initial questions on getting the correct number of participants, the correct number and types of design disciplines, and the correct length of design session, what we did provided results. Five people from three different disciplines seemed sufficient. In their professional environment game designers tend to manage art designers and software designers, hence having only one game designer and several members of the other disciplines seemed to be a natural fit. During the post-test questions, the participants themselves mentioned that there was a good mix of people and largely enough time for a first session, but that more sessions would be required, and more people in following sessions.

The design tools that we provided to the designers seemed reasonably useful since the designers were able to work. It is difficult to say for certain if the tools were adequate because the participants didn't use any tools we didn't provide, and they didn't ask for any tools that don't exist yet.

It was interesting that during the post-test questions the participants complained that several useful tools were missing from the repertoire we provided them with—even though some of those tools were in fact present. In addition, the participants never used SMART Ideas, which is ostensibly very useful but also not widely known about. From these pieces of evidence we concluded that in the February study we would have to do a better job of advertising the design tools that we were making available.

Introducing the design problem at the design session, and not before, did successfully allow us to observe the design process right from the beginning.

In the end, the designers produced very little actual design content. Besides a few sketches, they produced a four-page point form document. Taking into account the different phases and number of people in each phase, there were 13.75 person-hours of work done, which produced 4 pages of legible material. That's about 0.3 pages per person-hour. Some possible reasons why the output was so low include:

- The designers were spending a lot of time trying to understand the problem.
- This was the first design session and they are intrinsically less productive.
- The designers were nervous because of all the strangers observing them.
- When working together (2 out of 3.5 hours), the participants used their tools ineffectively.
- The tools made available to the designers were insufficient and/or ineffective.

- The design problem was poorly chosen.

At this point it was hard to answer the question of how designers represent their designs, in part due to the small size of the study and also because of the small amount of results produced by the participants. We felt that design diagrams must exist in the design vocabulary even though we didn't see any. It was possible that the designers were still at too early a stage in the design to need any formal notations or diagrams.

During the post-test questions the participants mentioned that it was important to split up the team into sub-groups in order for the work to be more effective. Since the team was not very effective during the December study, and we wanted them to be effective so that we could obtain more results, we decided to take their advice. Therefore, we decided that in the February study we would encourage the design team to split up without creating formal phases.

5.6.2 Conceptual Findings on P-Tab Requirements

The variety of different positions taken as a group by the designers caused us to ask ourselves some new questions:

- Which positions allow the greatest number of active participants (*i.e.* people actively writing or contributing, not passively watching or listening)?
- Which positions offer the best visibility of design artefacts and decisions to non-active participants?
- How does the variance in number of people in a group affect positions?
- How do tools affect how participants position themselves in a work area?

We wanted to understand why the effectiveness difference between 3 and 4 designers was so great. By reviewing our results, we could see that it was a question of visibility. On a horizontal surface, up to 3 people could crowd around a piece of paper such that they could all see it well and all speak to each other easily. On a larger table such as the one we used for the test, the fourth and later participants were not able to see the paper very well. Even with a smaller table the fourth and later people would be looking at the paper from an awkward angle, thus reducing visibility.

It was only when the sketches were placed on a vertical surface that all of the participants could see them well. Thus, the projector and the whiteboard allowed groups of 4 or more people to all take part in the discussion. A major difference between the whiteboard and the projector was that up to 3 people could simultaneously write on the whiteboard, whereas only one person could work on the computer at a time.

Thanks to our experimental results, we can begin to answer the question of what impacts the effectiveness of a design session and how.

Figure 17 through **Figure 20** show the characteristics of the various environments we made available to the participants during the field observation, including a table, whiteboard and projector. The numbers and values given for each diagram reflect what we observed.

Here follows a brief explanation of the terms used in the diagrams:

Max. Participants: the greatest number of participants that can write or draw with relation to the same task at the same time.

Communication: measures the average quality of the communication that is possible between all of the participants. This is a subjective measure based on the perceived effectiveness of the environment. Rated as *high* or *low*.

Visibility to Observers: measures how well non-active participants can see design artefacts. Observers can still take part in a discussion, but are unable to write or draw anything. Rated as *high* or *low*.

Recopying Required: this has the value *yes* for manual non-computerized systems like pen and paper or a whiteboard, and *no* for computerized systems. If recopying is not required, this indicates that the environment saves time for the designers.

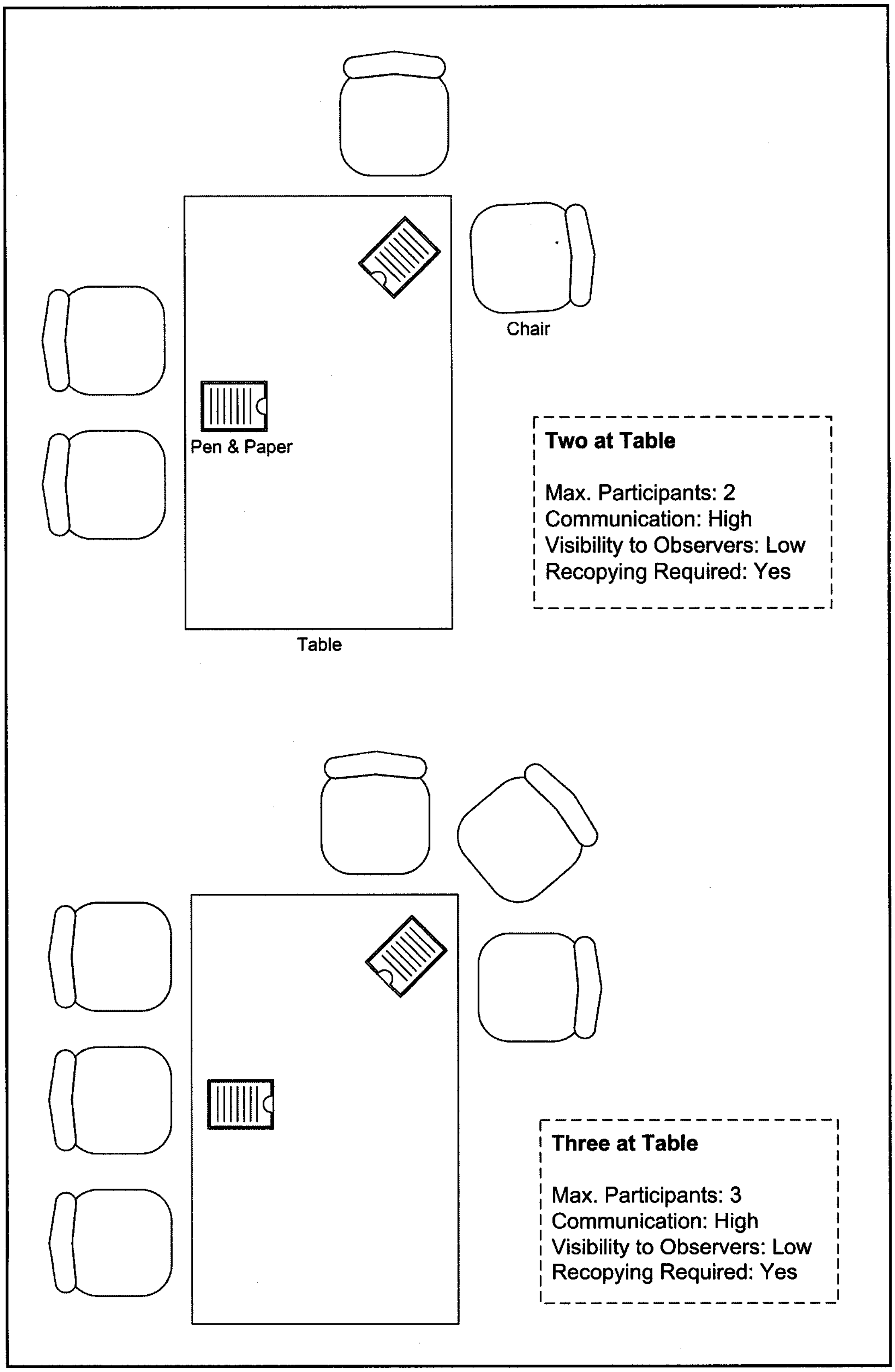


Figure 17: Two to Three at Table

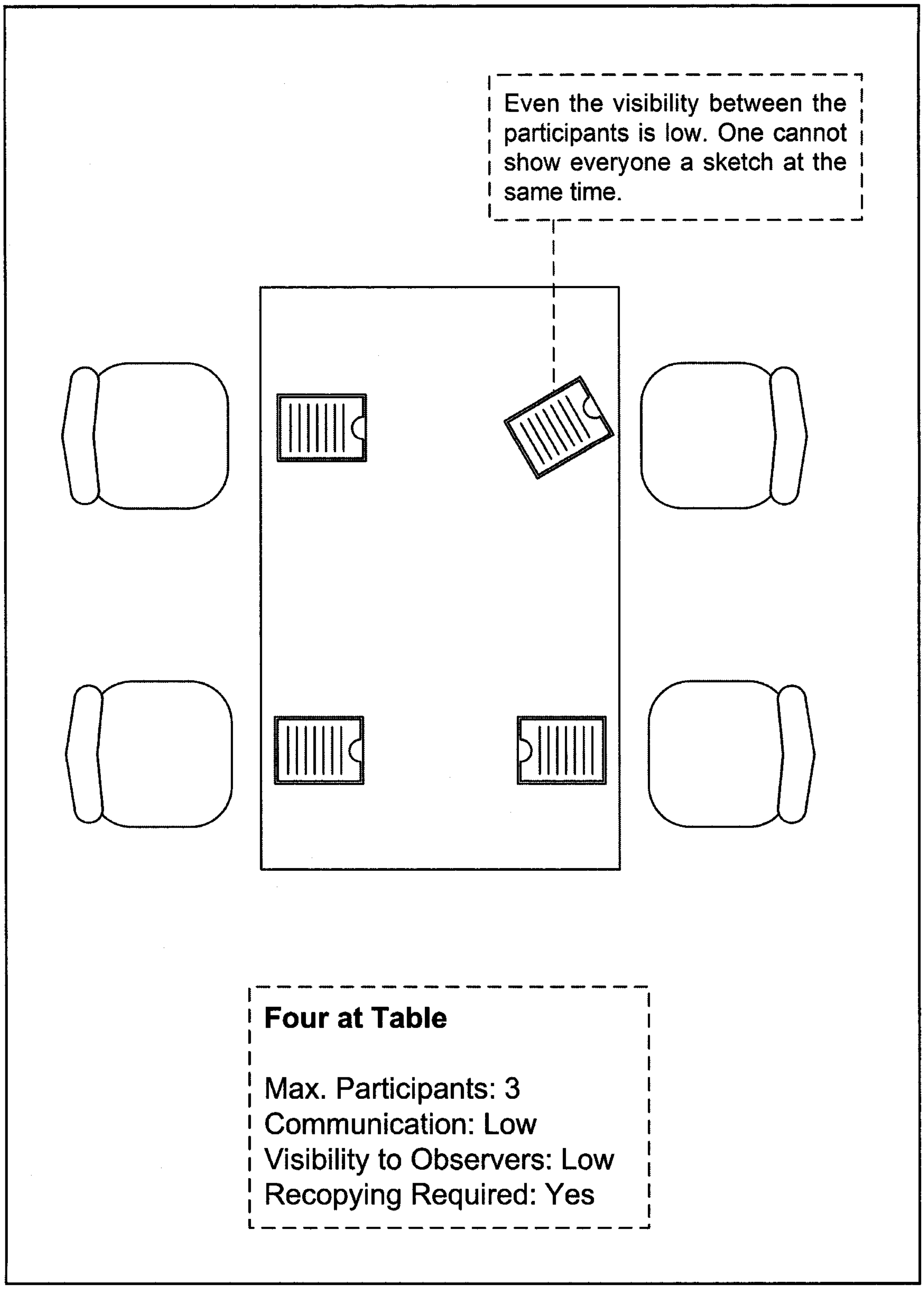


Figure 18: Four at Table

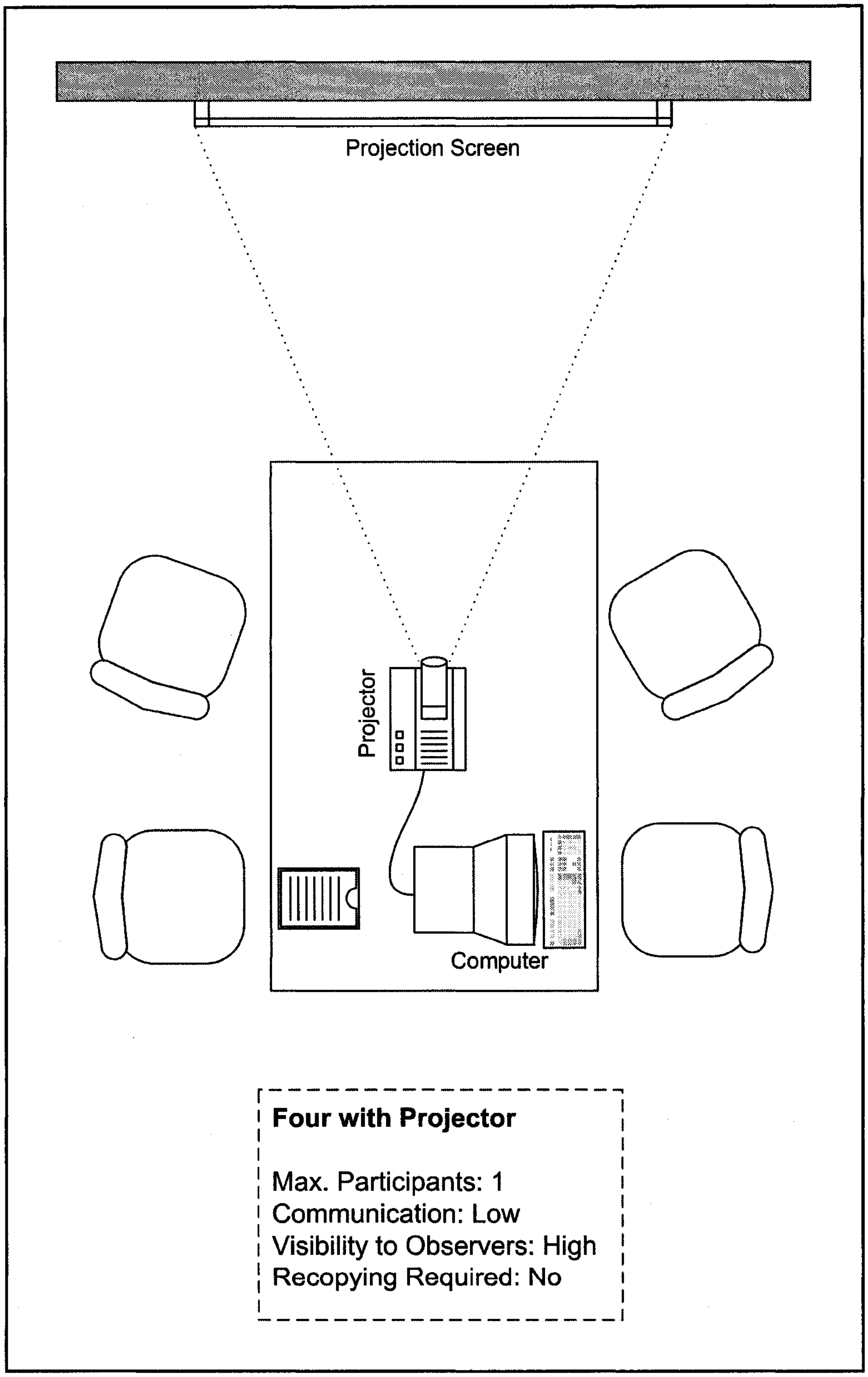


Figure 19: Four with Projector

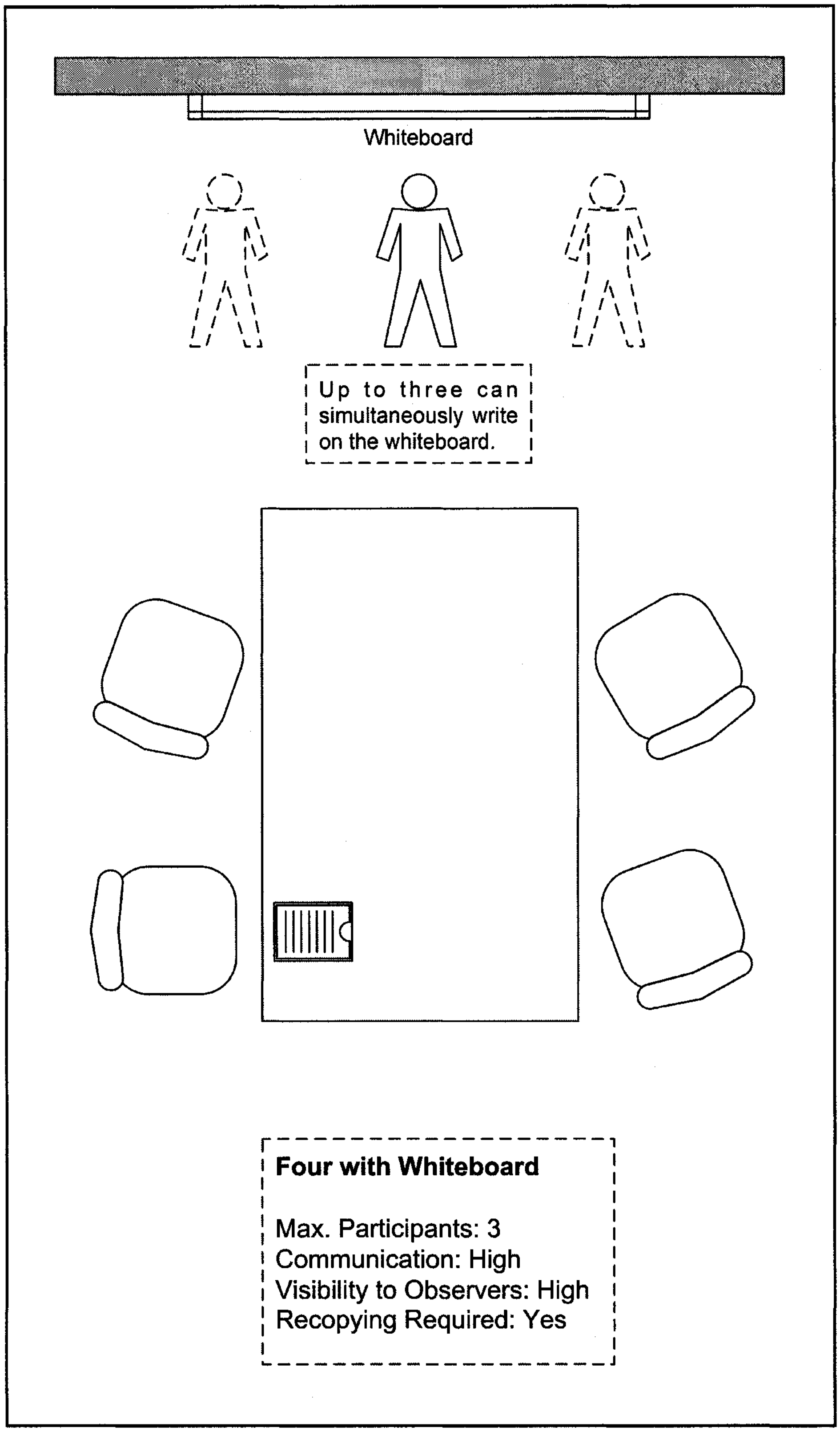


Figure 20: Four with Whiteboard

Based on our findings, we can make the following additional observations:

- The better the communication between a design session's participants, the more participants can be actively involved.
- The more active participants involved in a design session without getting in each other's way, the more effective the session will be.
- The greater the visibility of a design session's active participants' sketches and discussions, the more effective the session.
- The less time spent recopying information, the more effective the session.

Based on these observations and our experimental results, we can propose the following hypothesis, to be confirmed in future research:

- A flat table is only an effective design environment for 2 or 3 designers. A vertical surface is effective for design teams of 4 or more people.

5.7 Concluding Remarks

In this experience report we described the December field study that we performed in order to learn more about the P-Tab's requirements. In the end, we met most of our goals for the study.

5.7.1 Methodological Conclusions

We successfully gained new practical experience on performing field studies. For example, we learned that we had to change our recording technique to make it more effective. We failed to do a good job of recording what the designers were doing. However, thanks to our increased practical knowledge, we hoped to overcome this problem and do a better job in the next study.

We learned many important lessons regarding how to perform field studies, including:

- In the February study we would need to have fewer testers in order to disturb the participants less.

- In the next study we would need to advertise the tools we had made available to the designers so that they could make use of all of them if desired.
- In the next study we would need to encourage the participants to split up into sub-teams on their own, so that we could observe the effect on how they positioned themselves and how they worked.
- In the next study we would need to better record what the participants were doing and writing in order to be able to make more observations of the design process.

5.7.2 Conceptual Conclusions on P-Tab Requirements

We did not meet our goal to learn more about the nature of design, for the reason that we failed to properly record what the participants were writing. Our inability to see the participants' sketches as they were drawing them and discussing them made it much harder to understand what they were doing and how they were thinking.

In spite of this problem, we successfully elicited some new requirements for the P-Tab by uncovering the surprisingly large difference between 3 and 4 participants, and the resulting need for a vertical work surface.

Here is a list of what we learned regarding the P-Tab's requirements:

- There was a great deal of difference in how participants seated themselves and how they worked between groups of 3 or less and groups of 4 or more.
- Horizontal surfaces seemed to be effective for teams of 2 or 3 participants, but for teams of 4 or more a vertical surface was required for maximum effectiveness.

6 P-Tab February Field Study

In this experience report we will describe the second of two field studies that we performed in order to learn more about the Participatory Tangible Board (P-Tab)'s requirements. The second study took place during the month of February, 2006.

6.1 Motivation

In a field study, researchers go to the representative users' workplace and observe them in their normal work environment (a field observation), and ask them questions about what they do, how they do it, what software functionalities they need, and so on [Nielsen, 1993].

In the December study, we were able to improve our understanding of how to perform a field study, and obtain more direction in eliciting the P-Tab's requirements. In this February 2006 study, we intended to put into operation all of the lessons learned from the December study and consequently carry out a more effective field study that would provide us with more and better requirements for the P-Tab.

6.2 Goals

Our main goals were the following:

- To elicit more requirements for the P-Tab, and gain a better understanding of the project scope, deployment context and design constraints.
- To learn more about the relationship between group size and horizontal or vertical work surfaces.

We intended to meet our main goals in part by accomplishing these secondary goals:

- To help the designers be more productive, so that they would produce more material for us to evaluate.
- To get a better view of what the designers were doing, so that we could better evaluate the material they created.

6.3 Methods, Test Plan and Infrastructure

The February study took about 12 working days to prepare over the course of one month. The much shorter time was due to experience derived from the December study, a pre-established waiting list of participants, and the fact that no new materials needed to be purchased for the lab.

6.3.1 Test Plan

Here were our decisions on how we had to conduct the February study, based on our results from the earlier December study:

- We planned for the study to take place on 2006-Feb-04 from 13:00 to 17:30.
- We planned for the field study to take place in Concordia University's Engineering, Computer Science and Visual Arts complex (the EV building) in Concordia's Usability and Empirical Studies lab, room EV-9.210. We would use our research group's lounge and guest room (EV-9.189) to greet the participants upon their arrival.
- Camera quality had to be much higher, so that we could see what participants were sketching on the table or whiteboard. This would allow us to better understand the design process, and perhaps how relatively effective or creative different tools and group positions are.
- We needed to encourage the participants to split up into smaller groups or work individually—while everyone was still in the same room. This would allow us to better understand the effect of individual work on the group, and the effect of group work on the individual.
- We needed more participants so that we would have more specialists in each field, and more people in general. We wanted to see what group dynamics would arise from that.
- We needed to give the participants more clear goals and provide more detailed requirements, in case this was why the results from the first study were so meagre.

- We would provide more and better tools for the participants: such as a scanner (to make Adobe Photoshop more useful), Adobe Illustrator, Flash (for prototyping) and Internet access (for reference and generating ideas). This would hopefully help produce better results.
- This time we would quickly introduce the participants to all of the tools that we had made available to them.
- We would ask participants to write their names on their sketches so we could confidently identify who did what.
- We considered making a horizontal whiteboard available, to see how it would be useful.
- We would not ask for help from other Human-Centered Software Engineering (HCSE) team members this time. We would work on executing the field study with just the two of us (Jonathan Benn and Rozita Naghshin), that way there would be a minimum of interference to the participants.
- We would carry over some of the participants from the December study to the February study. This had multiple advantages: they could explain what happened in the December study (and hopefully speed up the newcomers' understanding), we could observe how they explained what happened and perhaps learn something about the transfer of design knowledge between disciplines, and it would also be an effective simulation of something that might realistically happen in a work environment.

6.3.2 Laboratory Setup

We made the following design tools available to the participants (and let them know at the beginning of the study which tools were available):

- Whiteboard with dry-erase markers and eraser.
- Pens, pencils, paper, rulers, markers, etc.
- Computers: a Mac and a PC. The Mac had the software Macromedia Flash, Adobe Photoshop and Adobe Illustrator installed. The PC had the software

Microsoft Word, Microsoft Visio, Adobe Photoshop and Adobe Illustrator installed.

- Scanner (connected to PC).
- Projector (connected to PC by default, but could be connected to Mac instead).
- Internet (on both computers). Enabling Internet forced us to keep on the noisy switch in the room. This was a trade-off that resulted in lower sound quality for the study recordings.

For this field observation we put a lot of effort into making sure that we could get a much better view of the designers and what they were doing. Instead of just having one camera, we set up three—each with its own sound input focused on a different part of the room. **Figure 21** shows the layout of the lab after we finished setting it up.

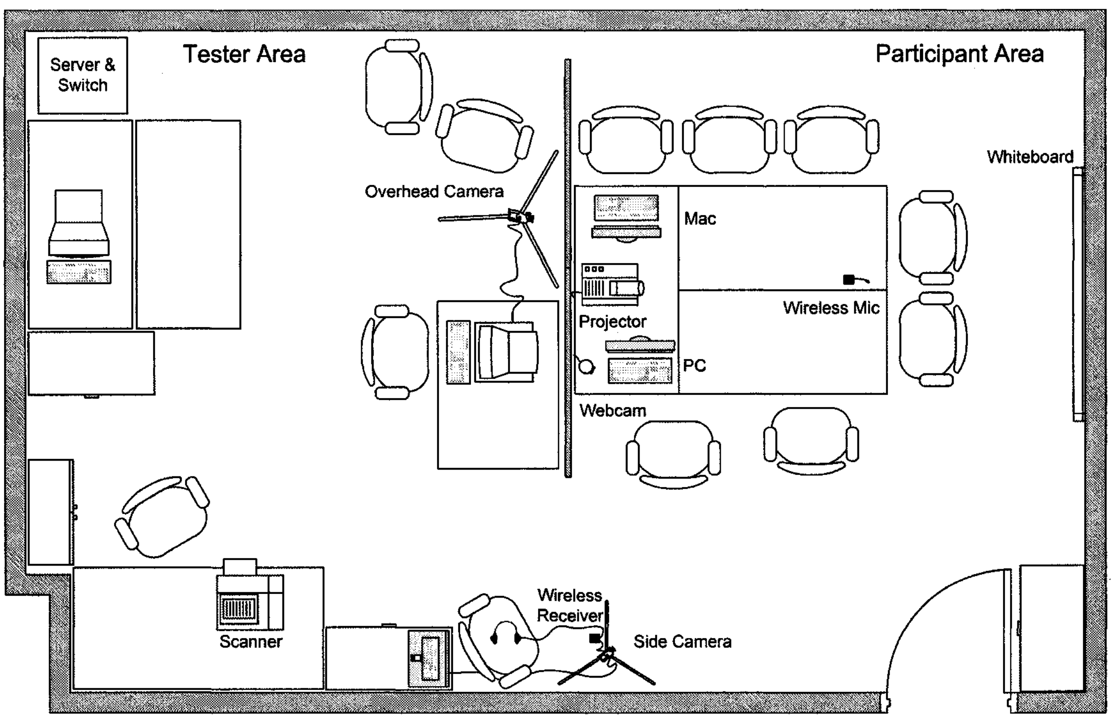


Figure 21: February Field Observation Setup

One camera (the *side camera*) recorded the participants, their conversations near the whiteboard, and the whiteboard itself. It used a wireless microphone.

One camera (the *overhead camera*) was located near the ceiling (thanks to a very large tripod) and recorded the contents of the table on which the participants were working, and also recorded any conversations taking place on the far side of the room from the side camera.

The last camera was a webcam being used in conjunction with Morae Recorder. The webcam recorded the users of the PC and their conversations, while Recorder recorded what they were doing with the PC. Unfortunately, there was no way for us to record events on the Mac since Recorder was Windows-only software.

While we intended to include a horizontal whiteboard for the field observation, it turned out to be too large to fit on the table we had in the lab. Hence, it was excluded.

6.3.3 Documentation

We prepared another set of documentation for this second study:

- A slightly modified consent form for the participants.
- An introduction document that would help remind us of everything we needed to talk to the participants about before the test began.
- We prepared another procedure document (**Section 11.1.1**, p.100), but skipped writing a new roles and responsibilities document since only two testers were involved we both knew what responsibilities were involved.
- An improved and more detailed design problem statement (**Section 11.1.2**, p.101).
- We prepared new post-test questions (**Section 11.1.3**, p.105). We planned to ask the post-test questions individually to each designer, instead of as a group. We hoped that we could potentially get more and better information from each of them, in an environment where their responses would not be being judged by their peers. If one of them had a complaint about a peer, they would be able to freely express it.

6.4 Population and Selection Method

In terms of finding new designers for the second study, we tried to keep as many of the designers from the previous study as possible. As it turned out, only Brian and George were able to take part again. The remaining participants we found through our waiting list or personal contacts. Hence, unlike with the December study, this time we didn't need to advertise.

We non-randomly selected the participants for the study. We selected based on participation in the December study, the ability to make a firm commitment to the February study date and time, and for maximum amount of design experience. We got firm commitments to participate from 3 software designers, 3 art designers and 1 game designer. All of the participants arrived for the study.

The participants are listed below, with their names changed to protect their identities:

- Three art designers: Brian, Dorothy and Upton
- Three software designers were George, Scott and Yardley
- One game designer: Tim

6.5 Data Collection and Analysis

We collected data from our study by reviewing the recorded video footage and taking notes on relevant data. We marked down the time of each observation, so that we could view the same situation from different cameras if necessary. In our notes, we focused on what tools the participants were using, how they were arranging themselves, and what sorts of cross-discipline dialogue they were saying. For the data resulting from the field observation, see **Section 11.2** on p.105.

We also summarized the participants' answers from the post-test questions. These notes can be found in **Section 11.3** on p.111.

6.5.1 Review of Field Observation Results

This time the video footage was of very high quality, and the three different video angles and sound inputs gave us the ability to listen in on different parts of the group's conversations, and to see what they were doing from different angles. This often allowed us to see what was really going on, when only one camera would have made this impossible.

On the other hand, because we kept on the room's noisy network switch (in order to enable Internet for the participant's computers), the sound quality for the recordings was lower than that of the December study.

The participants' social dynamics were much more interesting in the February study. Instead of working together in the largest possible group, they split into sub-teams all in the same room (possibly because we encouraged them to do so). The teams split by discipline, although we did not explicitly ask them to do that. This had interesting results. Sub-teams would ask each other cross-discipline technical questions. They would provide instructions or ask for clarification of instructions. Sub-teams would surreptitiously listen in on the conversations of the other sub-teams, and then spontaneously join in when the topic was relevant to their own work. When a distraction (such as a telephone call or need for coffee) caused one member of a sub-team to leave, anyone left alone temporarily joined another sub-team. This provided more cross-pollination of ideas.

Only two testers (Jonathan Benn and Rozita Naghshin) conducted the second study. We both introduced ourselves to the participants. Afterward, even when we had to enter or leave the usability lab, it didn't seem to disturb the participants at all. This was in stark contrast to the December study, where the testers' presence and movements disturbed the participants.

In the February study, the participants produced a much larger quantity of design material than in the December study. They produced 24 pages of notes, including art and design diagram sketches.

This increased efficiency could have been due to many different factors, including:

- We provided the participants with more detailed requirements.
- The fact that this was a second session (and that these may be intrinsically more efficient). A lot of the conceptual work had already been done and was summarized to the team by Brian.
- A lack of strangers entering and leaving the room, disturbing the participants.
- Higher efficiency due to better social dynamics arising from the team of participants splitting up into sub-teams.
- The larger group size. There were 7 participants in the February study, and only 5 participants in the December study.

In a study of this kind, it's impossible to say which of these factors was the most important. We can only recognize that they were there. However, it would be interesting to learn the factors' relative importance through future research.

Sketching and doodling was an essential activity in the design study. It was used by individuals to generate ideas or pass the time, and it was used by groups to communicate ideas more clearly.

When they were provided with food and drink, the participants ate and drank a lot. This would have put any large, horizontal smart board at risk. In particular, the risk of spilling liquids is significant.

There were many instances of participants using jargon from their discipline that was not understood by a participant from another discipline. When the term was explained it was usually understood in the end. Sometimes the term was not understood, especially if the explanation was brief.

We failed to make it clear to the new designers from the outset that the current study was a continuation of the December study. It's not clear how much of an impact this had on the design process.

Brian became the advocate for the design established in December. He asked us to provide him with the written design from the December study, if only to refresh his memory. Given the brevity of the December design notes, this suggests that the product's design was largely in the previous team's heads and that the written form was just a memory aid. He repeatedly defended the old design, and pushed the February participants to accept the decisions that were previously made. Finally, Tim asked Brian to relate everything he could remember about the first study. This design communication was largely verbal.

The software sub-team used the whiteboard to write down one of its design diagrams. This diagram was then used for reference when the team was sitting at the table. At time (2:12:50) Scott asked us if he needed to recopy the whiteboard or if we had recorded it.

Macromedia Flash was available to the designers in order to allow them to create rapid prototypes. The participants were informed that this tool was available, however, no designer ever used it.

6.6 Findings and Recommendations

6.6.1 Methodological Findings

Based on the results from both the December and February studies, here are some recommendations for the usability lab itself and for conducting field observations and usability tests:

- The usability lab requires a one-way mirror and a back door, so that any strangers observing a usability test can enter, observe participants, and leave the observation area without being seen by the participants.
- Limit the number of testers that deal directly with the participants, and make sure that they are officially introduced.
- Don't have any additional sources of noise or distraction that could disturb the test (*e.g.* servers, printers, telephones with a high ringer volume, etc.).

6.6.2 Conceptual Findings on P-Tab Requirements

Based on the results from both studies, here are some findings regarding the nature of designing in a multi-disciplinary team:

- Sketching and doodling was a very important activity during both design studies. Furthermore, in the post-test questions nearly every designer cited pencil and paper as being the tool that made them feel most productive or creative. The only exception was Scott, who preferred the whiteboard—which still uses a marker for writing and sketching.
- Consequently, we can propose that the P-Tab would have to have excellent support for sketching in order for it to be of any use to designers.
- A horizontal surface can be useful during the "design by committee" stage of design (when everyone is simply sitting evenly around the table and talking). But while this stage of design seems to help the design team understand what's going on and what to do, it doesn't produce any immediate results.
- A horizontal design surface is optimal at producing results for 2 or 3 people. More than that, and some people will not be able to see sketches and hear the discussion as well, making them feel left out of the discussion.
- A vertical design surface such as a whiteboard, on the other hand, allows a large number of people (4 or more) to see sketches. This allows them to at least take part passively in the discussion, even if only a handful of people (perhaps 1 to 3) can actively write on the whiteboard at once. Dorothy explicitly mentioned in the post-test questions that a presentation (presumably with a projector) or whiteboard would be most useful for communicating designs.
- There is no significant advantage to using a vertical surface for 3 or fewer people, since they can easily see each other's work on a horizontal surface anyway. This does not stop small groups from using a whiteboard. According to Scott, the size of the whiteboard allowed him to step back and see the whole picture, its size made it easy to collaborate with others simultaneously, and it was easier to erase things than on pencil and paper. It seems that the large size advantages are only effective for a vertical surface, because a horizontal surface wouldn't allow

designers to step back to see the whole picture (unless they had a ladder) and depending on the size of the table might have more inaccessible surface area.

- Consequently, we can propose that the P-Tab requires a vertical display surface for designers as well as a horizontal display surface. Either surface alone would be insufficient to meet the designers' needs, but together they allow the right mix of small team work and large team communication and collaboration.
- In a large group of designers, the problem of needing to recopy a design from one medium to another is significant. A sketch hand-drawn by a pair of designers may need to be communicated to a larger team. To do this, the sketch must be recopied onto a vertical surface, or copied to digital format (from which it can have multiple copies printed or be displayed with a projector). Modifications made to the diagram during group review will then need to be recopied back to the original version.
- Consequently, we can propose that in addition to its capacity to support sketches, the P-Tab must also have facilities to help designers clean up their sketches to make them more presentable, it must support the automatic conversion of sketches into functional diagrams, and it must allow sketches to be transferred between the horizontal and vertical display surfaces.
- Since human beings like to eat and drink, the P-Tab should be designed to accommodate these activities.

6.7 Concluding Remarks

In this experience report we described how we performed the February field study, and what we learned from it. We successfully met our secondary goals of making the designers more productive, and of getting a better view of what they were doing. Accomplishing these secondary goals fed into the success of our primary goals, which were to elicit more P-Tab requirements and further investigate the properties of horizontal vs. vertical surfaces.

Regarding horizontal vs. vertical surfaces, what we saw during the February study confirmed our observations during the December study. Even though large horizontal surfaces (such as tables) are very useful for design work, large vertical surfaces (like whiteboards) are also necessary for many different reasons, including:

- Communicating with large groups of people at the same time (more than two other people).
- Allowing a close-up and far-away look at sketches or notes on the surface.
- The size of the writing surface allows many people to collaborate simultaneously on the same task.

We were able to elicit the following requirements for the P-Tab:

- P-Tab requires excellent sketch support.
- P-Tab requires a combination of horizontal and vertical display surfaces.
- P-Tab must facilitate cleaning up sketches to make them more presentable.
- P-Tab must allow the automatic conversion of sketched diagrams into formal diagrams.
- P-Tab must allow sketches and program files to be seamlessly transferred from display surface to display surface, and back again.
- P-Tab must be constructed to allow eating and drinking around it or on it.

7 Conclusion

Through our investigations of design literature, new design environments and software designers in action, we contributed to identifying and addressing the drawbacks of modern computerized design tools. We met our research goals, which were to uncover the Participatory Tangible Board (P-Tab)'s requirements (see **Section 7.1**) and propose design guidelines that would meet those requirements (see **Section 7.2**, p.79).

Our main investigations were as follows:

1. To research the state of the art in design theory and learn more about the nature of software design.
2. To explore how designers might benefit from the new technologies that are being developed.
3. To design, conduct and analyze two empirical studies of software designers. By studying designers performing design tasks in a multi-disciplinary environment, we aimed to better understand the social and cognitive processes that underlie design activities.

In researching software design we learned that software design involves: the iterative development of a correct and fit software product, the empirical measurement of design models and prototypes, and the description of software components, interfaces and algorithms. Our investigation of software tools gave us insight into some of their weaknesses, such as their general lack of support for sketching—a vital design activity.

The continued emergence of immersive computerized environments (such as virtual reality, augmented reality and tangible interfaces) makes new design interaction metaphors possible. Our investigations of the new environments being developed today allowed us to consider how these new environments could be applied to design. For example, they could be used to facilitate visualization and rapid prototyping, and provide automated feedback.

Our investigations showed that pencil and paper is still the tool of choice for most designers (especially during the early stages of design) in spite of the many advantages of using computers. This confirms our initial observation that contemporary computerized software design tools do not adequately support design activities. The drawbacks of these computerized tools include an inability to foster creativity, support visualization and enable group work. As a result, designs are regularly produced using conventional means (such as pencil and paper or marker and whiteboard). Often, hand-drawn designs are then transferred to computer form, which is typically time-consuming, error-prone and difficult to automate.

7.1 Summary of Findings for P-Tab Requirements

Based on all of our research and studies, we have established lists of requirements and recommendations that the P-Tab should adhere to for it to be useful to designers. Meeting these requirements should be sufficient to make the P-Tab an effective design environment for any reasonable group size. A P-Tab implementing the requirements should, hypothetically, all software designers to produce better designs with less time and effort required. However, whether or not this is the case remains to be seen, and needs to be proven through further research.

The P-Tab has the following requirements:

- **P-Tab must have excellent sketch support.** Sketching must not need any additional skills to use. This confirms what we thought before we started our investigations. If anything, our studies showed that sketching is even more important to designers than we initially suspected.
- **P-Tab must allow emphasis to be placed on sketches or in diagrams,** such as by allowing some areas to be written in bold, underlined, written in thicker ink by pressing harder on the pen, or written in a different colour.
- **P-Tab must allow annotations to be inserted wherever appropriate.**
- **P-Tab must support rapid prototyping.** We did not know this before we began our studies. The importance of this came out through our literature searches, which emphasized the importance of rapid prototyping to creativity, because it

allows designers to inexpensively produce and evaluate many different design ideas.

- **P-Tab must provide design advice and feedback on design quality.** The necessity for this was suggested by the P-Tab project proposal, but we remained sceptical at first. In the end, our research showed that not only was this feasible (at least, within the next decade) but that it would help designers produce better designs by helping them evaluate different designs. Advice and feedback would also be of great help for beginning designers in training them to a higher level of skill.
- **P-Tab does not need to be easily portable.**
- **P-Tab must be immersive,** allowing natural interaction, which helps creativity. We learned this through our research of virtual reality and creativity literature.
- **P-Tab requires a combination of horizontal and vertical display surfaces.** This was our most surprising result of the studies. We underestimated how dynamic a large group of designers would be, and how necessary a vertical surface was for enabling communication between larger groups of people.
- **P-Tab must allow sketches and program files to be seamlessly transferred** from display surface to display surface, and back again. This was a surprising result that was made obvious during the February field study. We observed several instances of designers writing, and then copying and recopying diagrams. Easy sketch duplication and transfer would save them a lot of time.
- **P-Tab must allow the automatic conversion of sketched diagrams into formal diagrams.** The main use of many contemporary software design tools is for cleaning up diagrams initially created by hand. Hence, there is a need to formalize diagrams in order to make them easier to communicate. If this process were automated, it would make the designers' job much easier and allow them to concentrate on creative and collaborative work.
- **P-Tab must be constructed to allow eating and drinking around it or on it.** While we initially suspected that this might be the case, the February field study showed without a doubt that any large horizontal smart board would have a significant risk of food spills.

- **P-Tab must be able to run contemporary software,** *e.g.* Rational Rose, Adobe Photoshop, etc. This was not entirely expected, but it was shown to be necessary by both empirical studies. The designers were used to using specific software tools for certain tasks, and forcing them to abandon these tools would be counter-productive. At least for the current generation of designers (and hence, for the foreseeable future), it's necessary to support their needs for specific legacy tools.

We also recommend the following features for the P-Tab:

- Time-saving computerized features, *e.g.* cut/copy/paste, load/save, etc.
- Ability to receive multiple interactions simultaneously, by different people or by both of a single person's hands
- Support for zoom levels, so that parts of sketches or diagrams can be compressed or expanded. If possible, this could implement fisheye views.
- Support for layers in sketches and diagrams
- At least some of the P-Tab's interfaces should be widgetless, in order to improve the environment's immersiveness and designers' creativity.
- If the P-Tab uses wearable or hand-held equipment, that equipment should be lightweight and wireless.

7.2 Guidelines for the P-Tab's Design

Based on the P-Tab's requirements, as listed in **Section 7.1**, we developed guidelines for implementing the P-Tab. These guidelines suggest how the P-Tab might be physically constructed, and provide a little detail on how its software might function. A summary, visual layout and list of benefits for the P-Tab suggestion can be found in **Figure 22** and **Figure 23**. The terms used in the diagrams are explained on p.57.

The P-Tab should be composed of multiple parts: a large vertical smart board (perhaps 1.5 m x 1 m) for everyone's use and a series of smaller smart tablets (about 40 cm x 30 cm) for the use of individuals. Smart tablets can be laid flat on a table, held up at an angle or even held up vertically.

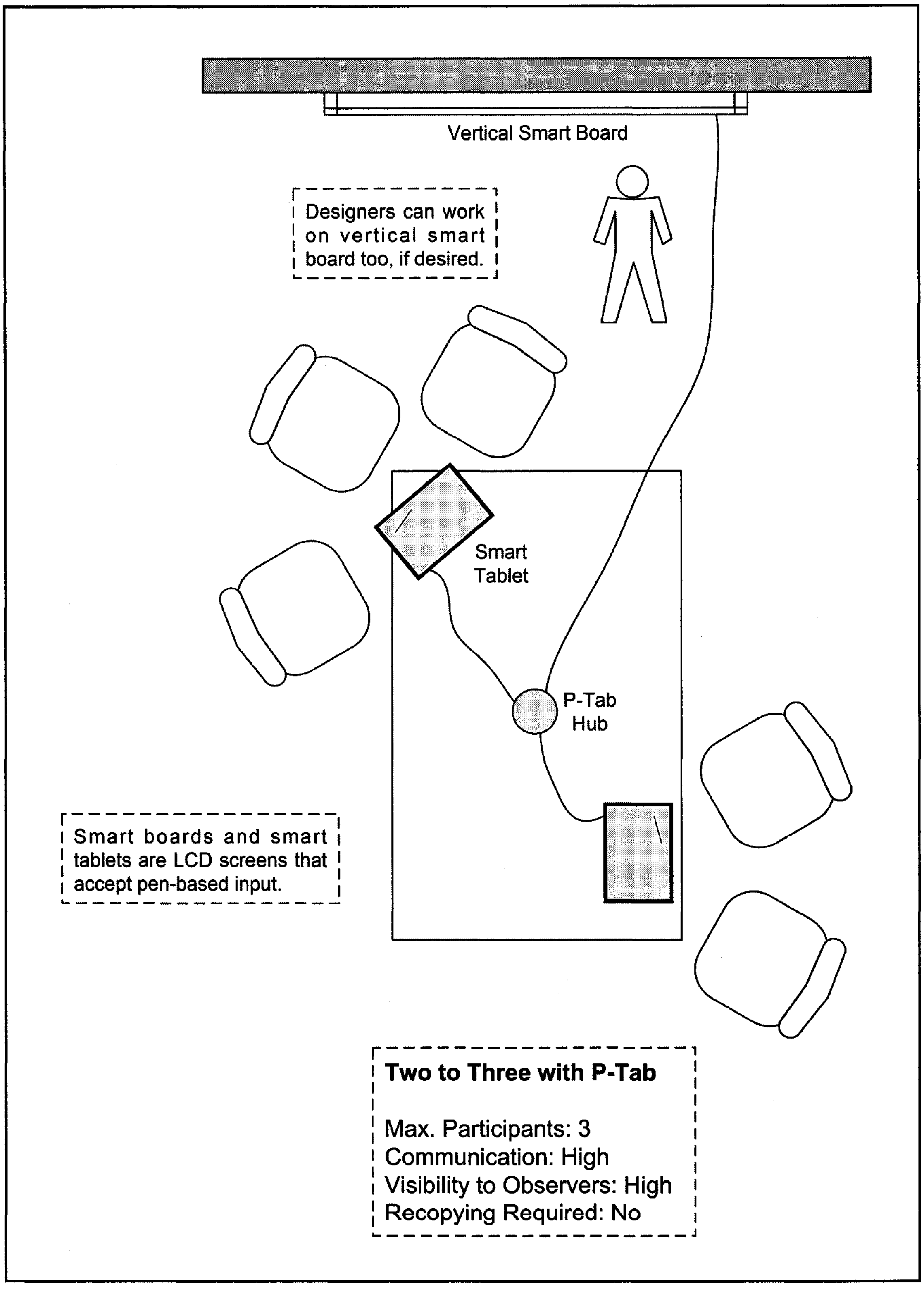


Figure 22: Two to Three with P-Tab

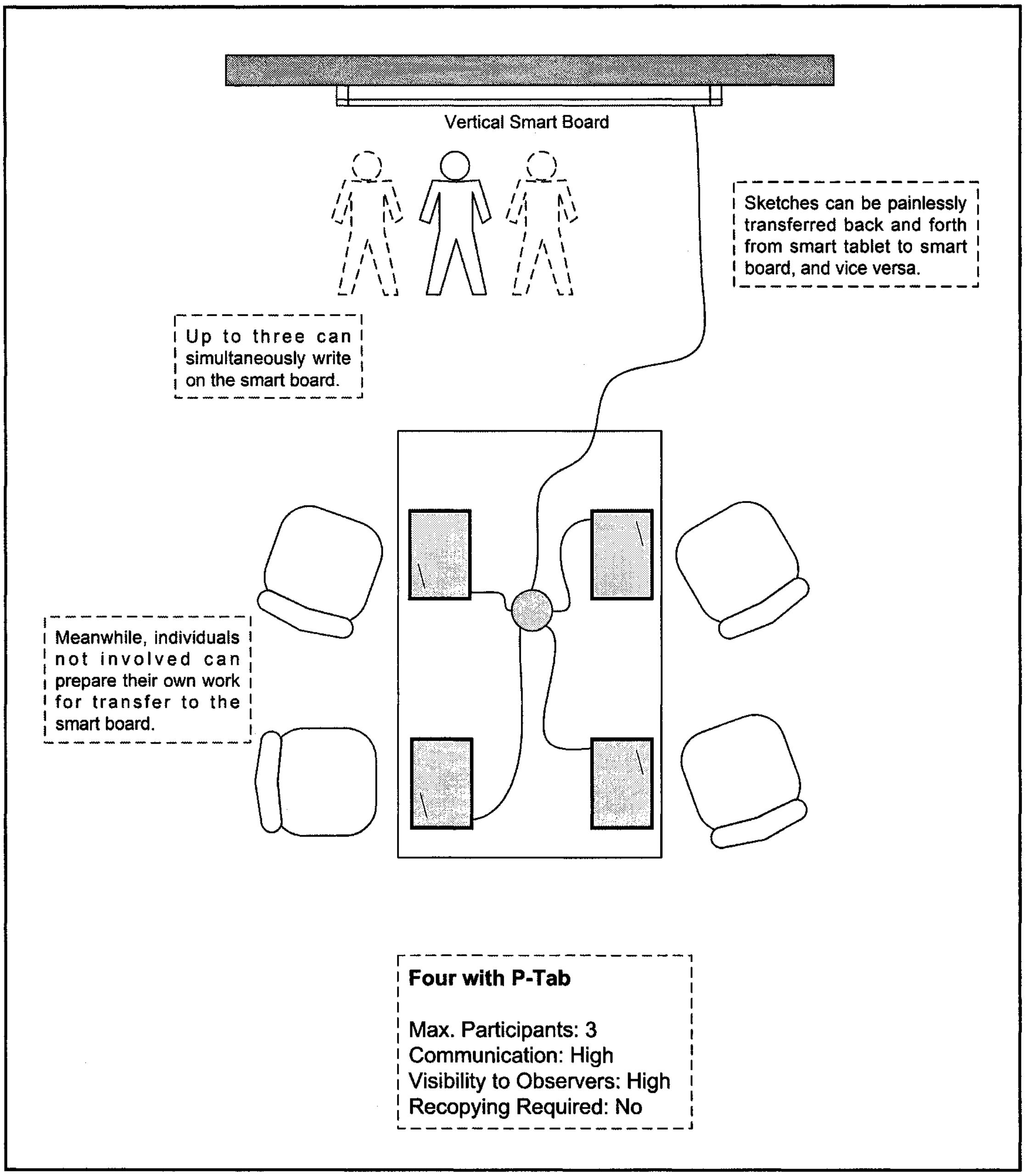


Figure 23: Four with P-Tab

A smart tablet is an LCD screen that can receive input from a special pen (instead of a mouse). It replaces the standard computer monitor/mouse/keyboard combination. A smart board is just a very large version of the smart tablet.

The smart devices should support sketching. It should be very easy to accomplish, ideally as easy as using pencil and paper. Ideally, sketches should never need to be recopied manually. They should be able to be saved, modified and transferred to another medium.

Ideally, sketches should be shown vertically or horizontally at will. To this end, the smart devices should be able to communicate with each other via tangible metaphors. In other words, a sketch on one smart device should be easily transferable to another smart device, such as by a simple pointing device. A typical scenario would be where individuals sketch something on their tablet, and then they transfer the sketch to the vertical smart board by pointing at their sketch and then pointing at the smart board. Next, team members observe the sketch, annotate it and modify it. They then transfer the modified sketch back to the original person's smart tablet.

7.3 Concluding Remarks

In this thesis, we have succeeded in uncovering many of the P-Tab's basic requirements and provided guidelines on how to implement the P-Tab in order to meet those requirements. Together, the requirements and guidelines form a hypothesis on how to address the shortcomings in existing software design tools. Further research is needed in order to confirm this hypothesis. A researcher's most likely next step will be to begin creating a physical P-Tab prototype that can be iteratively developed and empirically tested in order to answer the many remaining questions on how to best develop software design aids.

The P-Tab should be composed of a vertical wall-mounted smart board and a series of smaller hand-held smart tablets. A smart board or tablet is an LCD screen that accepts pen-based input. Each smart component should be interconnected with the others via a network. The P-Tab should support basic sketching and easy sketch transfer between components. The inclusion of effective rapid prototyping tools, sketch conversion/cleanup tools and software agents should further improve its capacity to aid software designers in performing their tasks.

8 References

[Alexander, 1977] C. Alexander, S. Ishikawa, M. Silverstein, M. Jacobson, I. Fiksdahl-King, and S. Angel, *A Pattern Language*. New York: Oxford University Press, 1977.

[Bailey, 2002] B. Bailey and J. Konstan, "Supporting Innovation: A Behavior-Sketching Tool for Early Multimedia Design," University of Minnesota, Minnesota 2002-May-05, 2002.

[Beck, 2002] K. Beck, *Test Driven Development: By Example*: Addison-Wesley Professional, 2002.

[Boehm, 1979] B. W. Boehm, "Software engineering: R & D trends and defense needs," in *Research: Directions in Software Technology*, vol. Ch. 19, P. Wegner, Ed. Cambridge, MA: MIT Press, 1979.

[Budgen, 2003] D. Budgen, Software Design, 2nd ed: Addison Wesley, 2003.

[Convard, 2005] T. Convard, "Conception assistée par ordinateur en environnement immersif," in *Département Communication Homme-Machine*. Orsay, France: Université Paris XI, 2005, pp. 151. http://www.limsi.fr/Individu/convard/theseconvard.pdf

[Coplien, 1998] J. Coplien, *Multi-Paradigm Design for C++*: Addison-Wesley, 1998.

[Csikszentmihalyi, 1997] M. Csikszentmihalyi, *Finding flow: The psychology of finding engagement with everyday life*. New York: Basic Books, 1997.

[Davies, 1996] C. Davies and J. Harrison, "Osmose: Towards Broadening the Aesthetics of Virtual Reality," *ACM Computer Graphics [special issue on Virtual Reality]*, vol. 30, pp. 25-28, 1996.

[Davies, 2004] C. Davies, "Virtual Space," in *Space in Science, Art and Society*, F. Penz, G. Radick, and R. Howell, Eds.: Cambridge University Press, 2004, pp. 69-104.

[Dumas, 1999] J. S. Dumas and J. C. Redish, *A Practical Guide to Usability Testing*: Intellect Books, 1999.

[Dumas, 2004] J. S. Dumas, R. Molich, and R. Jeffries, "Describing usability problems: are we sending the right message?," *Interactions*, vol. 11, pp. 24-29, 2004.

[Gamma, 1995] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns: Elements of Reusable Object Oriented Software*: Addison-Wesley Professional, 1995.

[Gould, 1985] J. D. Gould and C. Lewis, "Designing for Usability: Key Principles and What Designers Think," *Communications of the ACM*, vol. 28, pp. 300-311, 1985.

[Haapasalo, 2000] H. Haapasalo, "Creative Computer Aided Architectural Design: An internal approach to the design process," in *Department of Industrial Engineering*. Oulu: University of Oulu, 2000, pp. 189.

[Hammond, 2002] T. Hammond, K. Gajos, R. Davis, and H. Shrobe, "Sketch Recognition in Software Design," MIT Artificial Intelligence Laboratory, Boston 2002.

[IEEE, 1990] IEEE, "Glossary of Software Engineering Terminology," in *IEEE Std 610.12-1990 (R2002)*: IEEE, 1990.

[IEEE, 2005] IEEE, *Guide to the Software Engineering Body of Knowledge*, 2005-Feb-16 ed: IEEE, 2005. http://www.swebok.org/ironman/pdf/SWEBOK_Guide_2004.pdf

[Immersence, 2006] Immersence, "Osmose Description," 2006. http://www.immersence.com/osmose/osmose.htm

[Ishii, 1998] H. Ishii, C. Wisneski, S. Brave, A. Dahley, M. Gorbet, B. Ullmer, and P. Yarin, "ambientROOM: Integrating Ambient Media with Architectural Space," presented at CHI, 1998.

[Jones, 1970] J. C. Jones, *Design methods: seeds of human futures*. New York: Wiley-Interscience, 1970.

[Jones, 2000] S. Jones, "Towards a Philosophy of Virtual Reality: Issues Implicit in 'Consciousness Reframed'," *Leonardo*, vol. 30, pp. 125-132, 2000.

[Kurtenbach, 1996] G. Kurtenbach, G. W. Fitzmaurice, T. Baudel, and B. Buxton, "The design and evaluation of a GUI paradigm based on two-hands, tablets and transparency," presented at ACM Conference on Human Factors in Computing Systems, 1996.

[Larman, 2005] C. Larman, *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development*, 3rd ed: Prentice Hall PTR, 2005.

[Lucente, 1998] M. Lucente, G.-J. Zwart, and A. D. George, "Visualization Space: A Testbed for Deviceless Multimodal User Interface," presented at Intelligent Environments Symposium, Stanford University, 1998.

[MacLean, 1993] A. MacLean, V. Bellotti, and S. Shum, "Developing the Design Space with Design Space Analysis," in *Computers, Communication and Usability: Design issues, research and methods for integrated services, North Holland Series in Telecommunication*, P. F. Byerley, P. J. Barnard, and J. May, Eds. Amsterdam: Elsevier, 1993, pp. 197-219.

[Mayhew, 1999] D. J. Mayhew, *The Usability Engineering Lifecycle: A Practitioner's Handbook for User Interface Design*: Morgan Kaufmann Publishers, 1999.

[McRae, 2004] S. McRae, "Engineers help an artist access her virtual worlds," in *Concordia Thursday Report*, vol. 29, 2004. http://ctr.concordia.ca/2004-05/nov_18/05/

[Myers, 2004] B. A. Myers, J. F. Pane, and A. Ko, "Natural Programming Languages and Environments," *Communications of the ACM*, vol. 47, pp. 47-52, 2004.

[Newman, 2003] M. W. Newman, J. Lin, J. I. Hong, and J. A. Landay, "DENIM: An Informal Web Site Design Tool Inspired by Observations of Practice," *Human-Computer Interaction*, vol. 18, pp. 259-324, 2003.

[Nielsen, 1993] J. Nielsen, *Usability Engineering*: Academic Press, 1993.

[Owen, 2004] C. L. Owen, "What is Design?: Some Questions and Answers," Illinois Institute of Technology: Institute of Design, 2004.

[Parnas, 1972] D. L. Parnas, "On the Criteria To Be Used in Decomposing Systems into Modules," *Communications of the ACM*, vol. 15, pp. 1053-1058, 1972.

[Rekimoto, 2002] J. Rekimoto, "SmartSkin: An Infrastructure for Freehand Manipulation on Interactive Surfaces," presented at CHI 2002, 2002.

[Repenning, 2004] A. Repenning and A. Ioannidou, "Agent-Based End-User Development," *Communications of the ACM*, vol. 47, pp. 43-46, 2004.

[Seffah, 2005] A. Seffah, "Software Design Activities," J. Benn, Ed. Montreal, Canada, 2005.

[Shalloway, 2001] A. Shalloway and J. R. Trott, *Design Patterns Explained: A New Perspective on Object-Oriented Design*: Addison-Wesley, 2001.

[Shedroff, 2001] N. Shedroff, *Experience Design 1*: New Riders Publishing, 2001.

[Shen, 2004] C. Shen, F. D. Vernier, C. Forlines, and M. Ringel, "DiamondSpin: An Extensible Toolkit for Around-the-Table Interaction," presented at ACM Conference on Human Factors in Computing Systems (CHI), 2004.

[Sherman, 2003] W. R. Sherman and A. B. Craig, *Understanding Virtual Reality*: Morgan Kaufmann Publishers, 2003.

[Sommerville, 2001] I. Sommerville, *Software Engineering*, 6th ed: Addison-Wesley, 2001.

[TechSmith, 2006] TechSmith, "Black or blank screen when recording media playback," 2006. Answer ID 27, http://techsmith.custhelp.com/

[Turetken, 2004] O. Turetken, D. Schuff, R. Sharda, and T. T. Ow, "Supporting systems analysis and design through fisheye views: Demonstrating the usefulness of integrating context-based views into the systems analysis and design process.," *Communications of the ACM*, vol. 47, pp. 72-77, 2004.

[Wirfs-Brock, 2002] R. Wirfs-Brock and A. McKean, *Object Design: Roles, Responsibilities, and Collaborations*: Addison-Wesley, 2002.

9 Appendix A: Osmose Experience Report

9.1 Complete List of Materials

These are the materials we used to set up the usability test (see **Figure 6**):

Camcorder: a high-quality digital video camera with low light capability that is mounted on a tripod.

Computer ("Recording PC"): a PC equipped with dual 2 GHz Xeon processors, 1 GB RAM, Windows 2000 Professional operating system, Parhelia/P650/P750 Dual Head 128 MB graphics card, and Sound Blaster Live sound card.

Computer ("Logging PC"): a PC equipped with a 2.8 GHz Pentium 4 processor, 512 MB RAM, Windows XP Home Edition operating system, and Radeon 9200 128 MB graphics card.

Floodlight: a bright lamp designed to provide ambient lighting.

Headphones: standard cheap headphones.

Monitors: There were three different monitors used. They allowed test conductors, located in various places, to observe what the participant was seeing from inside the helmet. In **Figure 6**, Monitor (A) showed what participants could see with their right eye, while monitors (B) and (C) showed their left eye view.

Morae Recorder: software by TechSmith that records a computer's screen, mouse clicks, keystrokes, input from a webcam, and input from a microphone. Morae Recorder combines all of that information inside a single synchronized file. This software was installed on the Recording PC.

Morae Remote Viewer: software by TechSmith that connects with a computer running Morae Recorder, and allows the *data logger* (see **Section 9.2**) to see what's being recorded in real-time and to add annotations. This software was installed on the Logging PC.

Network Switch: a computer network apparatus used to connect the Recording PC with the Logging PC.

Premiere: software by Adobe that can display input from a digital video camera on a computer screen. This software was installed on the Recording PC.

Speakers: These are (D) in **Figure 6**. These allowed the testers to hear what the participant was hearing from inside the Osmose helmet.

Webcam: a standard cheap web camera mounted on a miniature tripod.

Wireless Microphone: a microphone composed of two parts: a wireless transmitter and a receiver. The receiver can simultaneously send its sound output to two different sources: directly into a computer's sound card, and to a speaker system (*e.g.* headphones).

9.2 Testing Roles and Responsibilities

During the usability tests our team and the people we tested adopted the following roles:

- **Participant:** The user of the system to be tested. In other words, the person on whom we put Osmose's vest and helmet, and that experiences the virtual environment while simultaneously being observed.
- **Technician:** The technician's original role was to create the usability testing environment for Osmose. Once testing began, his task changed to a support role. At the start of every testing day, the technician booted up Osmose and set up the testing environment. During the day's tests the technician helped calibrate the equipment to each participant, started and stopped recordings, and was generally available in case of technical issues. At the end of every testing day, the technician shutdown Osmose and securely stored the equipment.
- **Welcomer:** The welcomer assisted the participant before the usability test. She organized a meeting place and time with participants, greeted them and brought them to the testing location and then gave them the consent form and the pre-test questionnaire (which requested basic information about the participant such as age, weight, height, profession, interests and virtual reality experience). The pre-

test questionnaire was answered by the participant without any help from the welcomer. Once this was complete, the welcomer introduced the participant to the monitor and her job was done.

- **Monitor:** The monitor assisted the participant during the usability test. The monitor answered questions, and sometimes offered directions. He or she provided help for the participant in putting on and taking off the Osmose helmet, vest and wireless microphone. In conjunction with the technician, the monitor calibrated the breathing for the vest. He or she enforced the think-aloud protocol, requiring participants to speak about what they were thinking and doing. Once the test was ready to begin the monitor turned off the room's lights, turned on the floodlight, and signalled the start of the test for the purpose of timing it. During the test the monitor made sure that the participant did not wander away from the magnetic sensor controller (marked by the X in **Figure 6**). Once the test was complete, the monitor introduced the participant to the interviewer.
- **Data Logger:** There were two types of data loggers: a manual data logger and an automated data logger. The manual logger wrote notes by hand, marking down when participants needed assistance, noting problems they had had or obstacles they had run into, and making general observations. The automated data logger recorded events using Morae Remote Viewer, marking off whether or not tasks were completed successfully, and marking when tasks were started and stopped for the purpose of timing them.
- **Interviewer:** The interviewer assisted the participant after the usability test. The interviewer gave participants the post-test questionnaire, which asked for information regarding their perspective on where they went in the virtual world, what they did, whether or not they found the experience immersive, and how they would better implement the system were they the designer. The participants were not given a paper to fill out; the interviewer asked all of the questions and wrote down all of the answers. The reason for that was to leave the participant free to talk about the experience and help prompt the participant for useful information.

During a usability test each role was in theory taken by one person, with the exception of the data logger that was taken by two people simultaneously (one manual and one automated). Technically, since the visible roles (welcomer, monitor and interviewer) all had their responsibilities taking place at different times they could all be taken care of by one person. Hence, from four to six people were required to run every usability test. Typically, the monitor also took the role of the interviewer thus resulting in a crew of five. We performed dozens of tests over the course of three weeks, and the members of our research group worked in shifts in order to prevent anyone from getting exhausted.

Having the technician on hand during every test, while seemingly not strictly required, turned out to be important. We discovered a bug in Osmose that caused the screen-saver to turn on after about 10 minutes (when one session of Osmose took about 15 minutes). Hence, the technician had to be present to prevent the screen-saver from turning on. Furthermore, Osmose would occasionally crash, and it was important to have the technician on-hand to reboot the system as quickly as possible.

9.3 Testing Procedure

Our testing procedure evolved over time as we developed the testing environment, and conducted pre-tests and real tests. This is the procedure that resulted:

1. (Only once at the start of the day) Technician sets up the equipment and turns on Osmose.
2. Participant arrives.
3. Welcomer welcomes participant.
4. Welcomer asks participant to fill out the consent form.
5. Welcomer asks participant to fill out the pre-test questionnaire.
6. Welcomer introduces participant to monitor.
7. Monitor explains test process to participant.
8. Monitor brings the participant into the testing area.
9. Technician begins the recording.
10. Monitor helps participant put on vest, helmet and wireless microphone.

11. Monitor, technician and participant calibrate the vest breathing detector.
12. Technician starts an Osmose session and the usability test begins.
13. Monitor provides participant with basic navigation training, as Osmose provides an approximately 3-minute period for this before the virtual world is entered.
14. Usability Test, Part 1: *Free Exploration.* Monitor tells participant to freely explore the virtual world for 5 minutes.
15. Usability Test, Part 2: *Specific Tasks.* Monitor asks participant to perform the following series of tasks (which takes about 5 minutes):
    a. Look up
    b. Look down
    c. Look left
    d. Look right
    e. Turn around (360 degree turn)
    f. Move up
    g. Move down
    h. Move left
    i. Move right
16. Usability Test, Part 3: *Think Aloud Protocol.* Participant is told to freely explore again for another 5 minutes, but this time is asked to think aloud about everything that he or she is doing.
17. After a total of about 15 minutes, the Osmose session ends automatically, thus effectively ending the usability test.
18. Monitor helps participant remove the vest, helmet and wireless microphone.
19. Technician ends the recording.
20. Monitor introduces participant to interviewer.
21. Interviewer asks participant questions, filling out the post-test questionnaire as they go along.
22. Interviewer says goodbye to participant.
23. Participant leaves.
24. (Only once at the end of the day) Technician puts away all of the testing equipment and turns off Osmose.

10 Appendix B: P-Tab December Field Study

10.1 Documentation

Below are reproductions of documentation we used for the field study. Only relevant documentation is being included in this appendix. For example, the consent form and call for participation are not included.

10.1.1 Roles and Responsibilities

During the Participatory Tangible Board (P-Tab) Design Study we will adopt the following roles:

Participant

- A participant is one of the designers being observed during the test session.
- There should be 6 participants in all, two graphic designers, two software designers and two game designers. In practice, the quantities might differ.

Technician

- The technician is responsible for the technical aspects of a test session.
- Sets up the laboratory where the study will take place, so that participants can be recorded and the data loggers can do their job.
- Starts, stops and stores recordings.
- Monitors the status of an in-progress recording
- Sets the lighting to the appropriate level.
- Keeps track of time, and informs the other testers when it's time for a change.
- Is available in case of a technical emergency.
- Takes down and stores the equipment at the end of a session.
- Only one technician is required

Monitor

- Monitors are responsible for dealing directly with participants.
- Greets participants and brings them to the lounge and test location.
- Gives participants the consent form and pre-test questionnaire

- Provides the information participants need to perform their task.
- Answers participant questions and offers help if necessary.
- Interviews the participants after the test, and fills in the post-test questionnaire.
- There should be 2 monitors, since the participants end up getting split into two groups during the middle of the session.

Data Logger

- Data Loggers are responsible for observing. They provide additional information and insight regarding the test session in progress.
- Uses Morae Remote Viewer to add supplemental information to the session recording.
- Records when the different phases of the session start.
- Takes note when participants run into obstacles, ask for assistance, show displays of emotion, etc.
- Makes general observations relevant to the study. For example, noting what participants are doing, or when they accomplish tasks.
- Ideally there should be 2 data loggers: one observes the participants, and the other observes how the participants are using the P-Tab prototype.

Ideally, the participants should never really be aware of any of the testers except for the Monitors. In practice this will be difficult.

Each test session requires a crew of 4 or 5 in order to handle the 6 participants.

Participants will be split into two groups with 2-4 people in each: Group 1 and Group 2. Group 1 will contain the graphic designers and one game designer. Group 2 will contain the software designers and one game designer. One monitor takes care of each group (hence we can label them Monitor 1 and Monitor 2).

10.1.2 Procedure

12:00 PM

1. Technician sets up the equipment.

1:00 PM

2. Participants arrive.
3. Monitors welcome participants and lead them to the lounge.
4. Monitors ask participants to fill out the consent form.
5. Monitors ask participants to fill out the pre-test questionnaire.
6. Monitors bring the participants into the testing area.
7. Monitors explain test process to participants.

1:30 PM

8. Phase 1 begins: both groups design together.
9. Technician begins Phase 1 recording.

2:30 PM

10. Phase 1 ends.
11. Technician ends Phase 1 recording.
12. Group 2 leaves the testing area and goes to the lounge.
13. Phase 2 begins: Group 1 designs alone.
14. Technician begins Phase 2 recording.

3:15 PM

15. Phase 2 ends.
16. Technician ends Phase 2 recording.
17. Group 1 leaves the testing area and goes to the lounge.
18. Group 2 returns to the testing area.
19. Phase 3 begins: Group 2 designs alone.
20. Technician begins Phase 3 recording.

4:00 PM

21. Phase 3 ends.
22. Technician ends Phase 3 recording.
23. Group 1 returns to the testing area.
24. Phase 4 begins: both groups design together again
25. Technician begins Phase 4 recording.

5:00 PM

26. Phase 4 ends.
27. Technician ends Phase 4 recording.
28. Participants return to the lounge.
29. Technician puts away all of the testing equipment.
30. Monitors ask participants questions, filling out the post-test questionnaire.

5:30 PM

31. Monitors say goodbye to the participants
32. Participants leave.

10.1.3 Design Problem

You are being asked to design the following game for the mobile phone:

The player takes the part of a wealthy Montrealer who has just been dared by her (or his) friends to spend all of her money—if she can. This is harder to accomplish than it would seem at first, as this character has a trust fund and money is constantly pouring into her bank account. Plus, to make matters worse, she's a picky shopper and won't buy just any product at just any price. She does have standards after all, and doesn't like getting ripped off!

Your **Goal** is to produce a design that will be detailed and coherent enough to make the implementation of the game possible. In terms of **Game Design** this means writing down all of the major game rules and game play mechanics. For **Graphic Design** this means representing all of the major settings and characters of the game. On the **Software Design** side, this means laying out the major modules and classes of the software. Needless to say, since each of these areas affects the others you will all need to work together in order to create an effective design. You do only have about 3 hours, and we aren't expecting any miracles—do what you reasonably can within the allotted time.

Choose a good **Title** for the game.

Characters to choose from: this is your design decision. Players may only have one fixed character role, or multiple character roles to choose from. Create a profile and personality for each player character role the game will support.

The **Target Audience** is teenagers and young adults of *both* genders, ages 13 to 25.

Mobile Phone Design Constraints include:

- Typical play time on a mobile phone is 5-10 *minutes*.
- You can only assume the following limited assortment of buttons: 0 through 9, up, down, left, right, ok, cancel, and side.
- The screen is colour, with a resolution of only 128 pixels wide by 160 pixels high.
- Processing power and storage space are extremely limited compared with a desktop computer. For example, 3D graphics are out of the question.

Advertising and **Distribution** are out of your hands, however, feel free to provide suggestions on how the finished product should be advertised and/or distributed.

If you need **More Detailed Requirements**, or some **Ideas** to jump-start your creative process, just ask.

10.1.4 Post-Test Questions

1. Did you have enough time to meet the design challenge? Did you have too much time?
2. With regard to the people you worked with, would you have liked a different mix of disciplines, or a different number of people?
3. Was there any tool missing from the suite that we provided you?
4. Did you have any communication problems with your fellow designers?
5. Did you enjoy yourself? Would you participate in another paid study?

10.2 Field Observation Results

After the field study was over, we went over the video footage from the field observation. Unfortunately, our results were not very good. An error in the camera settings led to the contents of the table and the whiteboard being completely illegible, which made it difficult to know what the participants were doing. Below is a transcript of the information we were able to assemble from the video:

Phase 1—all participants: Adrian, Brian, Ed, George and Jerry.

- The participants read the design problem and discussed for about 13 minutes.
- The participants started using the pen & paper type materials. They sketched and wrote while discussing as a group for another 14 minutes.
- Jerry started writing on the whiteboard, with the other participants watching and discussing. This lasted 9 minutes.
- Brian started writing on the whiteboard for another 21 minutes.
- All of the designers sat down and watched the whiteboard, and Adrian got up and started working on it. This lasted 3 minutes.
- Phase 1 ended. During the entire phase, participants were either spread around the table, or standing up in front of the whiteboard.

Phase 2—art designers: Adrian and Brian.

- They used the whiteboard together for 7 minutes.
- They sat down and sketched for 3 minutes while they waited for Photoshop to load on the computer.
- They *sketched* with Photoshop for 10 minutes (there were no legible results).
- They sketched with pen and paper for 19 minutes.
- Phase 2 ended a little early at the request of the participants. During the entire phase, Adrian and Brian worked as a pair.

Phase 3—software designers and game designer: Ed, George and Jerry.

- Adrian and Brian spent the first 4 minutes of this phase explaining their previous work to the software/game team.

- When Adrian and Brian had left, then Ed, George and Jerry arranged themselves in a triangle at the corner of the table and began sketching and discussing together. They worked like this for 27 minutes.
- The three participants began using the computer and projector. They spread out evenly around the table, and Jerry sat at the computer. He turned on Microsoft Word and began working on it with feedback from Ed and George. This lasted 11 minutes.
- Phase 3 ended.

Phase 4—all participants

- Ed, George and Jerry stayed where they were and were joined by Adrian and Brian, who sat in such a way as to keep the distribution of people around the table even.
- The five participants worked together in this way for 56 minutes.
- Phase 4 ended.
- By this point the design team had managed to produce a four-page point-form Word document, as well as some sketches.

Throughout all phases of testing, the testers frequently entered and left the usability lab. There were 5 testers in all, and they sometimes needed to go to the washroom, speak on the phone, get something to drink, etc. It was often obvious that this disrupted the design session.

10.3 Post-Test Answers

Below is a summary of the participants' answers to the post-test questions:

- Adrian said that he had enough time to address the requirements and preliminary design, and that more time would have been a waste. By the end he was losing his focus, and with more time he wouldn't have gotten any more done.
- George believed that there was enough time to address the requirements, but not enough for a detailed design or a prototype.
- Ed would have liked more time to go over software structure.

- Brian said that at the point they were at now, it was necessary for the art designers to work individually for a period of time, and then come back together to discuss their results.
- Jerry said that at his workplace people would typically do design work individually, and then come together occasionally to share their ideas and work on a whiteboard, a process that typically took from 2 to 10 days.
- Adrian thought that the design session had a good mix of specialists and generalists.
- Brian thought that the mix was good for brainstorming but that more people, and more specialists in particular, would be required for the next session to get a more detailed design.
- Jerry said that he typically designed in pairs and that it was faster that way. However, he thought that working in a multi-disciplinary team was a good idea for encouraging "buy-in"—that is having all the stakeholders agree with the design.
- The participants claimed that the following tools were missing: scanner, Macromedia Flash, Internet access, Adobe Photoshop, Adobe Illustrator, Rational Rose.
- Jerry's favourite tool was a whiteboard. He would typically use a digital camera to take a picture of it if he needed a digital copy.
- Jerry said that there were no communication problems between the participants that they weren't able to work through—it was just a question of perspective.
- Brian said that there were no communication problems this time, but that in the next sessions the problems would come out as discussions became more specific to a domain.
- Adrian mentioned that they never discussed code (*i.e.* software) the whole session.

11 Appendix C: P-Tab February Field Study

11.1 Documentation

Below are reproductions of documentation we used for the field study. Only relevant documentation is being included in this appendix. For example, the consent form and participant introduction document are not included.

11.1.1 Procedure

12:00 PM

1. Technician sets up the equipment in the lab.

1:00 PM

2. Participants arrive.
3. Monitors welcome participants and lead them to the lounge.
4. Monitors ask participants to fill out the consent form.
5. Monitors ask participants to fill out the pre-test questionnaire.
6. Monitors bring the participants into the testing area.
7. Monitors explain test process to participants

1:30 PM

8. Design session begins
9. Technician begins recording.

4:30 PM

10. Design session ends.
11. Technician ends two of three recordings.
12. Participants return to the lounge and wait their turn.
13. Each participant, one at a time, enters the lab again
14. Monitors ask participant questions, filling out the post-test questionnaire.
15. The monitor gives the gift
16. Monitor says goodbye to participant, and s/he leaves

5:15 PM

17. Technician ends the last recording.

18. Technician puts away all of the testing equipment.

11.1.2 Design Problem

You are being asked to design the game ***Shop 'til You Drop*** for the mobile phone:

The player takes the part of a wealthy Montrealer who has just been dared by her (or his) friends to spend all of her money—if she can. This is harder to accomplish than it would seem at first, as this character has a trust fund and money is constantly pouring into her bank account. Plus, to make matters worse, she's a picky shopper and won't buy just any product at just any price. She does have standards after all, and doesn't like getting ripped off!

Your **Goal** is to produce a design that will be detailed and coherent enough to make the implementation of the game possible. In terms of **Game Design** this means writing down all of the major game rules and game play mechanics. For **Graphic Design** this means representing all of the major settings and characters of the game and creating a prototype. On the **Software Design** side, this means laying out the major modules and classes of the software.

Needless to say, since each of these areas affects the others you will all need to work together in order to create an effective design. You do only have about 3½ hours, and we aren't expecting any miracles—do what you reasonably can within the allotted time.

The **Target Audience** is teenagers and young adults of both genders, ages 13 to 25.

Shop 'til You Drop should be designed for mobile phones. Mobile phone **design constraints** include:

- Typical play time on a mobile phone is 5-10 minutes.
- You can only assume the following limited assortment of buttons: 0 through 9, up, down, left, right, ok, cancel, and side.
- The screen is color, with a resolution of only 128 pixels wide by 160 pixels high.

- Processing power and storage space are extremely limited compared with a desktop computer. For example, 3D graphics are out of the question.

Game Mechanics

Character Selection

- Players have three character roles to choose from, each affects the game by having different product preferences (which might imply a difficulty level and/or style of game).
- Provide short description of character and details on product likes/dislikes.
- In addition, starting wealth and rate of increase can be determined at this stage to set a difficulty for the game (*e.g.* an easy-going character might be more appropriate for an easy-going player).

Character's Apartment

- The apartment shows all the products that have been purchased to date. The character can't purchase the same product twice, but can perhaps purchase a better/newer product to replace it with.

Malls

- At the beginning of the game only one mall is unlocked and can be entered. The others are locked until the player completes the preceding malls.

Entering Mall

- Throughout the game the amount of cash remaining and failure condition (*e.g.* time limit, limit on number of stores visited) are displayed.
- Navigate the mall and enter stores to buy goods.
- Obstacles get in the way of speedy navigation.
- Fast navigation leads to better results—less money accumulating (hence, there's room for player strategy and optimization)

Goods/Services Types in Stores

- Technology: Phones, Music, TVs, Stereos, Computers, Games
- Clothing: Hats, Shirts, Pants, Suits, Dresses, Skirts (Kilts?)

- Services: Travel, Pets, Beauty, Food

Browsing

- Player browses through the store's list of goods and selects them one by one for purchase. Their icons, price, description, etc. is available.
- Every time the player attempts to add a product to the shopping cart, the character must be convinced that it's a good purchase.
- The character's facial expression represents how happy she is with the purchase: 0-dislike, 1-disinterest, 2-impartial, 3-like, 4-love
- Initial facial expression depends on the player's making a good choice for the character. After that, convincing improves the reaction.
- The character can only accept the purchase if she is at least at level 3. At level 4 she spends extra money (*e.g.* giving a tip or buying gloves to go with that).
- The character can be convinced in different ways: perhaps with button mashing, perhaps by conversing and/or haggling with the store representative (*e.g.* "Could I get 10% off?", "Do these pants make my butt look big?"), perhaps by talking to herself (*e.g.* "Sure this ice cream will go straight to my hips, but it'll taste *so* good!"), or perhaps by calling her friends/family and asking for advice.
- If the player fails to convince the character and feels that enough time has been wasted, then s/he can give up and try a different product or purchase the goods already in the shopping cart and go to a new store.

Purchase (Checkout)

- This is the satisfying part for the player, where s/he is rewarded with the cash level going down, a cool sound, animation, picture of the apartment, etc.

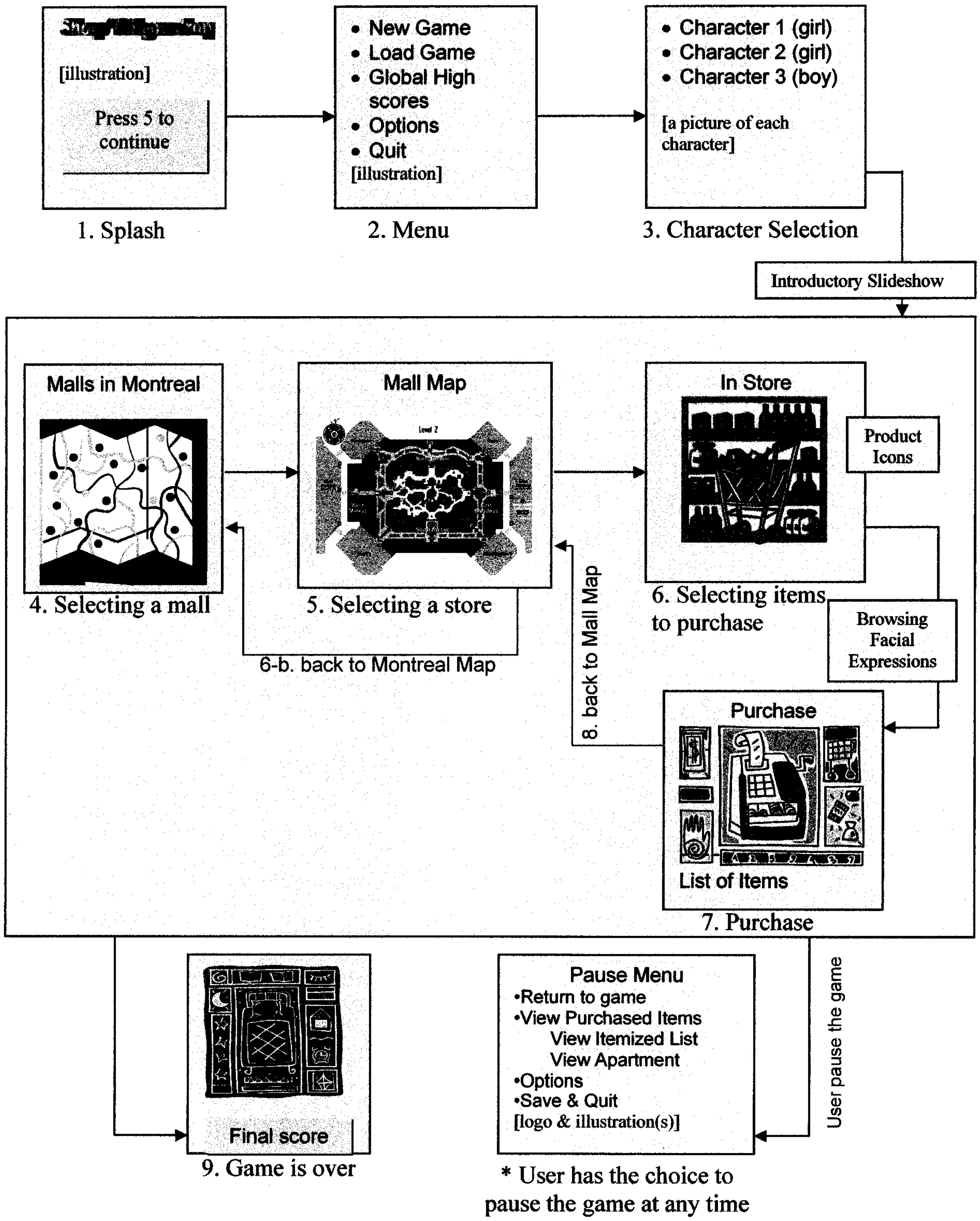
Pause Menu

- Recalling that this is a cell-phone game, the game can be paused at any time with one button, and it should be very easy (two button presses) to save and quit.

Winning and Losing

- The player wins if the character spends all of her money

- The player loses if she runs out of time, or some other factor (*e.g.* if each store can only be visited once and they've all been visited).



11.1.3 Post-Test Questions

1. With regard to the people you worked with, would you have liked a different mix of disciplines, or a different number of people?
2. Did you have any communication problems with your fellow designers?
3. Was there any tool missing from the suite that we provided you?
4. Which tool made you feel most productive?
5. Which tool made you feel most creative?
6. In a real project, how would you record your design decisions and communicate them with others?

11.2 Field Observation Results

After the field study was over, we went over the video footage from the field observation. Below are notes on the events that took place, with a focus on how the participants arranged themselves and on cross-discipline questions the participants asked each other. Each note is prefixed with a timestamp, in the format (hour : minute : second), which is relative to the start of the side camera's recording.

At first all of the participants seated themselves around the table. Upton sat near the PC while Yardley sat near the Mac.

(0:01:20) We begin telling the participants what we expect of them and what tools we have made available for them.

(0:04:00) The design session begins. Tim starts off the discussion by asking the team basic game design questions, like what their audience is and how the game will be fun.

(0:04:45) Scott asks the team for clarity on how the game actually works.

(0:06:53) With regard to the game requirements, Scott says, "This doesn't strike me as fun."

(0:07:56) George asks Tim how the general process of game design works.

(0:11:10) Tim recommends to the group that they sell the game through its characters.

(0:11:20) Scott says that from a software point of view, there's no reason why the game should be limited to only three characters.

(0:11:45) Scott asks the team, "How does the game work?"

(0:13:45) Scott asks the team, "What will the characters do in the game?" This is a cross-discipline question.

(0:15:15) Brian asks for the sketches and design documentation from the December study. He says that even if they're brief that they will help him refresh his memory of the events.

(0:16:25) Scott begins writing software design brainstorming notes on the whiteboard.

(0:16:45) Tim uses the term "RPG" (an abbreviation for Role-Playing Game). Scott doesn't know the term and asks for clarification. This is an issue of cross-discipline jargon.

(0:18:05) Brian begins writing on the whiteboard. He is writing about the game's user interface. Scott sits down.

(0:19:10) Scott asks for software-related information from the other designers, and uses the term "GUI" (an abbreviation for Graphical User Interface) to disparage the information they've provided him with so far. Brian doesn't understand the term. Even after an explanation, Brian continues to focus on artistic and visual issues.

(0:21:50) Scott says he has no feel for what the game does.

(0:25:50) There is a conflict between the old vision of the game (from the December study) and the new vision from this study. Brian objects when the vision shifts, and defends the old design.

(0:29:53) Scott contributes a game design idea to the group.

(0:31:00) Brian begins sketching on the whiteboard, while talking to the team. At the same time, Scott gets up to jot another annotation on another part of the whiteboard and then sits down again.

(0:36:21) Scott realizes that this is the second design session, and that there was already one two months ago (perhaps we should have made this more clear?).

(0:46:45) The direction of the discussion brings Brian to mention "levels", a game play element mentioned in the previous study, but not mentioned up to this point in the current study.

(0:47:40) Tim suggests that the design team assign roles to its members and split up.

(0:52:50) Tim asks the participants, "Why am I here?" because they seem to imply that he is not needed. To which Scott replies, "Why are any of us here, except for you?"

(0:55:50) Brian does not understand the difference between the terms "real-time" and "turn-based." Together, Scott and Tim explain the terms to him.

During the last hour, the conversation has been almost completely between Brian, Scott and Tim. In the meantime, Upton has been sketching on his own or occasionally using the PC. Similarly, Yardley has been occasionally using the Mac on his own. George and Dorothy have been listening, only speaking very rarely.

(1:03:05) Brian finishes recounting his experiences from the last study, and Tim provides a summary of his own understanding.

(1:10:30) Brian continues to advocate for the old design.

(1:17:00) Scott requests that the teams split up (about 30 minutes after the original suggestion by Tim).

(1:23:00) During the discussion, Scott recognizes an element that is important for his software design and says, "That's important for us."

(1:23:45) The participants split up into sub-teams based on design specialty. Tim and Brian work together on game design. Scott, George and Yardley work together on software design. Dorothy and Upton work together on art design.

(1:28:30) Tim asks the software design group (Scott, George and Yardley) the question, "How much memory does that take up?" with regard to a game feature. Scott responds, "We can do that."

(1:31:30) Brian asks the art design team (Upton and Dorothy) if they'll work on facial expressions, something that he and Tim have been discussing.

(1:33:30) Scott asks the game design team (Tim and Brian) a question regarding how the game will work.

(1:36:25) Tim overhears a question on the other side of the table, being posed by the software design team amongst themselves. He stops his own conversation with Brian to answer their question, and they discuss the game's requirements and what game elements are static or dynamic.

(1:43:55) Upton attempts to use the scanner, but the scanner fails to function.

(1:45:00) Scott gets up and starts using the whiteboard, erasing what was previously on it.

(1:47:30) Brian asks Tim a technical, software-related question. Tim answers this question directly and does not pass on the question to the software design team.

(1:48:00) George and Yardley get up and join Scott at the whiteboard. George asks us for access to Microsoft Visio, but the PC is currently being used by Upton.

(1:52:20) Tim's cellphone rings and he leaves his conversation with Brian to answer it.

(1:52:32) Brian is by himself. He approaches the art design team and asks them how they are doing.

(1:54:20) Tim finishes his telephone conversation and then starts discussing with the software design team at the whiteboard. Scott asks Tim about the game's time component, and they have an exchange of ideas. Tim and Brian then resume their conversation.

(2:01:05) Tim mentions one of the game's requirements to Brian. Scott, who is currently designing something related to that requirement, turns to them and talks to them about it.

(2:02:40) Scott asks the game design team about one of the game's features.

(2:08:05) The software design team leaves the whiteboard and sits down at the table.

(2:09:20) Tim accidentally cuts his lip and leaves the room. Brian joins Dorothy and Upton and looks at their drawings. Scott leaves to get himself a coffee.

(2:10:45) Tim and Scott return to the usability lab, and all of the participants return to their original positions around the table.

(2:12:50) Scott asks if the design information on the whiteboard is being recorded, or if he needs to copy it down.

(2:14:30) Scott begins copying some information from the whiteboard, creating new diagrams based on that information.

(2:25:35) Scott gets up and goes to the whiteboard. He edits the diagram on the whiteboard while George and Yardley watch.

(2:27:10) Scott gets up again to make more adjustments to the diagram on the whiteboard.

(2:29:00) George asks Scott a question related to the diagram on the whiteboard.

(2:32:07) Tim asks Upton and Dorothy if they can meet his game design requirements. George listens to their conversation.

(2:33:00) Brian, who clearly disagrees with Tim's view of the technical feasibility of something they are discussing, asks Scott a question about meeting the game design requirements.

(2:33:55) Upton, Dorothy and Tim all listen to Brian and Scott's discussion.

(2:35:20) Tim asks the art design team how many icons they think they can fit on the screen at once.

(2:36:00) Brian, who has been listening, gives his opinion.

(2:38:00) Dorothy joins Brian and Tim in discussing the game design, leaving Upton out of the discussion.

(2:38:45) Upton asks Tim a software-related technical question.

(2:39:30) Scott rises from his seat to look for some new pen & paper type tools, and asks for one that's out of reach.

(2:42:45) Scott asks Tim a question.

(2:45:20) Tim gives instructions and reminders to the programming team.

(2:48:35) Tim makes a passing comment in the direction of the art design team, with regard to some new design requirements they might address, but when he sees that they're already concentrating on something else he lets the matter drop.

(2:50:15) Brian, who wants a certain software feature as part of his game design, confirms with the software design team that this feature is feasible.

(2:55:05) George confers with Scott and Yardley about the whiteboard diagram.

(2:56:30) We end the design session.

Even though the scanner had worked during the setup process, during the study the scanner failed to function due to some sort of network problem. This was unfortunate, because the art designers requested a scanner several times. It seems that they might have used the computers more if the scanner had functioned correctly.

We also had a request for scissors, but those had gone missing since the December study (they had accidentally been misplaced and we never found them again until after the study was over).

The projector was never used.

Another thing we noticed was that at time (1:48:00) in the study George wanted to use Visio, but this software was only available on the PC being used by Upton. The software Upton was using, however, *was* available on the Macintosh. Thus, George and Upton could have switched places so that they could both continue what they were doing, but they didn't.

The overhead camera, whose video footage was being recorded by Adobe Premiere, ended up taking up way too much hard drive space (the recording speed was on the order of 1 MB per second). We started running out of room on the hard drive about two-thirds of the way through the test session and had to delete some recordings and ration out the remaining space by only recording when participants were working at the table.

11.3 Post-Test Answers

Below is a summary of the participants' answers to the post-test questions:

Tim—game designer

- The mix of people was good, but for an early design session 3 people is ideal.
- I didn't have any communication problems with the others, we found a middle ground. However, we didn't go into as much technical detail as I would have with other game designers.
- The tools I was provided with were sufficient. For heavy art research more would have been required, like a drawing table and blue pencils.
- Pencil and paper makes me feel most productive and creative.
- I use my art sketches directly to convince others or communicate my vision with them, but after that they're never used again except by me individually.
- To communicate a game design with others, I write it in a Microsoft Word document and then place the document under revision control. This way, we always have a good copy and we still have access to previous revisions.

Scott—software designer

- This design task was hard at first because two members of the team thought that we were continuing from a previous session, and the rest of us didn't know that.
- After Tim and I pushed Brian to hammer out the details of how the game worked the session went much better.
- In terms of the mix of people, Tim helped define the user's requirements. George and Yardley were not skilled enough as software designers to help me list the project's requirements. The artists didn't have much to do at first until the requirements were elicited. However, their presence did allow them to know what was going on.
- Having only 2 or 3 people would have been better.
- It seemed to me as though George and Yardley had just finished their first software design class. They wanted to design according to the Unified Process and didn't understand which steps of the Process could be skipped and which steps had to be done. They tried too hard to force the more practical, agile process required in this situation into the process they learned in school.
- For this task, I never needed more than pencil, paper and the whiteboard. At this point, the only tool I'd want would be a drawing tool like Microsoft Visio or Rational Rose. I only use these tools when I have a firm design and I want to make it look nice or derive code from it. For a tool like Visio, the ability to move things around gives me flexibility and helps me to think about the problem.
- The whiteboard is the tool that makes me feel most productive and creative, because it's big and lets me step back and see the whole design at once. It also makes collaboration easy and it's easier to erase things than on paper.
- To communicate my designs with others, I'd use the Unified Modeling Language (UML) and annotations, *e.g.* writing down assumptions to be checked later.

Brian—art designer

- It's important to have many people and many disciplines at a design session. In this case there was a good mix of people. However, you need all of the designers to be at a similar skill level. I had this problem, and I saw that Scott had this problem too. It takes a lot of time to make up for the fact that some team members are less skilled than you are, and in a short time frame less skilled designers might not be able to make themselves very useful.

- No tools were missing.
- Pencil and paper make me feel most productive and creative. Talking and communicating are also useful creative tools.
- I'd communicate a design with a computer. Which software I'd use depends on the situation, for example Adobe Photoshop or Adobe Illustrator. I typically do 70% to 80% of my work on computer. For brainstorming, pen and pencil is still best.

Yardley—software designer

- I would have liked a coordinator or director in the project. At the beginning we were a bit confused about how to proceed, and a director would have also helped to solve disagreements.
- I would have liked it if more computers were available.
- The tools that make me feel most productive are a computer and pencil and paper.
- To communicate a design I'd use Microsoft Word and Rational Rose for UML.

George—software designer

- The number of people was good. The graphic designers were not helpful to me in figuring out the requirements and the design. I think a better mix of disciplines would have been 4 to 5 software designers and 1 graphic designer.
- I did have a communication problem with Scott. I wanted to use the Unified Process and he didn't.
- Missing tools included Microsoft Visio and Rational Rose. I know that Visio was available, but because Upton was using the computer I didn't have access to it.
- The tool that made me feel most productive was the whiteboard, but pencil and paper made me feel most creative.
- I'd use diagrams to communicate my designs to others.

Dorothy—art designer

- The number and mix of people was fine. More people might have led to more discussion, but it would be a good idea to split people up into smaller groups.

- I had no communication problems. I mostly worked with Upton and he was also an art designer.
- The only missing tool was the scanner, which wasn't working.
- Pen and paper makes me feel most productive.
- To communicate a design I'd make a presentation or show my work on a whiteboard.

Upton—art designer

- The number and mix of people was good for obtaining different ideas, but some people took things too seriously. Working as a team was not as good as working by myself. It was a good experience, but I couldn't design everything like that.
- I had no communication problems, my team members told me what to do and I did it.
- There were no missing tools except perhaps a "graphic pen", which would allow me to undo and restart.
- A pen makes me feel most productive, but it's ink and stains that make me feel most creative.
- To communicate my designs I'd show them on paper, or use a computer because it makes it easier to clean up the designs and make them more presentable.