WeSketch: A 3D Real Time Collaborative Virtual Environment that Improves the GUI Sketching Task

by

Hamilton Andrés Hernández Alvarado

A thesis submitted to the

School of Engineering

in conformity with the requirements for the degree of

Master of Engineering with focus on Informatics

EAFIT University

Medellín, Antioquia, Colombia

November 2010

Abstract

Many tools help Graphical User Interface (GUI) designers to produce Prototypes of their future applications, but when this job requires the collaboration with geographically dispersed partners, problems related to lack Awareness of the others' work, concurrent manipulation of virtual objects and non-fluent communication of actions and intentions of participants arise.

These research started with the identification of the real functional and awareness information requirements of a group of GUI designers while they carry out the GUI sketching task in a collaborative way. The findings, mostly related to awareness and real time collaboration, were used to specify WeSketch, a 3D collaborative virtual environment in which designers can perform the task in a very natural, friendly and efficient way.

We proposed the hypothesis: "WeSketch allows a small group of designers who are geographically dispersed to be more efficient and feel more motivated during the GUI Sketching task" and in order to validate it, a set of experiments were conducted. Different measures of time and subtasks were given comparing the performance of the designers while they used WeSketch against two other tools. WeSketch showed that Task Completion Time was dramatically reduced and users expressed a better comfort with this tool since they had a higher participation and better awareness of what their partners were doing and what was happening in the environment.

These results indicate that real time collaboration and awareness information are quite important features that should be included in groupware in order to provide a more efficient and pleasant support to collaboration of dispersed designers.

Acknowledgements

The following people have played a very important role in this work. Their support have made my studies and research possible and meaningful.

I want to thank to my tutor Helmuth Trefftz. He has been an unfailing source of guidance, support and encouragement. Additionally he has always given me a vivid example of discipline and overcoming without losing humility. I would like to thank him for making me feel like a friend. It has been an honour to count with his life experiences.

Thanks also go to my friends in the SINFOCI research group, William, Maria, Jorge, Alexandra, Faber and Jaime who have been always cheering, ready and willing to contribute to this work, giving me valuable part of their time to help me with any issue.

I would also like to thank Marylin Tremaine for helping me find the best motivation to work hard on my Master, and a very special thanks goes to Camila for living my dream with me.

And thanks to my parents, Luis Enrique and Elvia María for their comprehensive and continued encouragement and support of my studies.

Contents

Chapter 1

Introduction

One of the most important tasks in User Centered Design (UCD) is Prototyping [15]. For complex software this task is usually performed by groups of designers who get together to design and choose the best Graphical User Interfaces (GUIs) for a set of requirements.

Traditionally, this task has been carried out using paper, markers and some other tools, but since this way doesn't allow designers to perform certain tasks such as: reusing parts of their designs (like with the copy-paste in computers), trying different appearances for the preliminary designs, tracing the produced artifacts along the other computerized stages of the UCD process nor working remotely; many applications have emerged offering tools to designers so that they can better manage the sketches related to specific requirements, increase their productivity and improve consistency using predefined and configurable widgets, working remotely and easily storing and retrieving produced artifacts.

Nevertheless, an intense study of the most popular of these tools has shown us that none of them offer an interaction as natural, transparent and participative as the traditional manner, keeping the participants unaware of what others are doing and forcing them to coordinate their contributions, which causes long moments of inactivity by some designers which, in turns, makes them get bored.

The Awareness factor has been recognized to affect directly the communication, fluency, productivity and, in general terms, the experience of a group of users while they perform a collaborative task [12]. The awareness of the workspace, the artifacts, the group and the tasks are some of the requirements that applications for groups should consider.

Some of the existing tools for groups allow teams to get together in 3D virtual environments where they can work collaboratively while being aware of what is going on. In these applications, the information about the group members (identity, location, state), the artifacts and the workspace is rich, but the problem of a very restrictive coordination of contributions still remains and none of these tools offer objects like the widgets that 2D applications offer. Additionally, a new problem arises; accuracy and quick response at designing gets compromised since controlling small objects like 2D widgets in a 3D environment is more difficult, as we found during our studies.

As Cook and Churcher [7] announce for the collaborative software engineering case, delivery of successful collaborative design applications is challenging, and the most critical success factors are those associated with providing effective support for aspects such as awareness and concurrency control.

Based on the functionalities, constraints and lacks we found in the 2D and 3D applications, the perceived judgement of their users, the real and specific needs of GUI designers that our studies revealed, and the existing theories about awareness and concurrency, we built WeSketch, a highly competitive application for prototyping that allows physically distributed groups to perform the task in a very friendly, participative, vivid and informative way, similar, in this sense, to the traditional manual one, and as organized and efficient as the computer assisted version.

1.1 Contributions

This work offers the following contributions to the Computer Supported Collaborative Work, Human Computer Interaction and User Centered Design community:

- Shows that designers who work collaboratively with the current prototyping tools are not happy, comfortable nor satisfied with the features offered by the most popular and accesible tools.

- Shows advantages of 2D applications for prototyping against 3D collaborative virtual environments.

- Identifies real needs of GUI designers to improve their collaboration experience in terms of communication and participation.
- Describes a practical applied case of UCD.
- Details the design and implementation of an application for GUIs prototyping.

1.2 Outline

This document is organized as follows. Chapter 2 provides a review of the state of art related to practical works on collaborative virtual environments, prototyping tools and awareness support. An extract of the theoretical works related to these topics is also provided.

Chapter 3 breaks down the user centered design process followed to conceive the tool. First a Requirements Analysis is described considering a Competitors Analysis and a set of User Studies conducted as inquiry techniques. Then, the architechture of the tool and design considerations are explained.

Chapter 4 describes a set of experiments carried out to validate the proposed hypothesis. The results of these experiments are detailed and the findings are interpreted.

Chapter 5 concludes the document by pointing out the findings of this thesis, how this work is being continued and questions that could lead to future work.

Chapter 2

Background

2.1 Prototyping

User Centered Design (UCD), as other software process models, aims at boosting the quality of the built software by improving the user experience. This is gained involving users into the product design process, seeking that the results meet the needs of them in a understandable, attractive, and easy to learn and use manner.

One of the most important activities of UCD performed to satisfy these specific needs is the Graphical User Interfaces (GUIs) prototyping [15]. The product of this iterative and incremental task is usually a group of prototypes or sketches that grow in fidelity as the process moves forward. The low fidelity sketches, also called wireframes or mockups, represent early ideas of design, usually answers to clients requirements, and are very convenient at the beginning of every software project. It's well known that inclusion of prototyping into the software process saves time and costs in the subsequent implementation and software testing phases [16].

Some prototypes support interaction with the users in order to test their usability. Paper prototypes, for example, are tested by usability experts by selecting a set of paper sketches simulating screens, reading the users a task to accomplish and allowing them to navigate through the sketches to carry out the task without any guidance. Experts expose an initial sketch to the users and change the sketch as the users simulate interaction with the controls included in the screen.

Many computer applications have emerged offering tools to designers so that they can better manage the sketches related to specific requirements, increase their productivity and improve consistency of the sketches using predefined and configurable widgets, working remotely and easily storing and retrieving produced artifacts. Navigation and usability testing of digital sketches is also supported by some of these applications.

In order to accomplish this task during the construction of complex applications, the number of profesionals involved into the development group grows and their participation is no longer temporal nor spatially synchronized necessarily, this complicates communication of the team and the simultaneous work over the artifacts to produce.

At the beginning of this project, 9 applications for prototyping were chosen and studied to evaluate their ease of use, diversity of tools for the prototyping task and their support for collaboration. Some considered criteria for this initial selection were popularity, offered features and comments on blogs. The selected tools were: Microsoft Visio [17], SmartDraw [18], Axure RP Pro 5.6 [19], IPlotz [20], Irise Studio 8.0 [21], JustInMind Prototyper 3.2.1 [22], MockFlow [23], Pencil [24] y Balsamiq Mockups [25]. A comparisson of the tools is detailed in figure 2.1.

	Application								
Features	Visio	Smart Draw	Axure	Iplotz	Irise	JustInMind	MockFlow	Pencil	Balsamiq
Pre-drawn shapes/stencils	A	A	A	A	A	A	A	A	A
Suited for GUIs	A*	A*	A	A	A	A	A	A	A
Interactive sketches	-	-	A	A	A	A	A	-	A
Navigable sketches	-	-	A	A	A	A	A	-	A
Collaborative	-	-	A	-	A	-	A	-	A
Real Time Collaboration	-	-	A	-	A	-	A	-	-
Awareness of the Group	-	-	-	-	-	-	-	-	-
Awareness of the Artifacts	-	-	-	-	-	-	-	-	-
Unsynchronized Collaboration	-	-	A	-	A	-	A	-	A
Communication with the team	-	-	-	-	-	-	Ch	-	Wk
Mode of sharing work	-	-	Chk	-	Chk	-	Chk	-	Wk
Real time collaborative discussion	-	-	-	-	A	-	-	-	-

A: Available - Ch: Chat - Wk: Wiki - Chk: Check-in/Check-out
* not exclusively

Figure 2.1: Aplications for prototyping and their features

Some of these tools are well accepted among GUI designers, since they allow designers to build sketches of their applications in a faster way but without collaboration of other members of the team, in other words, they are single user applications [25]. Others offer Check-in/Check-out systems allowing asynchronous collaboration with team members. A fluid real time collaboration is not possible since there is no up to the minute information about the group and coordination activities take designers out of their main purpose.

2.2 CSCW

People involved in the Computer Supported Collaborative Work (CSCW) research line have been producing tools that support work of groups as the described ones. Some of these applications, known as Groupware, include shared Boards or Text editors that allow people located in different internet nodes make changes and be aware of the actions of others [7]. Different authors have formulated core issues for CSCW that overlap and complement each other. Tom Gross [10] has described the fundamental requirements for cooperative work as articulation of work, situated action, double-level language, mutual influencing, information sharing and sociotechnical dimension of design, which represent the key features that should be included in every application for groups. Special attention has been paid to some issues that have been identified as challenges in these kind of applications, namely Articulation of Collaborative Work , Adaptability and Awareness [26].

2.3 Awareness

Awareness is nowadays a frequently used term in the Human Computer Interaction area for being a critical feature required to give to the user the sensation of being in total control of a tool and when a fluent communication and collaboration between users of a collaborative application is intented. [27].

Awareness can be defined in very simple words like “knowing what is going on”" [28] or “the information used for people in order to accomplish their activity in the right way” [29]. In the context of a groupal collaboration, awareness has been defined as “the knowledge of the activities of the others that gives a context for the individual ones” [26]. Specifically, formal definitions of awareness related to different issues of an interaction can be found, some of these are:

- Situation Awareness: Perception of the elements of the environment inside a volume of time and space, the understanding of their meaning and the projection of their state in a near future. [30].

- Awareness of the Group: Information and Knowledge of multiple aspects of the group and its members, to wit:

  - Informal Awareness: Pervasive experience of who is around, what these persons are doing and what they are going to do. This information is necessary to start an spontaneous interaction. [10].

  - Artifacts Awareness: Up to the minute knowledge of the artifacts and tools that others are using while they do their job [12].

  - Social Awareness: knowledge of information such as interest and attention, or emotional state of a conversation partner. [10].

  - Group-Structural Awareness: information about the group itself and its single members like their roles and responsibilities within the group, their status or their positions on certain issues. [10].

  - Workspace Awareness: information about other participants' interactions with the shared space and the artifacts it contains. The other's identity, location, level of activiy, intentions, changes to the workspace and to the object inside it [10]. This information is useful for many of the collaboration activities like coordinating an action, managing coupling, discussing about the task, anticipating actions of the others and identifying the opportunity to assist to each other. [12].

- Awareness of the Task: refers to the information required to transparently understand the tasks and their state of completeness in a group. [31].

- Objective Self-Awareness: refers to the process of taking oneself as the focus of one's own attention, or becoming aware of oneself. This information is used to identify discrepancies with an expected behavior, such as performance or similar [31].

One of the factors that directly affects communication of a geographically dispersed team is Awareness of the group and some authors have studied and proposed mechanisms that a software application should provide in order to improve the awareness [32, 27, 13, 10].

2.4 Real-Time Collaboration

Another aspect, that is highly related to core issues in CSCW and studied by this project is Real Time Collaboration. The Real-Time term is used to designate computer systems that update information at the same rate as they receive data [33]. So real-time applications are those applications in which users have immediate feedback of their actions. Considering the CSCW focus, a real-time collaboration is then a collaboration in which every participant has immediate feedback of the actions executed over a shared environment and artifacts. This feature is highly related to awareness of the situation and the workspace. Many applications have been wrongly considered to provide a real time interaction for groups. Some of the applications for prototyping described above announce real-time collaboration as their star features, but when designers try to collaborate in real-time they usually have to face frustration when finding remote collaborations mediated by check-in/check-out systems that cause long moments of inactivity and misinformation by parts of the collaboration, which in turns, make them get bored. Good real-time collaboration applications offer tools to their users trying to improve feedback and awareness not only from the system but the other participants. Some of these tools are What You See Is What I See, telepointers, multi-user scrollbars, radar views and distortion-oriented views [34].

2.5 Related Work

The following practical and theoretical works are extracts related to Awareness, Groupware, Collaboration and Interaction. Some of them could be considered close approximations to the work described in this document and some others served as the basis for the definition of some interaction techniques by pointing out theories and knowledge about humans' practices and collaboration.

2.5.1 Collaborative Design in 3D Space [1]

This work describes a collaborative design environment where Augmented Reality is applied during the design tasks of a multidisciplinary team. Participants in the collaboration walk around a table using head mounted displays to see the augmented information generated over the real scenario. Information is shared with geographically dispersed participants. Each participant has its own view of the object so rotating, moving and zooming virtual objects is allowed and the common model is not affected. Every change generates a request for acceptance that others must confirm to update the common model. To facilitate interaction with points and objects the grid and snap mode was implemented and a virtual stlyus was provided for interaction with the virtual objects. Something to exalt about this idea is that the achieved realism with augmented reality can be higher than the obtained with a completely virtual environment. Some of the limitations of this work are that audio based communication is not provided and although changes of the designs are seen in real time it is not possible to edit the model by several users at the same time.

2.5.2 The Virtual Round Table - a Collaborative Augmented Multi-User Environment [2]

In this work, authors developed a tool that allows interaction with 3D objects by multiple users. Users are present in the same place at the same time and manipulate real objects, virtual objects are superimposed. One of the difficulties faced in this project was to track the user's point of view without using pattern recognition as in [1], because they wanted free movement of objects. Objects are recognized by video tracking using two cameras to provide better information about depth. Most of the technological development of this project is based on image processing, both for tracking objects and for tracking the user's eye. This project extends the physical work environment into a virtual one while preserving the verbal and non-verbal communication and cooperation mechanisms. Restrictions announced by authors are that in this version of the project only translations of the objects are allowed. To allow six degrees of freedom tracking ultrasonic-based emitters and additional wires are required so the portability of the system is highly limited. Furthermore, accuracy at manipulating objects is not as good as it should and superimposed virtual objects occlude the real ones reducing their affordance level.

2.5.3 Supporting Telepresence by visual and physical cues in distributed 3D Collaborative design Environments [3]

The authors present a new interaction technique to support telepresence in augmented 3D collaborative design environments. They propose the improvement of the awareness of participants of a collaboration with synchronized rotary tables and the inclusion of virtual shadows for visualizing the motion of the hands of remote participants. With this, focus and gestures performed on virtual objects can be communicated. Since provided shadows are representations of real hand movements, awareness of user intentions is very similar that one obtained in a conventional same place interaction.

2.5.4 Territoriality in Collaborative Tabletop Workspaces [4]

This article documents the results of two observational studies, in which authors analyzed the way people collaborate on shared tables, both formal and casual situations. As a result they have identified that people employs the territory of a surface fragmenting it in three types of territories, personal territory, storage territory and group territory, indicating significant spatial and functional characteristics of each one. Finally, they provide some relevant design implications to be considered for building applications that support this collaboration. These implications may also be applicable to a collaboration on a virtual table.

A comprehensive comparison of different projects related to collaboration on a shared table is provided in this work. Additionally they give important details required to allow the transition between individual and group work and the publication of private content. As important details, authors identify the need for a shared space in which shared objects are stored and that there is also a tendency to form groups around an area where several partners are working together.

2.5.5 Territorial Coordination and workspace awareness in remote tabletop collaboration [5]

This study compares the behavior of interaction partners on a table when they are distant and when they are in the same place. The idea of this study was to explore these interactions with territorial coordination and transition between individual work and group work. A task performed by the participants was to deliver furniture in a designated area and then coloring the elements of the stage. Again, shadow projections are used to inform actions of remote participants. This study identified that most participants of a collaboration prefer to use a mouse instead of a stylus as the mouse avoids obstructing the view of others on the shared area. Verbal communication to solve problems of coordination or awareness is also exalted.

One observation made by authors is that the assistance offered by partners was more coordinated when they saw their partner working aside. In this project, when participants performed operations on virtual objects, pop-ups appeared obstructing the vision of shared space sections of the work. This detail should be taken into account to prevent occlusion of important information and consider presentation of menus and similar pop-up information only for the user who performs an action related.

2.5.6 AuraOrb: Using Social Awareness Cues in the Design of Progressive Notification Appliances [6]

In this project, researchers built and evaluated a light signal device that informs users of a computer when there are pending notifications, like the arrival of a new e-mail. AuraOrb is located in an area where users can see it, when there is something new for the user AuraOrb changes its color, but only when the user focuses in AuraOrb it displays a short summary of the email heading or beginning of the incoming notification. Once the user looks away AuraOrb returns to the idle state. Additionally it offers a touch sensor that allows the user to open the complete notification in the computer, which has connection with AuraOrb. The study shows positive results on the phased deployment of notifications, so that the user's attention is not distracted and undertaken tasks are not interrupted, unless the user pays attention to them. This feature can provide a better experience for the designers when they want to focus on their work.

AuraOrb provides awareness of the task and the environment, informing users that there is something going on in a gradually and not disruptive manner. This is nowadays considered for the design of notification systems, such as instant messaging systems, and similar applications should consider it too.

2.5.7 Modelling and Measuring Collaborative Software Engineering [7]

Although this work speaks specifically of applications to support collaborative software engineering, some raised points have direct application in projects of the same type as the described in this document. This paper indicates that the main problems encountered in the development of distributed collaboration applications are providing a good awareness and concurrent control. Authors say that many developers with a desire to coordinate their efforts are bound to accept extreme models of concurrent activity, and to use locks to prevent shared access to artifacts is a pessimistic solution that minimizes the parallel development. Similarly, they state that the optimistic models that allow users to update a single artifact separately to maximize the parallel development, add the difficulty of having to resolve conflicts in the changes when they are mixed. Finally the authors mention that an acceptable solution to these is to provide to participants of the collaboration the option to make the changes trusting that the group of participants respect collaborative policies by informing to the others before making crucial changes to shared work. This project describes a tool called CAIS, an architecture and some aspects that can be considered for building an application for the same purpose, among them a server, tools for client applications and propagation of events.

2.5.8 The GAZE groupware system: mediating joint attention in multiparty communication and collaboration [8]

The Gaze Groupware System is an application which implements the detection and tracking of the view to inform the participants of a distributed collaboration where the attention of users is fixed. Author explains why efforts should focus on providing visual cues instead of hearing or speech in multi-person systems. With Gaze Groupware System, it is intended that each user knows where the other users participating in a conversation are looking, by using colored light points representing each user. As a conclusion they mention that providing these visual cues improves awareness of who is speaking to whom and on what is being discussed.

2.5.9 What Groupware Functionality Do Users Really Use? Analysis of the Usage of the BSCW System [9]

For this work the author conducted a study of the use that around 10,000 users gave to the Basic Support for Cooperative Work (BSCW) system over a period of 10 months. BSCW is a web collaboration system with features like forums, file sharing, access privileges, authentication, search builder, document format conversion, versions management, event services, computer management, manage notes, appointments, contacts and tasks, and create and publish blogs. The functionalities and some quality characteristics supported by the system have been grouped into 12 categories. This study identified that more than 5 categories of actions performed in the system were: 1. access or reading documents, notes, objects and/or navigation of the workspace 2. creation or publication of new information 3. **awareness-related features** 4. modification of information or state of objects in the workspace 5. query meta-information objects.

Although this work focuses on a general purpose collaborative environment, it identifies and justifies the importance of inclusion and review of specific features in the current project. Additionally, the author exposes awareness related functionalities as the 3rd most important features in a collaborative system and it is mentioned that the awareness-related features require some time to become used to, which adds importance to the design of mechanisms of awareness much more usable and supports the interest in improving interaction and awareness for the current thesis.

2.5.10 Towards Flexible Support for Cooperation: Group Awareness in Shared Workspaces [10]

This work explores the foundations of CSCW (Computer Supported Collaborative Work) exalting them as requirements to be supplied by groupware applications, and lends a special interest in the requirements of Awareness and related. Requirements indicated are Situated Action, Double-Level language, Mutual influencing (which requires very good awareness mechanisms) and Information sharing. After describing in detail the main requirements of CSCW, the author notes that few developments or investigations describe the relationship between these requirements and that some of them are not considered.

In his work, the author focuses on the characterization of Group Awareness and subclases of awareness like Informal Awareness, Social Awareness, Group-Structural Awareness and Workspace Awareness, described in section 2.4.

2.5.11 Artifact awareness through screen sharing for distributed groups [11]

This article describes an investigation carried out to build a system that provides awareness of the artifacts that are part of an electronic work by a group of related persons.

The built system is a tool which uses screen-sharing, providing the different participants of the collaboration miniature portions of the screens of other users. These thumbnails can be enlarged for more detail and the viewer can participate by remotely pointing at parts of these screens. Additionally, they have considered levels of privacy that can be accessed by users through various protection strategies built for the system. The proposed tool doesn't offer information about location of participants nor artifacts in use.

2.5.12 A Descriptive Framework of Workspace Awareness for Real-Time Groupware [12]

In this paper the authors describe a theory that supports awareness for groupware designers. The work focuses on the Awareness of the workspace that includes the information that people have about people working with them. It also describes a framework that can serve as a basis for groupware designers, identifying some of the problems that designers usually have. Authors point out that paying attention to awareness of others has a direct impact on the flow and nature of collaboration.

2.5.13 Design for individuals, design for groups: tradeoffs between power and workspace awareness [13]

In this work, researches point out that when designing collaborative applications that allow users to work individually and as part of a group, designers find that there are conflicts in both types of interaction that need to be given priority and decide whether features should benefit the individual or they should benefit the group. This necessary balance is treated from the perspective of 3 common features of groupware: Navigating the workspace, handling artifacts and representing the view. As a proposal to the ideal solution which is to satisfy all the needs, interfaces techniques that reduce tension between individuals and groups are presented. One of these techniques is the well known Over the Shoulder View, which shares a screen to participants so that they can know what the others are doing. The shared content consist on information of a 2D Environment.

2.5.14 MOVE: Component Groupware Foundations for Collaborative Virtual Environments [14]

MOVE is a 3D collaborative virtual environment using the ANTS framework. MOVE is presented as a framework and has been developed ensuring that the extensibility of the system covers all aspects of the system giving way to the linking of components developed by third parties. It has provided a mechanism for Awareness in the server side to be so that group interactions can be recorded and informed. The awareness of the client side is basically the user's position and navigation through the 3D environment. MOVE supports up to 200 users sharing information. Artifact awareness is not included.

2.5.15 Other works about Collaboration

Saul Greenberg in his "Observing Collaboration: Group-Centered Design" [32] work, raises Workspace Awareness as a human factor. Additionally exalts the importance of a robust process for groupware development. Suggesting that Group-Centered Design should begin by observing the group and their actual working practices.

Chapter 3

Conceiving WeSketch: Construction of the Tool

In order to provide a high quality and satisfying application, we followed the main phases of a User Centered Design executing activities that involve users to the whole process. First a Requirements Analysis phase was conducted; identified the needs, a Design phase was executed including periodic tests applied to identify users' opinion and possible problems. Finally, we named the tool WeSketch and went through the final studies.

3.1 Requirements Analysis

3.1.1 Competitor Analysis

There is a wide range of applications for graphical user interface sketching that offer the same advantages and share the sames disadvantages. At the beginning of this proyect, 9 of the most known and mentioned tools were chosen and studied to evaluate their ease of use, diversity of tools for the task, and support of collaborative work.

The next tools resulted from our search of the most famous applications for the sketching task:

- Microsoft Visio [17]

- SmartDraw [18]
- Axure RP Pro 5.6 [19]
- IPlotz [20]
- Irise Studio 8.0 [21]
- JustInMind Prototyper 3.2.1 [22]
- MockFlow [23]
- Pencil [24]
- Balsamiq Mockups [25]

As shown in 2.1 most of these tools had in common features like:

- Pre-drawn stencils or widgets
- Suited for GUI's
- Interactive and Navigable sketches
- 2D environment

Very few, supported collaboration, only one of them supported Real Time discussion of the product for usability testing, and none of them supported real time Collaborative Design. Based on these features and restrictions (licensing included), we chose Mockflow [23] to conduct later studies.

In order to contrast the best 2D prototyping tools with immersive 3D environments for group interaction, we looked for the most popular 3D Collaborative Environments and analysed them to identifiy their features.

This search threw the next applications:

- SecondLife [35]
- OpenSim [18]
- Teleplace [36]

These environments had in common features like:

- Good Group Awareness
- Engaging experience
- Good communicationg systems

Only one of these tools allowed sharing of a whiteboard and collaborative drawing over it, additionally, interaction was very easy and it's license allowed us to exploit all the features we wanted to test. Teleplace [36] was selected as the best 3D collaborative environment for the next user studies.

3.1.2 User Studies

In order to identify the real needs of designers while they conduct a collaborative prototyping session, we conducted a group of user studies. [32] suggests that this is the best way to start a high quality Group-Centered Design process.

Three user studies with different settings were conducted, all of them were video taped. Our purpose was to observe a small group of GUI designers while they tried to sketch a group of 2D GUI interfaces to satisfy an specific and detailed group of requirements. We paid attention to the interaction and communication among designers, use and sharing of tools and artifacts, restrictions of the tools or methodology followed to accomplish the task and quality of the final product (the resulting sketch).

The first study consisted on the traditional manual prototyping task. Before the second and third studies, nine 2D prototyping tools and three general purpose 3D collaboration environments were compared to choose the most complete 2D and 3D applications for the studies, see section Competitors Analysis. Teleplace [36] and Mockflow [23] were the best 3D and 2D tools selected, respectively. Once every study was finished, the designers were surveyed using a standard Likert scale Survey with 3 open questions: What I liked the most was..., What I didn't like was... and What do you think that could improve the experience?, see Appendix 3.

For evey study, we considered Systems and Computers Engineering undergraduate students in their last career year. All of them had already taken at least one course related to HCI or Graphical

Interfaces Design. Additionally, we counted with collaboration of 3 professionals with a clear background on usability, who participated in the studies as users and later gave their opinion about the methodology and the tool.

3.1.3 1st study: Manual Sketching Session

The first study consisted on the observation of seven groups of designers (21 designers) while they produced a group of GUI prototypes manually, using paper, markers, pens, pencils and erasers.

Given this configuration, designer could have a fluent interaction and communication, rich information about the group and the artifacts, and they could concurrently work on a shared piece of paper (sketch).

Figure 3.1 shows the designers of one of the sketching sessions and the resulting sketch.

During the manual studies we observed users spending a lot of time drawing over and over controls on the page and then spending more time giving them a better appearance. Some of the users constantly looked at the others' drawings to ensure consistency. There were some moments in which designers wanted to include a drawing in a shared sheet but their arms collided with their partners' disrupting the development of the activity. A high percent of the groups of designers left their product unnamed so later it was difficult to identify ownership of the sketches.

These observations revealed that designers really needed a tool with attributes such as:

- several consistent configurable GUI widgets
- the possibility to add or modify a widget to the sketch in any moment without obstructing other participants
- a mechanism to store the sketches produced during a design session and assign them to specific functional requirements

3.1.4 2nd study: Sketching Session with 3D Application

The second study consisted on the observation of 7 groups of designers while they tried to carry out the prototyping task using the selected 3D tool which allowed designers to share a drawing

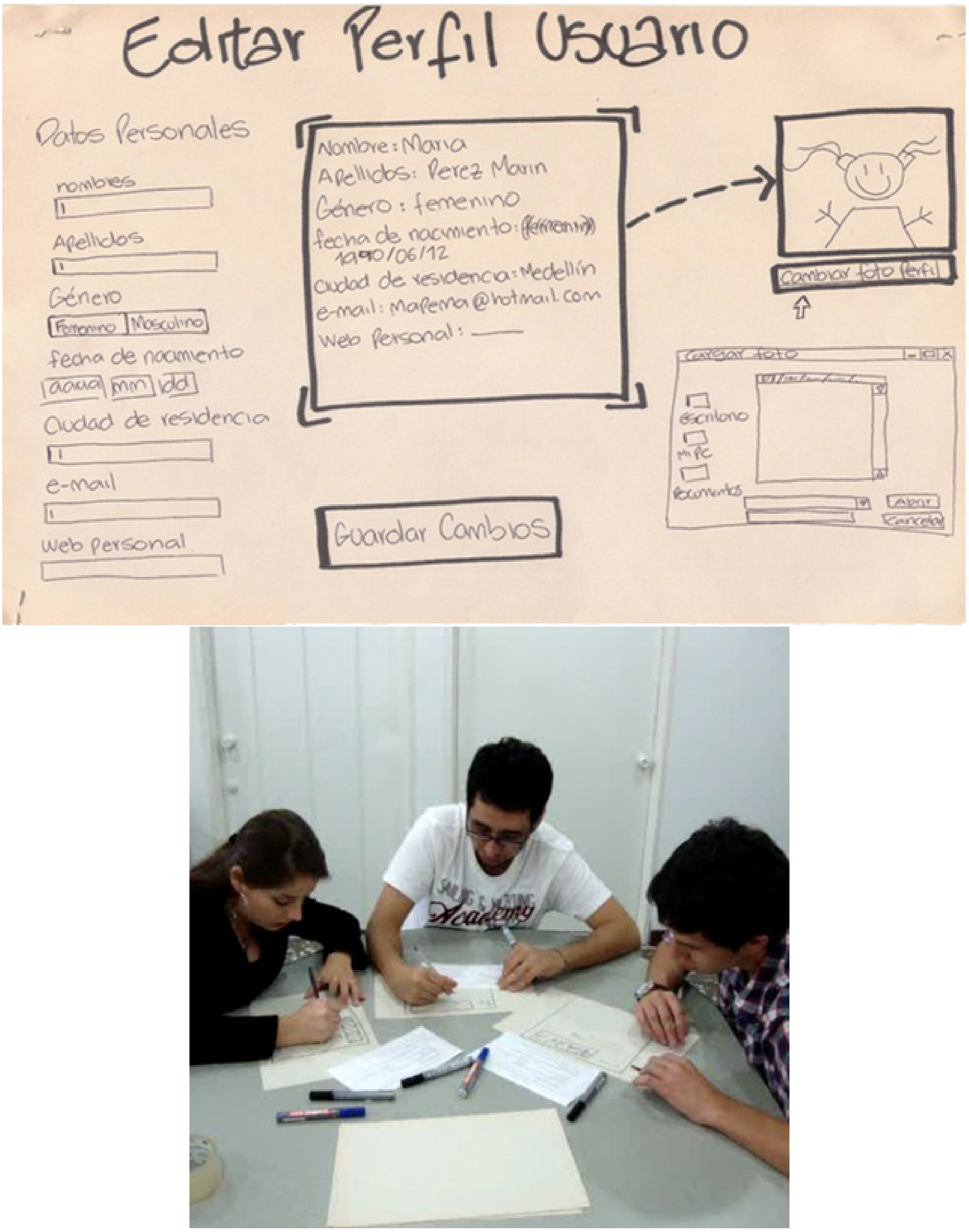

Figure 3.1: Top: Resulting sketch of the manual session. Bottom: Designers at the manual study.

panel, have visual information to be aware of what their collaborators are doing and establish a voice communication.

It is to mention that the shared drawing panel didn't offer pre-drawn stencils or widgets to use in their design. Figure 3.2 shows one designer working remotely connected with his partners in one of the sketching sessions and figure 3.3 shows the resulting sketch with the collaborative virtual environment Teleplace.

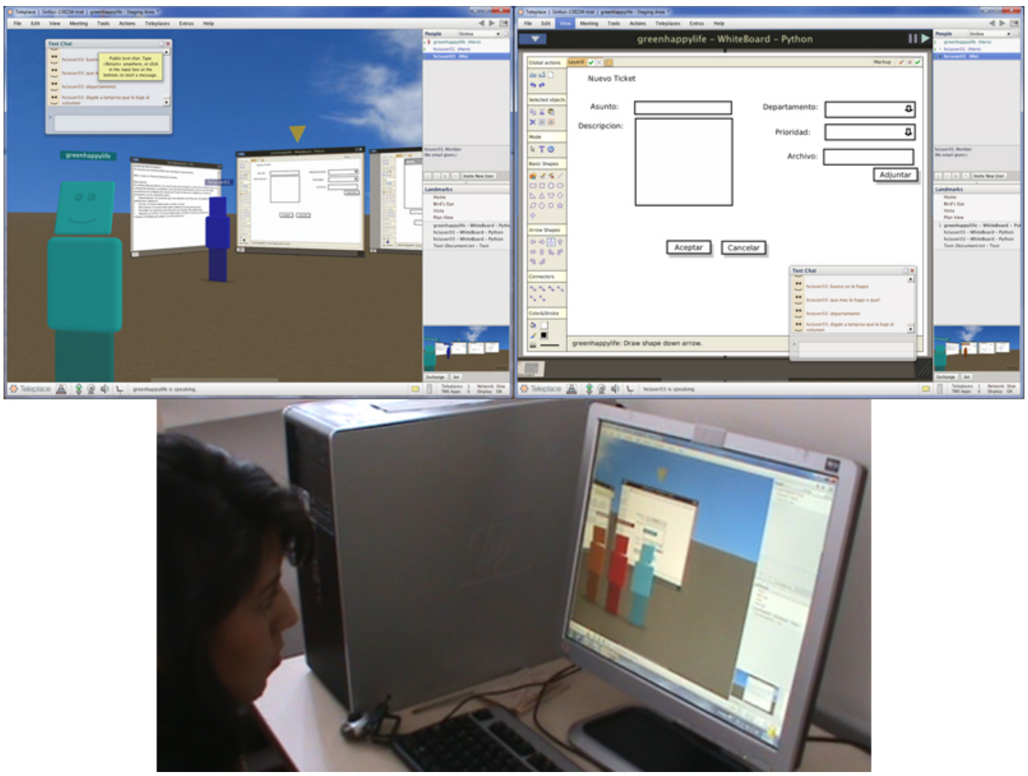

Figure 3.2: Left-top: users into the Teleplace 3D environment. Right: closeup to the shared panel in teleplace. Bottom: User with her partners using teleplace.

Although the tool used for this study offered very good awareness tools for the group they couldn't work simultaneously because only one participant was able to take control. As soon as another participant tried to contribute clicking on any object inside the shared panel, current owner of the focus losed it and his intentions were obstructed. This caused a high level of frustration and discomfort. Another problem observed was that designers found very difficult to draw controls over the shared panel because available tools only included 2D primitives such as ovals, rectangles,

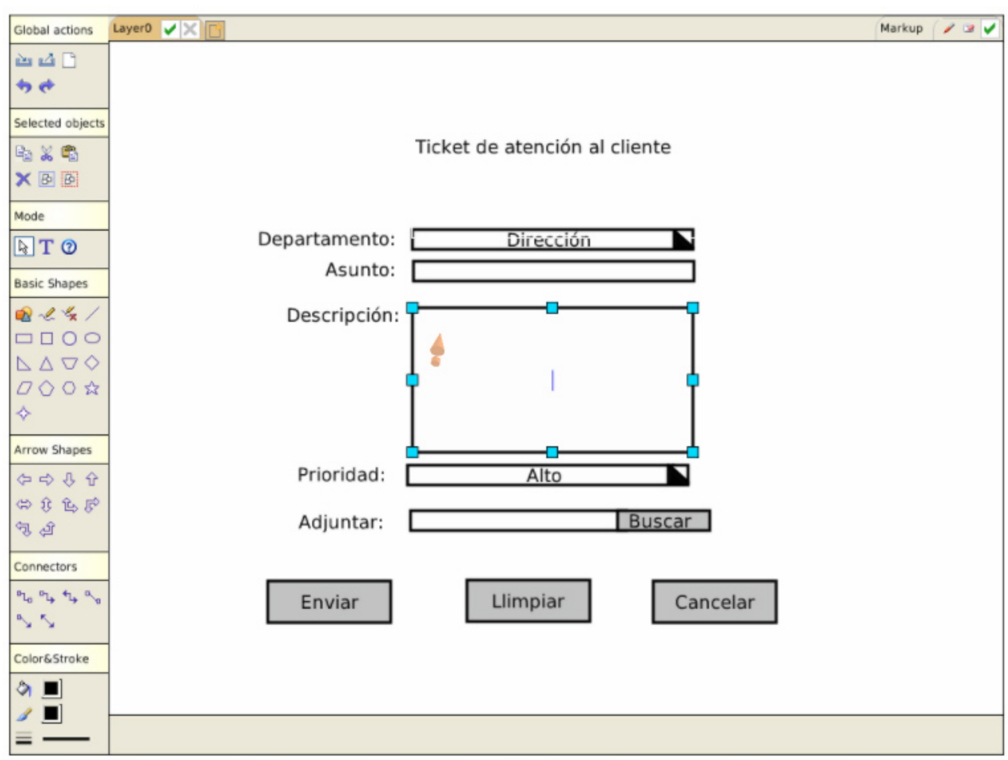

Figure 3.3: Resulting sketch of the design session with Teleplace.

lines, text and the basic operations such as grouping, copying, cutting, pasting and selecting, and additionally, given the fact that the panel was inside a 3D environment, users had problems to move part of their designs to an exact position; as an example, none of the designers could draw a neat combo box.

These observations during the study with the 3D Application showed that designers would definitively work better with:

- the possibility to add or modify a widget to the sketch in any moment without waiting for others to cede the turn
- pre-designed stencils or widgets, so that they don't have to re-draw every control
- more accuracy when selecting and dragging drawings around the canvas

3.1.5 3rd study: Sketching Session with 2D Application

For the Last study we considered 17 groups of designers (50 designers) while they carried out the task using the selected 2D prototyping tool which offered access to a wide set of GUI widgets that ease the prototyping task and a mechanism to share their contributions to remote collaborators via check-in/check-out revisions.

The main downside of this tool, is that communication could only be dealt with a text chat system. Figure 3.4 shows one designer working remotely connected with his partners in one of the sketching sessions and figure 3.5 shows the resulting sketch using the prototyping tool MockFlow.

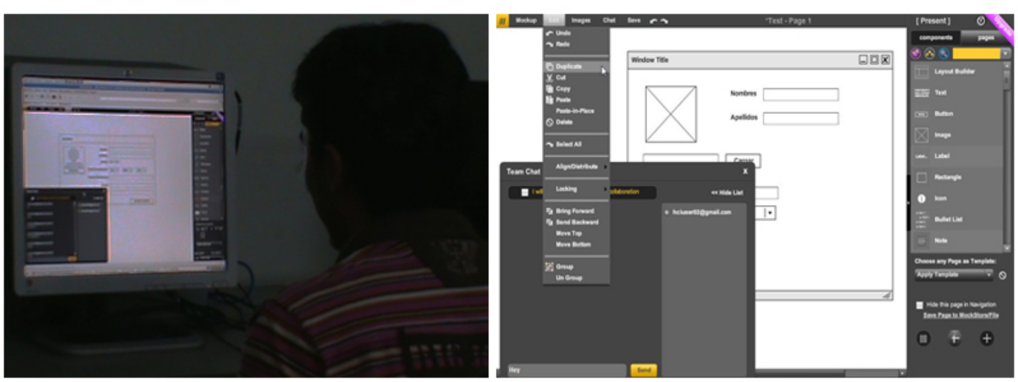

Figure 3.4: Left: Designer using Mockflow. Right: Mockflow's interface.

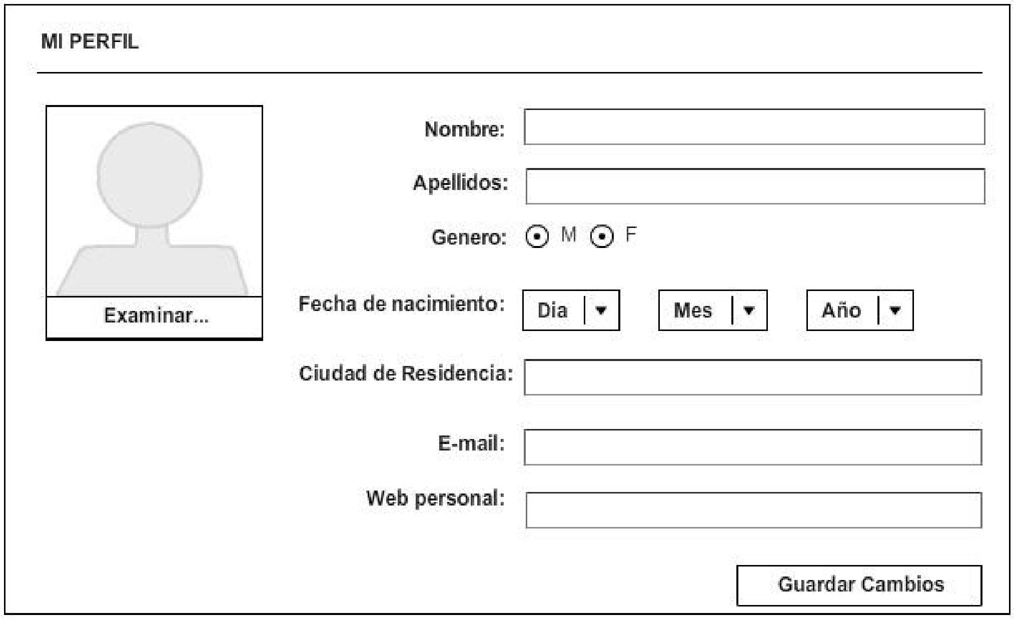

Figure 3.5: Resulting sketch of the design session with Mockflow.

From the observations we saw that users invested much more time trying to communicate by chat with their team than focusing in the real task, sketching. Another problem was that since the tool only offered a real-time collaboration based on check-in/check-out revisions, the experienced turned out very frustrating because none of them were aware of the others work unless they were notified by chat or the changes were submitted. Furthermore, the tool didn't offer a good support for merging changes, so every time the designers received a notification of a new revision available, they tried to merge their changes but they always lost their individual work. Some of them started copying their work before accepting the new update and then pasting their changes and this resulted successful, but tricky.

These studies with the 2D Application showed us that designers needed:

- better awareness mechanisms so designers can see what the others are doing
- the possibility to merge the work they produced isolated with the work produced simultaneously by the others
- a more fluent communication like the audio based one since chat slows down efficiency

3.1.6 Analysis of Studies

From the videos of the studies, we extracted rich information that shows the user's behaviors while doing their tasks. We focused on aspects like time invested on: sketching, communicating, understanding the requested task and idle or distracted. Additionally, we found the same behavior patterns identified by [4] in their Territoriality in Collaborative Tabletop Workspaces work.

Figure 3.6 and table 3.1 show the percentage of time invested on these criteria.

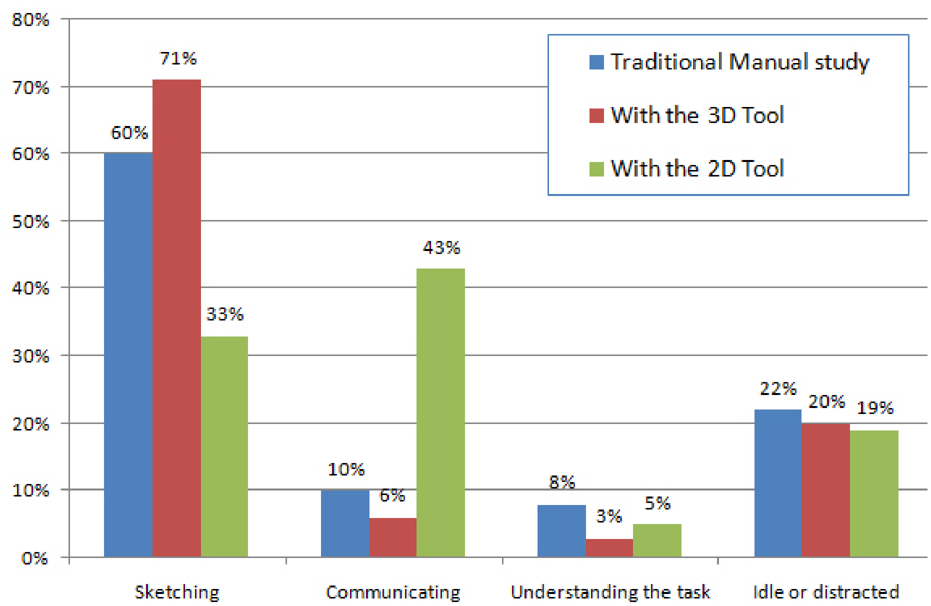

Figure 3.6: Percentage of time invested on the activities of the 3 user studies.

In every study, users invested an average of 1 minute understanding the requested task. The study with the 2D tool required more time for communicating because it only provided a text based chat, so users had to invest more time writing and reading in depth instructions and details, being this task the one that required the most of the time.

The manual study and the study with the 3D tool allowed designers to communicate while they were sketching at the same time, so the exclusive dedication to this task was reduced considerably. The manual study and the study with the 3D tool required more time for sketching because there were no already drawn widgets so users had to draw the whole widgets every time they needed them, being this task the one that required the most of the time in both cases.

	Study		
Activity	Traditional - Manual	With the 3D Tool	With the 2D Tool
Sketching: Finding a widget, adding a widget to the sketch, drawing a widget in the canvas.	60%	71%	33%
Communicating: Exclusively talking, using the chat or the voice system, marking and discussing details.	10%	6%	43%
Understanding the task: Reading/Reviewing requested task	8%	3%	5%
Idle or distracted: Sit idle or doing something different to the task while the user gets the turn.	22%	20%	19%
Average Total Time invested (minutes)	11.45	24.35	17.25

Table 3.1: Percentage of time invested on subtasks perceived in the users studies.

The average of the total time invested on the tasks is shown in figure 3.7.

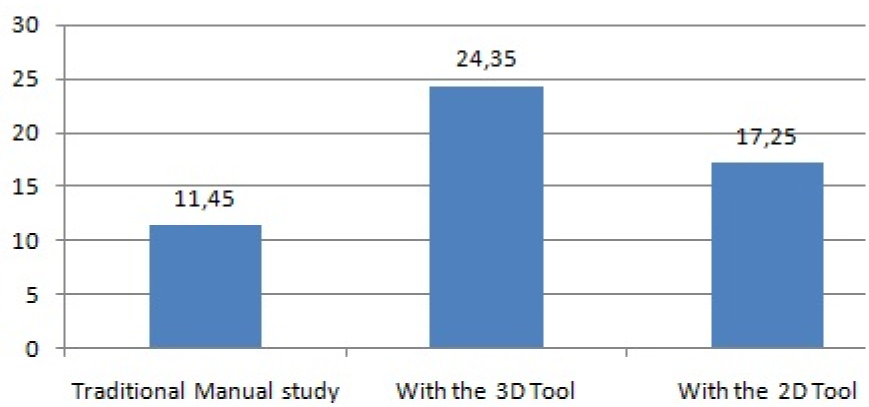

Figure 3.7: Average of time invested on the session in each study.

3.1.7 The Needs

Summarizing the findings in our studies and considering exalted items in the literature mentioned in Chapter 2, we have specified the next funcional, quality, and user experience requirements. Satisfaction of these needs can improve the experience of the group by reducing idle times and increasing efficiency at communicating and sketching.

Functional requirements

- About the management of Collaborative Design Sessions, user should be able to:
  - start (create) a new design session
  - join a design session
  - withdraw from a design session
  - expel a participant in a design session (desired for coordinator of a session)
  - Manage a list of tasks (desired)
- About Sketches management, user should be able to:
  - list existing sketches
  - create, edit, view, close, delete an sketch
  - editing a private sketch (desired)
  - make a suggestion (comment with text and/or audio) over details of a sketch
  - read and/or listen to a suggestion made on a sketch
  - ask another user to observe a detail in a sketch (the requirement also belongs to the following category)
  - see the sketches in a presentation view
  - export an sketch as a window desktop application or web page (desired)
  - enlarge the size of the canvas in a sketch
  - allow user to undo changes made on a sketch
- About the communication between participants, user should be able to:
  - see what others are viewing (Share view). Probably only in 2D, which is when you have a different view to that of other participants. Share It could prove a 3D view and test with this.
  - send a text message to another user (Chat System)
  - voice communicate with another user (Audio System)

  - point to an item on the view of another participant (Telepointer 3D, 2D cursor Share)

- About the management of requirements to be sketched during the meeting
  - manage requirements to sketch during a session (Create, Edit, Delete, List)
  - mark a shared section of the description of a requirement to sketch

Quality Requirements (Non Functional)

- Security
  - Access:
    * each participant in a collaborative design session must be identified
  - Authorization / Repudiation:
    * the system must guarantee that two participants do not attempt to perform conflicting operations on an object concurrently. Eg Two players trying to move the same widget to another position.
    * when attempting to delete a draft to be validated that all participants agree with the deletion.
  - Ownership:
    * the changes made to drafts must register identifying the user that performed. Minor changes to avoid saving changes are defined to be identified:
    * add / Remove Widgets.
    * edit internal properties of a Widget.
    * edit location or size of a Widget (all changes will be as one).
  - Availability:
    * information on collaborative sessions should always be updated in different prospective.
    * the system will operate correctly for a group of up to 6 participants.
  - Integrity:

        * the data that customers see will always be current with a maximum latency of a second. Example, if a user makes changes to a draft that is shared with other such changes will appear in the other in less than a second after making the change.

    - Privacy: We did not consider this aspect since we didn't observe behaviors that require confidentiality when carrying out a collaborative design session

Usability Objectives

- Navigation
    - Restriction: Do not allow participants to leave their jobs to roam the virtual environment.
    - The system should provide an easy way to move from one sketch to another.
    - In the sketches of an interface must offer a tree view display waterways where the structure of the draft.
- Interaction
    - Participants should be allowed to make direct manipulation of virtual objects and avatars, sketches and widgets.
    - Must have hot keys to frequently used tasks.
- User Experience. The idea is that the user task development is drafting a graphical interface:
    - Aesthetically pleasing
    - Useful
    - Enjoyable
    - That supports creativity
- Error handling.
    - Should allow the user to undo actions.

Awareness requirements

Although awareness is part of usability, this feature deserves detailed mention in this project. Some Awareness requirements herein have been identified with the user studies performed, others are proposed in Saul Greenberg and Carl Gutwin work [12].

- visually distinguish who is the coordinator of a collaboration session
- a participant must be able to see what colleagues are watching their work
- participants should be informed when another participant has joined them in the design of a sketch
- at leaving observation or work with a user in a design, a transition to the position of original work should be seen
- offer an alternate view of the objects added to a sketch (tree)
- offer a view of the cursor of the other participants during the editing of a sketch
- visually inform when a participant directly manipulates a widget while editing a sketch
- visually inform when a participant directly manipulates an object in 3D scene
- users should be informed of the load percentage of the stage, objects and sketches.
- the user information that should be visible to other users is: Name, Login, Gender, Status (Away, Available, Busy), Role (Coordinator, Participant), Color (of its cursor and pointer) and contact information

Reliability - Fault tolerance

- In case of falling, the server must ensure that the condition of artefacts is persistent and can be retrieved in a subsequent session.
- If you drop a client connection the system must do the equivalent to when a user leaves a collaboration session.

Efficiency

- Load and display an outline for preview or editing should not take more than 3 seconds.

- The connection to the server should not take more than 30 seconds, in case of the 30 seconds the connection will determine inaccessible.

- Entering a design session and load the 3D scenery should not take more than 5 seconds.

Portability

- The application must run on Windows XP, Vista and 7.

Version Control

- it should be allowed to load a previously saved version of a sketch.

- it should be allowed to see the history of changes made by a participant on a sketch.

3.2 System Architecture

In order to specify a complete tool that satisfies the real needs of designers, we considered the functional and awareness needs identified by our 3 user studies, the functionalities offered by the competitors software considered in section 2; the theories about Workspace, Group and Artifact Awareness [12], [Tom-97]; theories about Territorial Coordination [5]; studies about tools for Awareness in Groupware, Awareness Models for Collaborative Learning [29] and Success Cases of Telepresence applied to education [37].

The system developed is a common Client-Server application using Java technologies. The server is in charge of the management of the shared information used during a sketching session, transmit the awareness information from clients to clients using Kryonet [38] framework and persistence of the artifacts developed in every session using the Java Persistence API [39]. Every client is in charge of presenting the information about Awareness, the Chat and the Sketching system, broadcasted by the server. This visual information is presented using Java Swing and JMonkeyEngine [40]. The conference is started among clients using IP addresses without connection to the server using Java Media Framework [41].

Figure 3.8 describes the APIs, Technologies and Frameworks used to structure WeSketch.

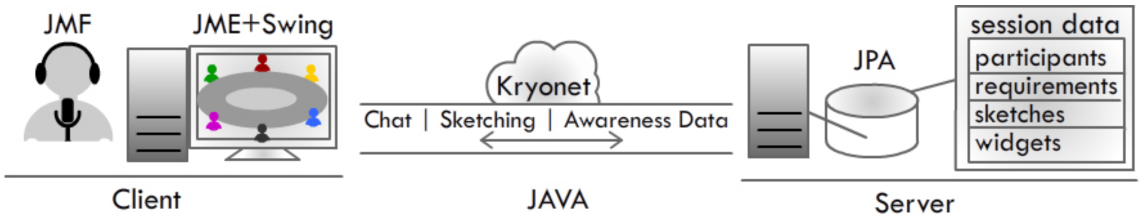

Figure 3.8: System Architecture

3.2.1 High Level System Architecture

The next figure shows the componentes and subsystems assigned to the different layers of the system's architecture that make part of the solution.

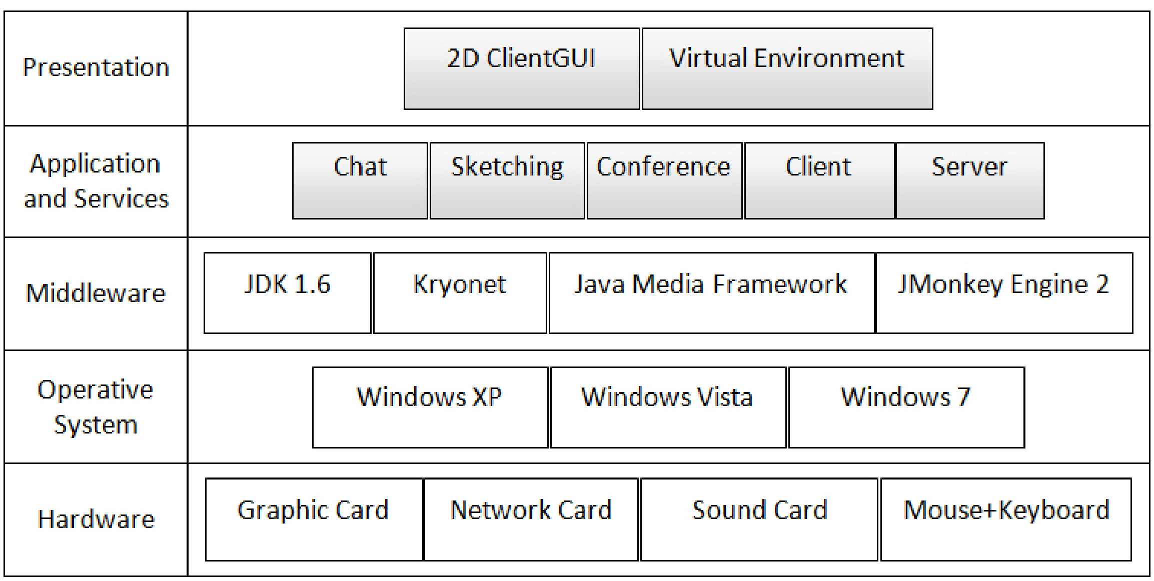

Figure 3.9: High level architecture of the System

The construction of WeSketch consisted on the implementation of the subsystems in the Application and Services, and Presentation layer. The remaining components are applications and open source APIs or frameworks that didn't need to be built.

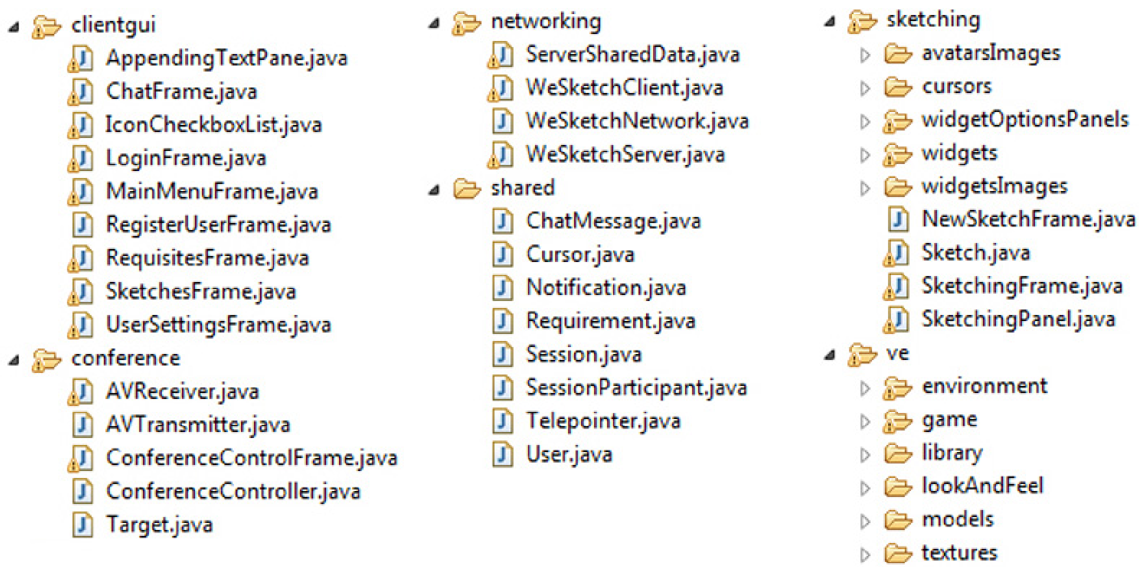

Figure 3.10: Packages and Source code of WeSketch

WeSketch was divided in 6 main packages. In figure 3.9 the Client and Server Subsystems appeared separated, for the final implementation we decided to group the main classes related to the Kryonet framework into a package named **networking**. Additionally, figure 3.9 mentioned the Chat system, but, since it's complexity is low, we have distributed related classes into more appropriate packages such as **clientgui** and **shared**. The **shared** package contains all the classes of objects that are sent through the net, these classes represent the information that is shared among the server and the clients. The rest of the subsystems are contained in a package with the same name given in the high level architecture.

Figure 3.10 shows a complete list of the source code files contained in the first level of packages, and figure 3.11 highlights the packages implemented and the most important classes of them and its relationships with the frameworks and APIs used.

3.2.2 Detailed Characteristics of Every Subsystem

Communication System

For the Chat System, we have provided an interface that allows designers to communicate with every participant in the design session and separately communicate with the current co-editors of

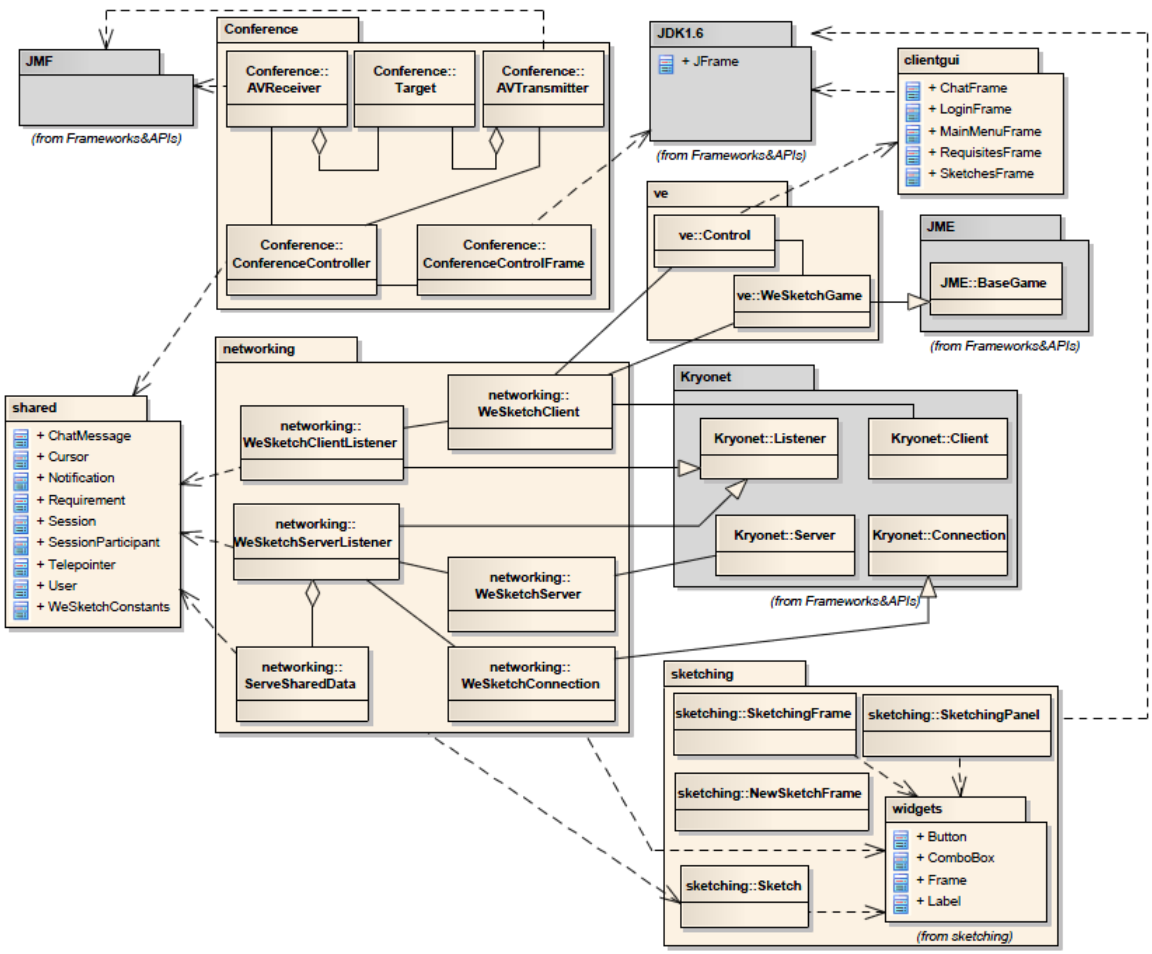

Figure 3.11: WeSketch's most important classes and its relationships with Frameworks and APIs. The widgets package contains 30 widgets.

a Sketch. In this way, we can simulate the behavior shown by the users while they carry out the manual face to face sketching task, in which groups are made and some discussions are private to the group and some public to the entire group. Additionally, as requested by our users, the chat informs with a Skype or Messenger like visual sign that there is a new message in any of the chats, and the chat window can be manipulated in such ways that won't affect the user's job.

The Voice-Conference System allows a selected group of participants to carry out a voice conference and add or remove participants to and from the conference in any time. A conference control frame has been provided to easily allow these functionalities. Figure 3.12 shows a zoom of the chat and the conference control frame.

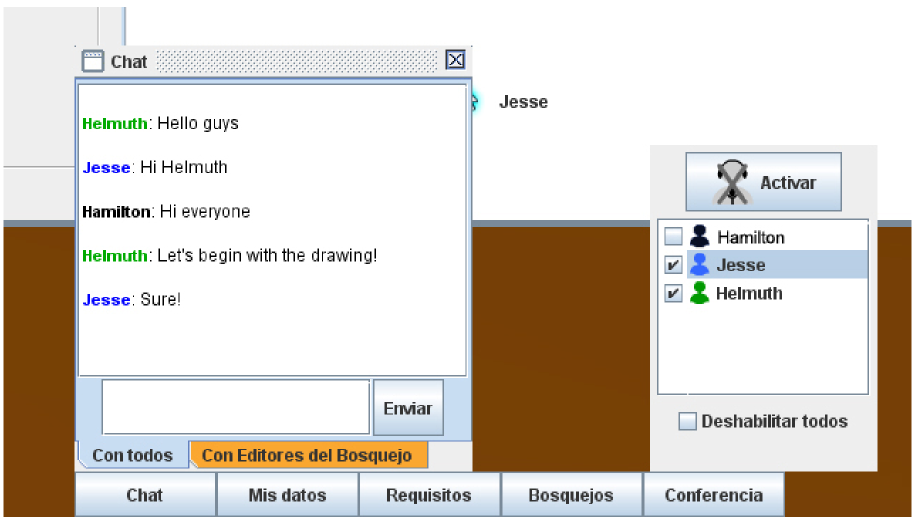

Figure 3.12: A zoom of the chat and the conference systems

Sketching System

The Sketching System provides a wide set of preconfigured GUI widgets, famous in most of the 2D competitors applications. In order to add a widget to a sketch it is only necessary to drag and drop one of the widgets provided in the widgets toolbar, and to edit the settings of the widgets a quick edit functionality containing the most common attributes has also been provided. The great novelty of this Sketching System is that allows co-editors of the sketch to contribute to the sketch (add/remove/modify) in any minute without turn taking and with a real time feedback of their actions. This quality deals with the problem mentioned by our users when they say “I don’t know what others are doing”, “I have to wait until they notify me of their changes” and “I can’t participate when another one is using the shared board”.

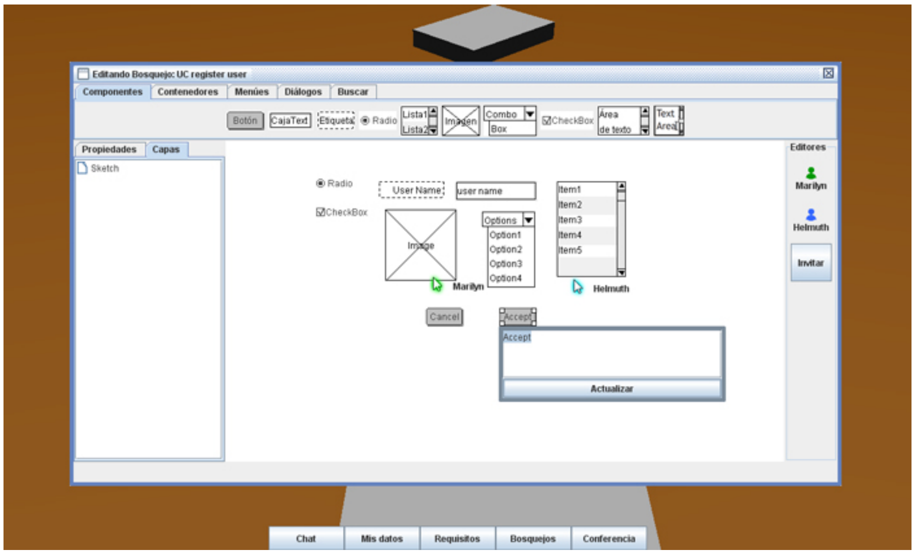

Figure 3.13: The sketching frame. 3 persons are editing the sketch at the same time

Awareness Tools

The Awareness Tools provided with the system are those related to available participants and artifacts in the workspace, remote representation of participants’ actions, representation of participants’ state and participants’ associations. The tools developed are: 2D telecursors (3.13), 3D telepointers, Avatars animation, Artifacts location and Territorial differentiation, as shown in figure 3.14.

Considering Metaphors, in order to assure that the developed tool maps the reality and the fluency of the traditional manual sketching task, we have tried to offer a 3D virtual environment where users can sit around a round table and use their personal and shared spaces as they would manually. The view of every user can be managed to simulate the movement of the user’s heads and this movement is transmitted to the others. Since pointing at objects and details of the artifacts is also frequent during the task, users of our system can use a 3D telepointer to point at any object in the virtual environment and a 2D telecursor to indicate details in a sketch. Users can see and access the artifacts over the table as they would in the real environment, but additionally they can have an organized view of the sketches and information about their authors and collaborators.

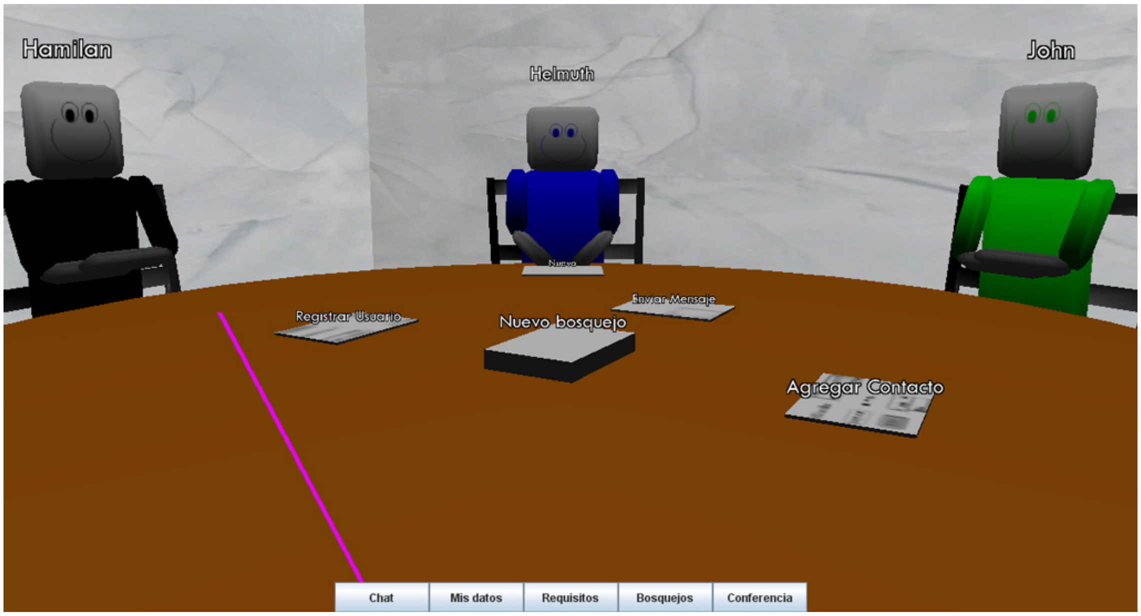

Figure 3.14: The Environment and Telepointer as awareness tool. Sketch in use by a participant appears close to his/her avatar. Unassigned Sketches appear around the white sheets pile.

3.2.3 Design Considerations

Some of the design decisions for the system didn't come out from requirements identified in the studies or features supported by existing competitor applications. Those ideas were extracted from literature and are product of informal usability discussions conducted with usability experts, GUI designers and students.

Number of users allowed in a prototyping session

Observing interaction around a table we noticed that there is not enough room to accomodate a lot of participants. Additionally, [12] defines the number of participants of a small group that can work together around a table as a maximum of 6. We decided to build our virtual room with a round table that allows up to 6 participants, see figure 3.15.

Location of Virtual Sketches

We decided to represent sketches as virtual white sheets located around a table, trying to simulate the real appearance of a design session when it is conducted traditionally. Considering the distribution and positioning of artifacts over a physical table according to the territoriality in tabletops [3, 4] we have located unassigned sketches around the center of the table, in the space denominated as public area and sketches being used are located in the private area, see figure 3.15.

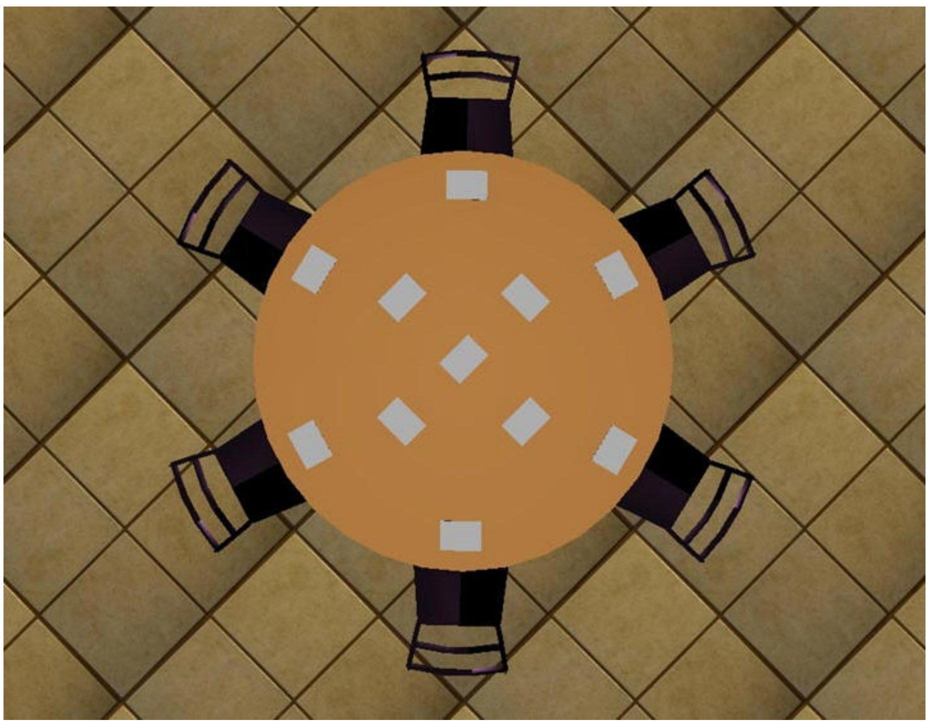

Figure 3.15: Virtual round table with support for 6 users and Sketches located in public and private space

Chat System appearance

One of the most common complains of users during the third initial study was related to the chat system. Users said that the chat box was extremely disruptive, appearing in the middle of the screen and that when the chat box was open they didn't know when a new message had arrived. Considering this we decided to provide a closable and movable chat frame that users could manipulate in an easy way. Additionally, we considered the phased notification philosophy proposed by [6] which is very common in current instant messaging systems and gave our chat box the blinking behavior to notify that a new chat message had arrived.

Sketching System and Concurrent Control

Users expressed that they would like to contribute to the designs in any moment and we wanted to satisy that. Even so, we needed to provide a mechanism that informs that an artifact is being used without restricting the intention of manipulating it in any moment. With this we tried to avoid troublesome concurrent manipulation of elements, like for example two users moving a widget in different directions. To solve this, we gave the widgets the possibility to inform who has selected it and thereby who is transforming it. Selected widgets exhibit controls around their border in the same color of the user who is using it, see 3.16. This solution is similar to the one proposed by [12].

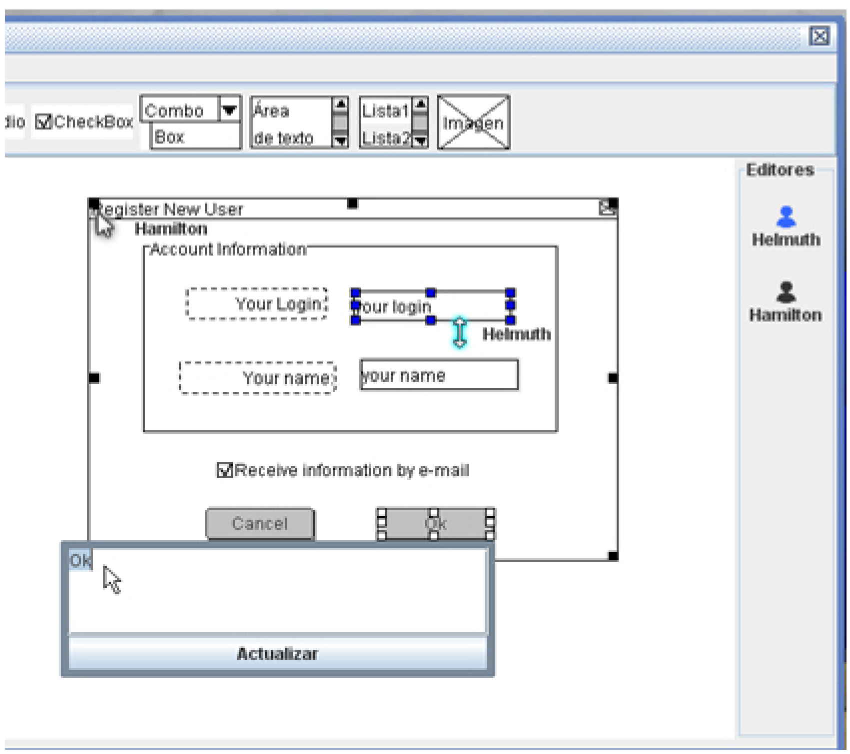

Figure 3.16: Concurrent manipulation of widgets and visual cues. Three users are sharing the sketch, each of them is changing one widget. Modification controls of every widget have the same color of the user owner.

Colored Telepointer Rays

Although telepointer rays are embedded in current 3D virtual environments and seem to work very well, we confirmed it's validity and functionality based on [8]. We provided colored visual rays comming out from avatars' heads and pointing at avatars or other 3D objects in the virtual environment. This gave more information to the users about focus and intentions of the team members. Although [8] also mentions eye contact as important, this kind of interaction is not provided since the task considered as main problem does not require this kind of contact, as we saw in our manual study.

3.2.4 Network Protocol

Nowadays, many communication protocols among computers exist, many of them are oriented to TCP-IP communications. WeSketch's network communication is one of them. Based on Kryonet Framework and it's advantages, we have designed a a Client-Server application using well defined objects for transmission and listener threads in every side that pay attention to the use of the communication channels (sockets).

Since our intention was to provide a Real Time collaboration in our collaborative virtual environment, we had to think carefully about the objects that we were going to transmit, their information and datatypes, and when to transmit any of them. Some initial designs were proved and after some improvements, this is what we got.

Information transmitted through the network

- Cursor: information about the type and position of a participant's cursor in 2D.

  Data: type, participantIndex, x, y and sketchId.

  When the server receives a cursor from an user it sends the cursor only to the co-editors of the sketch identified by sketchId

- Telepointer: information about the orientation of a participant's attention in 3D.

  Data: userIndex, direction

- WidgetData: information about a Widget Data: id, sketchId, location, size, text, imageRoute, enabled, zOrder, groupId, remotelyFocused, title, iconType, value, selected, font, alignment, widgetType To avoid the possibility of generating the same id for a widget by different users at the same time, we assigned the id the value resulting of the long representation of current time and date with the creator's index appended.

- ChatMessage: message sent by an user to the session or an specific user.

  Data: text, senderIndex, privateToId, groupId Since private messaging is allowed, it is possible that the server sends the ChatMessage only to user identified by privateToId. If groupId is different to -1 the message is sent only to a group o participants who are sharing the sketch identified by groupId.

- Sketch: information about a sketch and the widgets included in it.

  Data: id, title, description, author, requisiteRelated, widgets list, co-ediditors list

- Notification: a message identified by a signal that contains a set of parameters. The different notifications types and their parameters are described in table 3.2:

SIGNAL	PARAMETERS
NO_ROOM	-
DELETE_WIDGET	sketchId, widgetId
REMOVE_PARTICIPANT_FROM_SKETCH	sketchId, participantId
REMOVE_PARTICIPANT_FROM_SESSION	participantIndex
DELETE_SKETCH	sketchId
DELETE_USER	userId
LOGIN_ALREADY_USED	-
DELETE_REQUIREMENT_FROM_SESSION	requirementId
DELETE_REQUIREMENT_FROM_SERVER	requirementId
USERLOGIN_ALREADY_USED	-
ADD_PARTICIPANT_TO_SKETCH	sketchId, participantId
WRONG_PASSWORD	-
REMOVE_TELEPOINTER	participantIndex
MOVE_TO_FRONT	sketchId, widgetId
MOVE_TO_BOTTOM	sketchId, widgetId

Table 3.2: Notifications and parameters

- SessionParticipant: information about a participant in the collaboration session.

  Data: login, connectionId, ip, chairIndex, password, userId

- Requirement: description of a requirement to sketch during the session. Data: id, title, description

- User: login information Data: id, login, password, name, lastname, email, phone, sex
- Session

This information is sent over the network during different states of the client's participation.

It should be mentioned that communication between client and server is asynchronous, since messages sending can be invoked from everywhere in the client application but answers are only received by a listener thread. In the server side, there is only one listener thread receiving every incoming messages. The server send information only as response to specific objects received from clients.

Before starting the listening thread, the server begins loading information like registered users, saved requirements and previously created sketches with their internal widgets. Al this information is saved into an object known as the **sharedData**.

The table 3.3 details the states and transmission of clients to the server and the only one state of the server.

As mentioned above, clients receive objects from the server in an asynchronous way. Two threads have been provided and only one of them works at the same time. The first Thread is the **Login State Listener**. This thread only receives objects such as:

- Notification: just in case the server couldn't receive the client because of an incorrect login or password, the login is already in use or there is no room to accept the user.
- Session: if the client logged in correctly.

When this thread receives the Session object its execution ends, giving way to the Session Listener thread to start. The Session Listener thread receives objects such as:

- Cursor: when server informs that another participant is moving its cursor in the shared Skething Frame
- Telepointer: when the server broadcasts participants pointing at objects
- WidgetData: when changes occur to widgets or a new widget is added to a sketch
- ChatMessage: when client is destinatary of the a chat message

CLIENT	SERVER
	LISTENING
connects	registers connection
LOGIN STATE	
begins Login state listener	
creates and sends SessionParticipant	responds with: • if valid user then send Session object to the sender client SessionParticipant object to the other clients • if not valid user then WRONG_PASSWORD Notification object • if login is in use then LOGIN_ALREADY_USED Notification object • if room is full then NO_ROOM Notification object
IN SESSION STATE	
ends Login state listener and begins Session listener	
client could send any of the following in any moment:	
private ChatMessage object	send the ChatMessage to specified user
global ChatMessage object (to co-editors)	send the ChatMessage to everyone
sends Telepointer object	sends the Telepointer object to the others
sends REMOVE_TELEPOINTER Notification	sends the Notification to everyone
sends Sketch object	sends the Sketch object to everyone
sends ADD_PARTICIPANT_TO_SKETCH notification	sends the Notification to everyone
sends Requirement object	sends the Requirement object to everyone
IN SKETCHING STATE	
client could send any of the following in any moment:	
local ChatMessage object (to co-editors)	send the ChatMessage to specified sketch co-editors
sends Widget object	sends the Widget object to everyone
sends DELETE_WIDGET Notification	sends the Notification to everyone
sends MOVE_TO_FRONT Notification	sends the Notification to everyone
sends MOVE_TO_BOTTOM Notification	sends the Notification to everyone
sends REMOVE_PARTICIPANT_FROM_SKETCH Notification	sends the Notification to everyone
sends Cursor object	sends the Cursor to co-editors in the same sketch as the sender client
disconnects	sends REMOVE_PARTICIPANT_FROM_SESSION Notification to everyone

Table 3.3: Communication protocol during different states of clients

- Notification: when a general purpose Notification is broadcasted
- Sketch: when a participant creates a Sketch
- SessionParticipant: when a new participant has arrived
- Requirement: when a participant has created a new Requirement

None of the above objects require a response to the server, so actions affect only the virtual environment, the sketching frame or the non visible information of the design session.

Chapter 4

Experiments

We wanted to verify that WeSketch really improved the prototyping task so we postulated the next hypothesis: "WeSketch allows a small group of designers who are geographically dispersed to be more efficient and feel more motivated during the GUI Sketching task". In order to validate it, we designed the following experiments.

4.1 Description

We formulated 3 tasks consisting in 3 requirements of different systems to be prototyped. We gave clear details about the information that was required for every GUI. 37 fifth-year students of a Systems and Computers Engineering program were recruited for the experiments. 11 groups of 3 users and 2 groups of 2 users were formed. All of them had a background of at least one semester in Usability and GUI design.

Users' groups were instructed to design the GUIs using a different tool for every task. In a previous studies, we chose MockFlow as the best available prototyping tool for groups, and Teleplace as the best available 3D collaborative virtual environment for a rich collaboration. WeSketch was the third application. In order to reduce the impact on the performance using a tool caused by the learning process when users advance from the first task to the last one, we sorted equitably the order of the tools to be used by each group of users. At least one student from every group was videotaped during every task. More than 39 videos were analyzed and measures on effectiveness, iteration, productivity and communication fluidity were taken.

4.2 Results

Figure 4.1 illustrates the measures that support that WeSketch required the shortest Task Completion Time, being the most effective of all. We observed that users made fewer changes with Teleplace, this might be due to the difficulty of this subtask with this tool, and this lack of changes was reflected on the less quality of its resulting sketch. Additionally, Teleplace exhibits a bigger messages transmission, this is due to the audio conference easiness and the extended time that users needed to complete the task, by contrast, although WeSketch also offered audio conference, it has the least messages transmission; again, this is related to the used time. Figure 4.1 shows how WeSketch has achieved reduction of the wasted time being idle or wandering resulting from the users turn taking to manipulate the shared artifacts and application's defects that demand users to wait for a response. In this figure it can also be seen that using WeSketch, most of the time is used in the solution of the main problem, that's to say, sketching (adding and configuring widgets in a sketch). And finally, we can see that the time dedicated to exclusive communication is reduced with WeSketch since users can contribute to the task in any moment, reducing the chances that they dedicate only to give written or spoken instructions and opinions, and task lasts fewer.

A Survey with several Likert Scale and 2 open questions (1. Why? and 2. What would I do better?) was also applied to the users after every task was completed. The questions and user's scores are shown in 4.3.

From these images, we can identify that WeSketch is perceived as an improvement for the prototyping task and exceeds Teleplace and MockFlow in many ways. But we can also see that there are still some improvements that could be made in order to give a better experience to designers, such as: Offering an easier way of manipulating objects and Helping users to feel in total control of the tool.

From the open question "What would I do different?" the next suggestions were common among users:

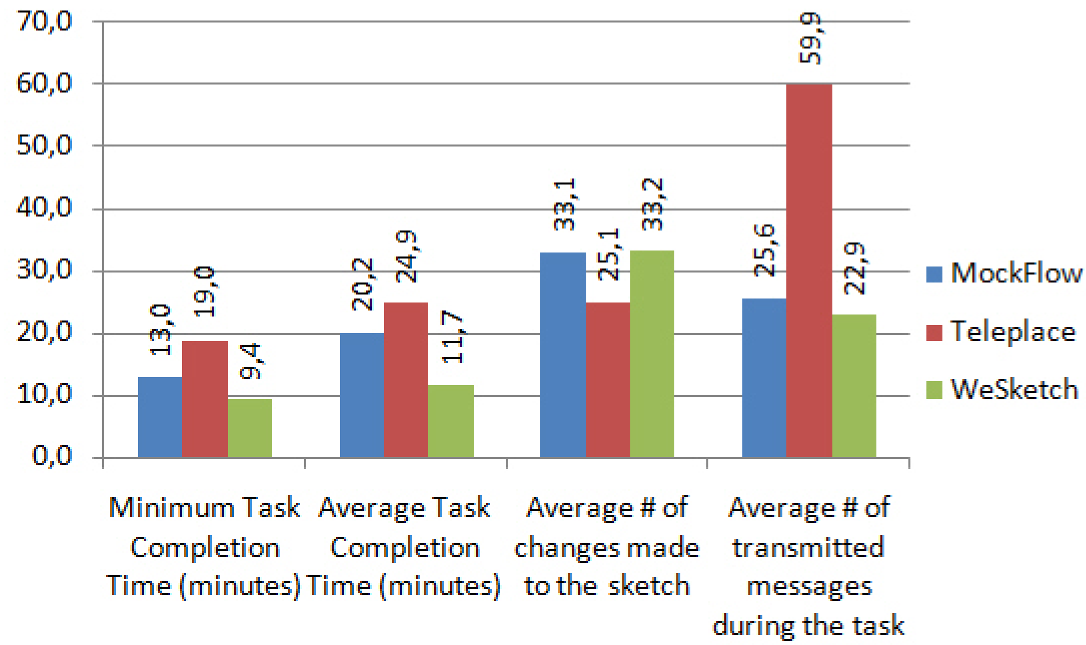

Figure 4.1: Measures related to Task Completion Time, Productivity, Iteration and Communication Fluidity

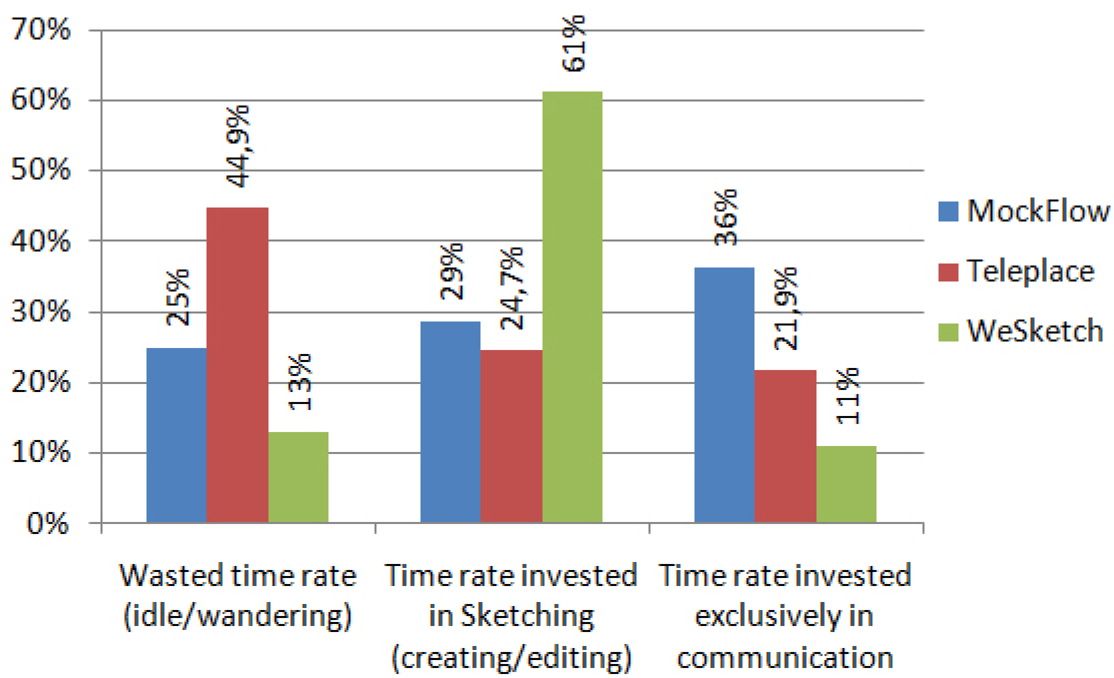

Figure 4.2: Measures related to Productivity

	Teleplace					MockFlow					WeSketch				
	Totally Agree	Partially Agree	Neutral	Partially Disagree	Totally Disagree	Totally Agree	Partially Agree	Neutral	Partially Disagree	Totally Disagree	Totally Agree	Partially Agree	Neutral	Partially Disagree	Totally Disagree
1 Collaboration was Effective, Fluid and Clear	26%	16%	32%	21%	5%	24%	33%	21%	15%	6%	72%	22%	6%	0%	0%
2 It was easy to make changes to the sketches	16%	5%	16%	42%	21%	30%	30%	9%	26%	4%	59%	25%	16%	0%	0%
3 Time was used productively	5%	21%	21%	37%	16%	27%	21%	27%	12%	12%	72%	22%	6%	0%	0%
4 Communication with the group was easy	33%	39%	11%	17%	0%	18%	33%	18%	21%	9%	84%	16%	0%	0%	0%
5 I found the tool Pleasant, Fun, Entertaining	5%	21%	42%	21%	11%	33%	33%	24%	9%	0%	71%	29%	0%	0%	0%
6 Doing the task was Efficient, Comfortable and easy	5%	21%	21%	42%	11%	33%	27%	21%	18%	0%	53%	38%	6%	3%	0%
7 I felt in total control of the tool	5%	26%	11%	32%	26%	24%	18%	30%	12%	15%	50%	47%	3%	0%	0%
8 I could use the tool as I use some others	11%	37%	21%	21%	11%	34%	38%	19%	9%	0%	52%	32%	13%	3%	0%
9 Managing objects used for the task was easy	16%	11%	26%	26%	21%	73%	21%	3%	3%	0%	63%	22%	13%	3%	0%
10 It was easy to begin the collaboration with my group	37%	32%	16%	16%	0%	45%	33%	6%	12%	3%	69%	31%	0%	0%	0%
11 I participated in the task in the way I wanted	21%	26%	26%	16%	11%	21%	33%	21%	6%	18%	77%	19%	3%	0%	0%
12 I participated in the task everytime I wanted to	21%	26%	32%	16%	5%	24%	24%	18%	15%	18%	75%	22%	3%	0%	0%
13 I was aware of what my partners were doing	58%	37%	0%	5%	0%	15%	18%	27%	24%	15%	90%	10%	0%	0%	0%
14 Information about others' actions was enough and clear	37%	47%	11%	5%	0%	36%	6%	30%	12%	15%	69%	25%	3%	3%	0%
15 I was aware of the state of the task and it's progress	68%	26%	0%	5%	0%	36%	15%	30%	6%	12%	84%	16%	0%	0%	0%

Figure 4.3: Experiment survey results

- Provide a contextual help for tools
- Allow more interactions among participants
- Provide a more pleasant appearance for the tool and the available widgets
- Improve response to some mouse events such as double click (conflicting with the JME API)
- Adding more hotkeys and functionalities in the widgets popup menu
- Allow aligning widgets with others in the sketch and/or fit them to a grid.
- Correct a few bugs of the application

Additionally, some good comments of the users about the experience using WeSketch were:

- "Audio communication was very clear"
- "We could work together at the same time"
- "Working concurrently is fun"
- "I always knew what the others were doing"
- "I don't lose my work" (MockFlow's problem)
- "The tool is proper for the task"
- "I participated actively in the task when I wanted"
- "Time was better invested and the task was completed very fast"
- "The interaction is similar to the manual one"
- "I can see the progress of the task and be aware of what is left to do"
- "The application is easy to learn because is very simple"

Some of these comments help us to determine that designers felt more productive, participative and aware of the progress of the task as well as they appear more motivated and pleased with WeSketch as prototyping tool than with MockFlow and Teleplace.

Chapter 5

Conclusion

As a result of the conducted studies we identified that many of the needs that GUI designers have while they carry out the prototyping task are not satisfied completely by the best existing applications. The identified needs and some theories related to group work such as workspace, team, artifacts and task awareness allowed us to specify and develop a tool that satisfies the most critical needs of physically separated GUI designers, increasing their productivity and willingness.

The most relevant needs were implemented so that we could validate the proposed hypothesis "WeSketch allows a small group of designers who are geographically dispersed to be more efficient and feel more motivated during the GUI Sketching task".

A set of experiments were carried out and the hypothesis was validated with positive results.

Different measures of time and subtasks were given, comparing the performance of the designers while they used three different tools, and WeSketch showed that the Task Completion Time was dramatically reduced. In addition, users expressed comfort with the tool and openly suggested improvements that could enhance the quality of the tool.

As result of the experiments, WeSketch was found to be useful for the User Centered Design process since it allows managing a group of requirements whose GUIs must be prototyped and run expert usability revisions over the produced artifacts.

Some improvements and desired functionalities were identified, showing that the tool offers better results and can be included in other stages of the UCD process, such as Coding and User Testing.

5.1 Future Work

We found and our users expressed that WeSketch could be more pleasant and useful if we provided some additional features such as:

- adding links between sketches and providing a navigational presentation view of the sketches
- providing more new GUI components such as Accordion, Playback controls and Maps
- allowing colorful and textured designs
- exporting the produced sketches to a common programming language like Java, C# or similar
- managing different versions of the sketches and allowing the retrieval of past revisions
- allowing changing appearance of the widgets, like the look & feel in some desktop applications
- sharing the exact view of others

Some of these suggestions had already been identified but were not yet implemented. Most of them are being developed at this moment and will soon be part of WeSketch. Our experiments focused only on discovering how users increased efficiency and comfort while they sketched together. We think that we could find worthwhile information if we studied developers when they exploit all the functionalities offered by WeSketch.

Currently, WeSketch informs the union of participants to co-edit a sketch by orienting avatar's heads to the shared artifact and pointing their 3D telepointers to the sketch, but we think that there may be a better solution to illustrate this, so we let this open to discussion.

And finally, the developed tool could also be expanded to fit other tasks where groups need to work concurrently and aware of their environment, their group and their resources.

Appendix A

Glossary

- Adaptability: ability to adjust easily to a new environment or different conditions [33]
- Affordance: in the context of HCI, it indicates the easy discoverability of how an object may be interacted with [42]
- Articulation of Collaborative Work: coordination of collaborative work in terms of "Who should do what, how, when and where". [43]
- Augmented Reality/Environment: kind of reality that combines information of the real world with information generated by a computer and then displayed to the user.
- Awareness: the knowledge of the activities of the others that gives a context for the individual ones [26]
- Collaborative Design: collaborative execution of the design phase in an engineering process [44]
- Coupling: association of two or more participants during a collaborative task.
- Grid and snap mode: interaction technique that automatically attachs elements moved by a user to the limits of a grid. This technique is very used for puzzle games and design applications.
- Group Centered Design: Process that considers needs of the groups to develop a high quality groupware solution [32].

- Head mounted display: display device, worn on the head or as part of a helmet, that has a small display optic in front of one or each eye.
- Mockup: preliminary design of a product, usually produced for teaching, discussion of requirements and to know first impressions [33].
- Prototype: interactive preliminary design of a product, usually produced for usability testing [33].
- Real-time collaboration: a collaboration in which every participant has immediate feedback of the actions executed over a shared environment and artifacts.
- Six degrees of freedom: quality of a virtual object that allows its translation and rotation over the x, y and z axes.
- Stencil: configurable template used to produce a design [33].
- Telepointer: visual representation of a user pointer transmitted to participants of a collaboration used to inform interest in an specific element.
- Widget: piece of software that represents a user interface control like a button, a label, a combo-box, etc., present in windows systems.
- Wireframe: basic visual guide used in web design [33]. Preliminary design of a graphical user interface, usually produced for discussion of requirements.
- Workspace: environment where people develop their tasks [33].

Appendix B

Tasks required to Designers

The next paragraphs were the tasks given to each person in the groups of designers during the experiments for the hypothesis validation.

B.1 Task 1

A group of graphical user interfaces is required for the following requirement:

FR01. Register a product in supermarket selling catalog. The information that the user will give for the product is:

- Code: numerical
- Name: string
- Description: long text
- Product type: (Cleaning, Food, Clothing, Liquor, Groceries, Toys)
- Is the product free of taxes? yes or no
- Upload a picture of the product: image (display a picture frame for the picture)
- Users should be able to cancel the task

B.2 Task 2

A group of graphical user interfaces is required for the following requirement: FR02. Sign up for an online technology forum Users will provide the following information:

- Username (Login)
- Password
- Confirm password
- E-mail
- Country (Colombia, Venezuela, Ecuador, Perú, Brazil, Panamá)
- Upload a picture of the user/Avatar (show picture frame by default)
- I accept terms and conditions
- Users should be able to cancel the task

B.3 Task 3

A group of graphical user interfaces is required for the following requirement: FR03. Process a form for Questions, Complaints or Proclaims in a health system The information that the user will fill in the form is:

- Id
- Name
- Last name
- Address
- Phone
- E-mail
- City: (Armenia, Pereira, Manizales, Cali, Medellín, Ibagué)

- Request type: (Question, Complaint, Demand, Proclaim)
- Description
- Users should be able to cancel the task

Appendix C

Survey

This is the survey applied to groups of designers after they completed every task during the initial user studies and during the final experiments for the hypothesis validation.

#	Tool:	Totally Agree	Partially Agree	Neutral	Partially Dis-agree	Totally Dis-agree
1	Collaboration was Effective, Fluid and Clear					
2	It was easy to make changes to the sketches					
3	Time was used productively					
4	Communication with the group was easy					
5	I found the tool Pleasant, Fun, Entertaining					
6	Doing the task was Efficient, Comfortable and easy					
7	I felt in total control of the tool					
8	I could use the tool as I use some others					
9	Managing objects used for the task was easy					
10	It was easy to begin the collaboration with my group					
11	I participated in the task in the way I wanted					
12	I participated in the task everytime I wanted to					
13	I was aware of what my partners were doing					
14	Information about others' actions was enough and clear					
15	I was aware of the state of the task and it's progress					

Table C.1: Survey

Bibliography

[1] Y. Shen, S. K. Ong, and A. Y. C. Nee, "Collaborative design in 3d space," in *VRCAI '08: Proceedings of The 7th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry.* New York, NY, USA: ACM, 2008, pp. 1–6.

[2] W. Broll, E. Meier, and T. Schardt, "The virtual round table - a collaborative augmented multi-user environment," in *CVE '00: Proceedings of the third international conference on Collaborative virtual environments.* New York, NY, USA: ACM, 2000, pp. 39–45.

[3] K. Sakong and T.-j. Nam, "Supporting telepresence by visual and physical cues in distributed 3d collaborative design environments," in *CHI '06: CHI '06 extended abstracts on Human factors in computing systems.* New York, NY, USA: ACM, 2006, pp. 1283–1288.

[4] S. D. Scott, M. Sheelagh, T. Carpendale, and K. M. Inkpen, "Territoriality in collaborative tabletop workspaces," in *CSCW '04: Proceedings of the 2004 ACM conference on Computer supported cooperative work.* New York, NY, USA: ACM, 2004, pp. 294–303.

[5] P. Tuddenham and P. Robinson, "Territorial coordination and workspace awareness in remote tabletop collaboration," in *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems.* New York, NY, USA: ACM, 2009, pp. 2139–2148.

[6] M. Altosaar, R. Vertegaal, C. Sohn, and D. Cheng, "Auraorb: using social awareness cues in the design of progressive notification appliances," in *OZCHI '06: Proceedings of the 18th Australia conference on Computer-Human Interaction.* New York, NY, USA: ACM, 2006, pp. 159–166.

[7] C. Cook and N. Churcher, "Modelling and measuring collaborative software engineering," in *ACSC '05: Proceedings of the Twenty-eighth Australasian conference on Computer Science*. Darlinghurst, Australia, Australia: Australian Computer Society, Inc., 2005, pp. 267–276.

[8] R. Vertegaal, "The GAZE groupware system: mediating joint attention in multiparty communication and collaboration," in *CHI '99: Proceedings of the SIGCHI conference on Human factors in computing systems*. New York, NY, USA: ACM, 1999, pp. 294–301.

[9] W. Appelt, "What groupware functionality do users really use? analysis of the usage of the bscw system," *Parallel, Distributed, and Network-Based Processing, Euromicro Conference on*, vol. 0, p. 337, 2001.

[10] T. Gross, "Towards flexible support for cooperation: Group awareness in shared workspaces," *Database and Expert Systems Applications, International Workshop on*, vol. 0, p. 406, 1997.

[11] K. Tee, S. Greenberg, and C. Gutwin, "Artifact awareness through screen sharing for distributed groups," *Int. J. Hum.-Comput. Stud.*, vol. 67, no. 9, pp. 677–702, 2009.

[12] C. Gutwin and S. Greenberg, "A descriptive framework of workspace awareness for real-time groupware," *Comput. Supported Coop. Work*, vol. 11, no. 3, pp. 411–446, 2002.

[13] ——, "Design for individuals, design for groups: tradeoffs between power and workspace awareness," in *CSCW '98: Proceedings of the 1998 ACM conference on Computer supported cooperative work*. New York, NY, USA: ACM, 1998, pp. 207–216.

[14] P. García, O. Montalà, C. Pairot, R. Rallo, and A. G. Skarmeta, "Move: component groupware foundations for collaborative virtual environments," in *CVE '02: Proceedings of the 4th international conference on Collaborative virtual environments*. New York, NY, USA: ACM, 2002, pp. 55–62.

[15] T. Granollers, "Modelo de proceso de la ingeniería de la usabilidad y la accesibilidad," GRIHO, http://griho2.udl.cat/mpiua/wiki/index.php/Inicio, 2005, accesado 7 Julio, 2010.

[16] C. Snyder, *Paper Prototyping: the fast and easy way to design and refine user interfaces*. The Morgan Kaufmann Series in Interactive Technologies, 2003.

[17] Microsoft, "Microsoft office visio," Microsoft, http://office.microsoft.com/en-us/visio/, 2010, accesado 26 Octubre, 2010.

[18] SmartDraw, "Smart draw official site," Smart Draw, http://www.smartdraw.com, 2010, accesado 26 Octubre, 2010.

[19] Axure, "Axure: Wireframes, prototypes, specifications," OpenSimulator, http://www.axure.com/, 2010, accesado 26 Octubre, 2010.

[20] iPlotz, "iplotz: wireframing, mockups and prototyping for websites and applications," iPlotz, http://iplotz.com/, 2010, accesado 26 Octubre, 2010.

[21] iRise, "irise: Visualize, innovate, deliver," iRise, http://www.irise.com/, 2010, accesado 26 Octubre, 2010.

[22] J. Prototyper, "Justinmind prototyper: Wireframe websites and mobile applications quickly and easily," JustInMind Prototyper, http://www.justinmind.com/, 2010, accesado 26 Octubre, 2010.

[23] "Mockflow," Produle Systems, http://www.mockflow.com, 2009, accesado 2 Junio, 2010.

[24] P. Project, "Pencil project: Sketching and prototyping with firefox," Pencil Project, http://pencil.evolus.vn/en-US/Home.aspx, 2010, accesado 26 Octubre, 2010.

[25] G. Guilizzoni, "Balsamiq Mockups," Balsamiq, http://www.balsamiq.com, 2010, accesado 2 Junio, 2010.

[26] P. Dourish, "The appropriation of interactive technologies: Some lessons from placeless documents," *Computer Supported Cooperative Work (CSCW)*, vol. 12, pp. 465–490, 2003, 10.1023/A:1026149119426. [Online]. Available: http://dx.doi.org/10.1023/A:1026149119426

[27] P. Dourish and V. Bellotti, "Awareness and coordination in shared workspaces," in *Proceedings of the 1992 ACM conference on Computer-supported cooperative work*, ser. CSCW '92. New York, NY, USA: ACM, 1992, pp. 107–114. [Online]. Available: http://doi.acm.org/10.1145/143457.143468

[28] M. R. Endsley, "Measurement of situation awareness in dynamic systems," *Human Factors*, pp. 65–84, 1995.

[29] D. A. Ovalle Carranza, J. A. Jimenez Builes, and C. A. Collazos Ordoñez, "Modelo de awareness en el CSCL ALLEGRO utilizando la arquitectura blackboard," *Ingeniería e Investigación [en línea]*, vol. 26, pp. 67–77, 2006.

[30] M. R. Endsley, "Toward a theory of situation awareness in dynamic systems," *Human Factors*, pp. 32–64, 1995.

[31] T. Gross, C. Stary, and A. Totter, "User-centered awareness in computer-supported cooperative work-systems: Structured embedding of findings from social sciences," *International Journal of Human-Computer Interaction*, pp. 323–360, 2005.

[32] S. Greenberg, "Observing collaboration: Group-centered design," pp. 111–118, 2008, very good summary of research on remote collaboration. Includes references to a framework on design team activities. [Online]. Available: http://grouplab.cpsc.ucalgary.ca/grouplab/uploads/Publications/Publications/2008-HCIRemixed.MITPress.pdf

[33] "The american heritage® dictionary of the english language," Houghton Mifflin Company, 2000.

[34] M. H. Tran, G. K. Raikundalia, and Y. Yang, "Methodologies and mechanism design in group awareness support for internet-based real-time distributed collaboration," in *Proceedings of the 5th Asia-Pacific web conference on Web technologies and applications*, ser. APWeb'03. Berlin, Heidelberg: Springer-Verlag, 2003, pp. 357–369. [Online]. Available: http://portal.acm.org/citation.cfm?id=1766091.1766137

[35] SecondLife, "Secondlife official site," SecondLife, http://www.secondlife.com/, 2010, accesado 26 Octubre, 2010.

[36] T. Inc., "Teleplace Whitepaper," http://www.teleplace.com/resources, 2008.

[37] J. Restrepo and H. Trefftz, "Telepresence support for synchronous distance," in *VRST '05: Proceedings of the ACM symposium on Virtual reality software and technology.* New York, NY, USA: ACM, 2005, pp. 63–67.

[38] "Kryonet framework. open source network communication api." Kryonet, http://code.google.com/p/kryonet, 2010, accesed August 2010.

[39] "Java persistence api," Oracle, http://www.oracle.com/technetwork/articles/javaee/jpa-137156.html, 2010, accesed August 2010.

[40] "Jmonkeyengine. open source 3d graphics engine," JME, http://code.google.com/p/jmonkeyengine, 2010, accesed August 2010.

[41] "Jmf: Java media framework," Oracle, http://www.oracle.com/technetwork/java/javase/tech/index-jsp-140239.html, 2010, accesed August 2010.

[42] D. A. Norman, "Affordance, conventions, and design," *interactions*, vol. 6, pp. 38–43, May 1999. [Online]. Available: http://doi.acm.org/10.1145/301153.301168

[43] M. I. Nurminen, A. Fjuk, and O. Sm?rdal, "Taking articulation work seriously - an activity theoretical approach," Tech. Rep., 1997.

[44] W. Shen, Q. Hao, and W. Li, "Computer supported collaborative design: Retrospective and perspective," *Comput. Ind.*, vol. 59, pp. 855–862, December 2008. [Online]. Available: http://portal.acm.org/citation.cfm?id=1454794.1455110

[45] R. O'Beirne, *Berkshire Encyclopedia of Human-Computer Interaction (2 Volume Set)*. Berkshire Publishing Group LLC, 2006.

[46] M. Villegas, H. Hernández, and W. Giraldo, "Implementación de un ambiente virtual colaborativo - especificación de un metamodelo de usabilidad," *Ingeniería e Investigación*, vol. 29, no. 1, 2010. [Online]. Available: http://revistas.unal.edu.co/index.php/ingeinv/article/view/15152

[47] OpenSimulator, "Opensimulator," OpenSimulator, http://opensimulator.org/wiki/Main_Page, 2010, accesado 26 Octubre, 2010.