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

Hamilton A Hernández
Sinfoci Research Group
University of Quindío
Armenia, Quindío, Colombia
hahernandez@uniquindio.edu.co

Helmuth Trefftz
Virtual Reality Lab
EAFIT University
Medellín, Antioquia, Colombia
htrefftz@eafit.edu.co

Abstract—**Many tools help GUI designers to produce prototypes of their future applications, but when this job requires the collaboration with geographically dispersed partners some problems arise like lack awareness of the others' work, concurrent manipulation of virtual objects and non-fluent communication of actions and intentions. WeSketch allows a small geographically distributed group to build GUI prototypes in a highly participative manner, keeping everyone aware of what is going on. A 3D virtual round table, virtual sheets simulating sketches, avatars representing users, an audio communication system and a shared view of sketches are the pieces that allow designers to discuss and co-design GUIs in real time offering a comfortable and pleasant experience. This paper describes the tool and the results of a set of experiments conducted in order to validate 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 than they do with current tools".**

Keywords- awareness, groupware, prototyping, user centered design, collaborative design

I. INTRODUCTION

One of the most important tasks in User Centered Design (UCD) is Prototyping [1]. For complex software this task is usually performed by groups of designers who work collaboratively to design and choose the best Graphical User Interfaces (GUIs) for a set of requirements. Traditionally, these designers get together in a physical room and work around a table or a shared board. This traditional way of prototyping doesn't allow designers to reuse parts of their designs (like with the copy-paste in computers), to try different appearance options for the preliminary designs, to trace the produced artifacts along the other computerized stages of the UCD process nor to participate remotely; so many applications have emerged offering tools that allow designers to better manage the sketches related to specific requirements, increase their productivity and consistency by using predefined and configurable widgets, work remotely and easily store and retrieve 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 [2]. 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 (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 [3] 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 judgment 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.

II. CONCEIVING WESKETCH

In order to provide a high quality application, a User Centered Design approach was followed, starting by a Requirements Analysis Stage and continued by an iterative and participative Design and Test stage.

A. *Requirements Analysis*

Nine 2D prototyping tools and three general purpose 3D collaboration environments were studied to evaluate their ease of use, diversity of tools for the prototyping task and their support for collaboration. Teleplace [4] and Mockflow [5] were chosen as the most complete 3D and 2D tools, respectively. These tools were used to identify the functionalities that best supported the prototyping task and the more frustrating software constraints, based on an expert revision and the analysis of a set of user studies.

1) User Studies: Three user studies with different settings were conducted and videotaped, and a survey using the Likert Scale and 2 open questions was applied. For each study, several groups of designers were requested to produce a set of GUIs for 3 detailed requirements given in text format. The first study was done with participants of every group sitting around a physical table using paper, pencils and similar materials, which is the traditional way of prototyping; the second and third study were conducted with users geographically dispersed connected by a computers network and using Teleplace and MockFlow, respectively.

2) Identified Needs: The next one is a summary of the features identified as mandatory for an application that solves the problems identified with the user studies.

Desirable features are ordered by the most important first:

- provide several consistent configurable GUI widgets
- possibility to add or modify widgets in the sketch in any moment without waiting for others to cede the turn nor disrupting their work
- better awareness mechanisms so that designers can see what the others are doing
- a fluent communication based on audio
- good accuracy when selecting and dragging drawings around the canvas
- a mechanism to store the sketches produced during a design session and assign them to specific functional requirements

B. *Design*

Based on the identified needs, some functionalities offered by other competitors software, theories related to Workspace, Group and Artifact Awareness [2], [12]; theories about Territorial Coordination [6] and studies about tools for Awareness in Groupware[7]; we designed an application that offers good support to the real needs of designers while they prototype GUIs.

WeSketch is a Client-Server application built in Java. The server is in charge of the management and persistence of the shared information during sketching sessions and broadcasting awareness, chat and sketches information among clients. Every client presents the awareness information, the chat messages and the shared sketches using 2D (Java Swing) and 3D objects.

WeSketch provides to GUI designers a 3D virtual room where they can get together to produce a set of sketches related to detailed descriptions of software requirements. Using a 2D Sketching frame designers can create GUI sketches collaboratively, editing them concurrently and being aware of what the others are doing

Additionally, clients can start an audio-conference using the Java Media Framework (JMF) [8] by using the IP addresses directly, without interacting with the server.

Figure 1 shows the APIs and Frameworks used to build WeSketch.

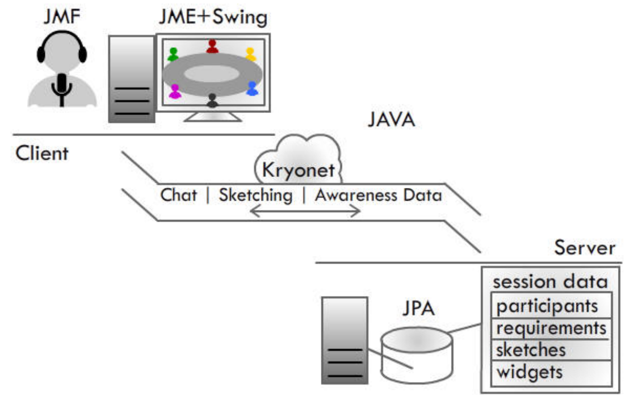

Figure 1. Frameworks and APIs. Kryonet: Networking framework [9]. JPA: Java Persistence API [10]. JME: 3D Graphics Engine [11].

Figure 2 illustrates the architecture of WeSketch by depicting the frameworks and APIs used and the most representative classes developed.

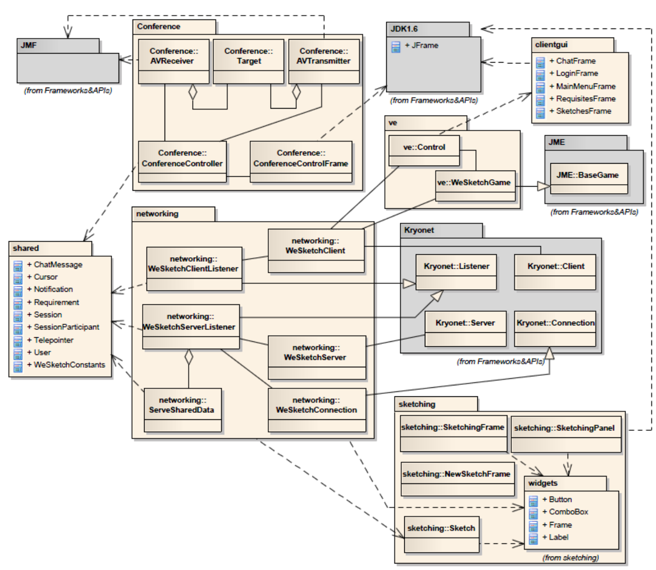

Figure 2. Overview of WeSketch's Architecture with the most representative classes. The package widgets contains 32 common widgets used during GUI design.

1) The 3D Environment: WeSketch has a 3D virtual environment mapping the metaphor of the traditional physical prototyping environment. We have included static objects such as a virtual room, a round table and seats for 6 participants, and dynamic objects such as colored avatars representing every participant, and paper sheets over the table. Designers can rotate their first person shooter view and see the avatars sitting around the table. The name of a participant appears as a billboard over the head of its avatar. Designers can also use a 3D laser to point at the objects of their interest and thus communicate it to the others; this event is followed by a rotation of the avatar's head towards the object of interest. The avatars will change their posture when they begin or stop working on a sketch (see Figure 3).

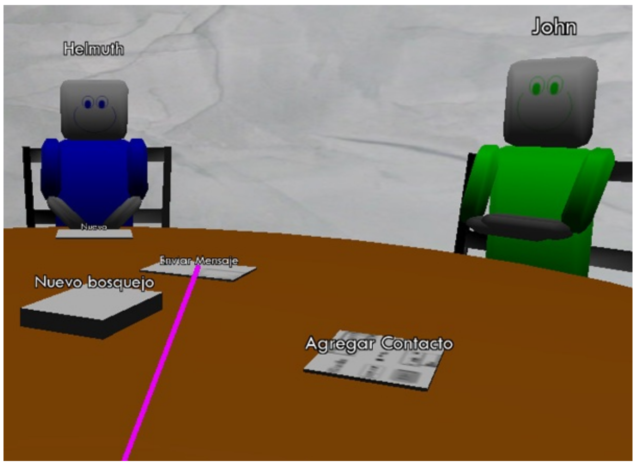

Figure 3. Part of the virtual room: the round table, 3D avatars in different postures/states, virtual sketches and a user's telepointer ray.

The existing sketches will appear as virtual sheets distributed over the table with the name of the sketch they represent displayed as a billboard and a preview of their content as the texture of the sheet. The sketches being used by a designer will appear on the private space of the designer, the others will appear around the center of the table, where a virtual pile of sheets will allow the designers to create new sketches.

2) *The 2D Interfaces:* 2D frames were included to easily capture input from users when they are registering in the system, signing in, providing information about a new sketch, setting the audio-conference, chatting, reading the requirements descriptions and editing a sketch. This frames are internal frames embedded in the 3D environment and will appear as a response to some events such as clicking a virtual sketch or clicking the chat button in the menu bar. Some of them are blocking dialogs and some others can be moved aside to have them available while paying attention to the virtual room or working with another 2D frame.

The real time sketching frame is the biggest 2D frame and is displayed when a designer clicks on a Sketch to edit it. This frame displays the widgets added by designers to the sketch and allows their direct manipulation by clicking and drag-dropping them. To add new widgets to the sketch designers use a toolbar in the frame with predefined and configurable widgets. This frame also offers awareness information such as 2D avatars with colors and labels representing participants working on the same sketch; cursors in different colors and types to communicate actions fired by the others; and colored controls around a widget to illustrate that someone is editing it. See Figure 4.

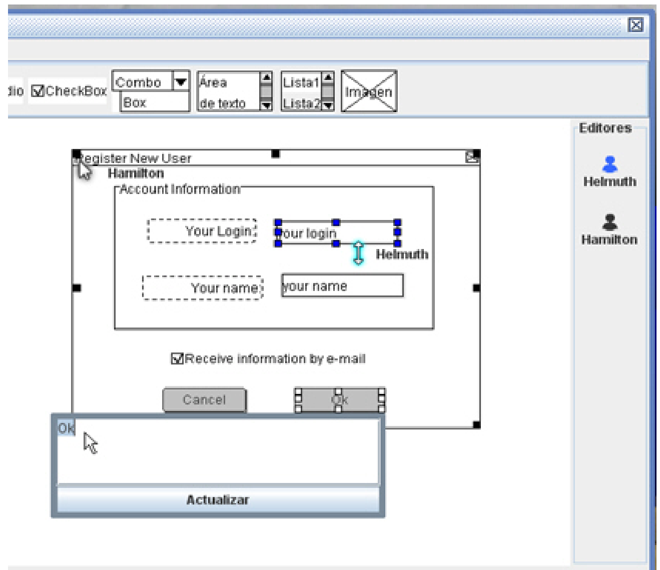

Figure 4. Zoom of the real rime sketching frame. 3 users are working simultaneously in the same sketch. Every participant is working on a different widget (unnamed cursor is the current users'). Selected widgets have colored controls according to the color of the designers using them.

III. THE 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 than they do with current tools". In order to validate it, we designed the next experiments.

We wrote detailed descriptions of 3 hypothetical functional requirements to be prototyped, giving clear details about the information that was required for every GUI. Every requirement was described so that they all had equal complexity and so that its resulting sketch required the same amount and complexity of widgets. One of them was:

System: A Health Insurance Information System
FR01: Submit a Question, Complaint or Proclaim.
Description: The user will fill a form for Questions, Complaints or Proclaims providing the next information:
-Document Id
-Full Name
-Address
-Phone
-E-mail
-City: (Armenia, Pereira, Manizales, Cali, Medellín)
-Request type: (Question, Complaint, Demand, Proclaim)
-Description
Additionally, users should be able to cancel the task.

For the experiments we recruited 39 fifth-year students of a Systems and Computers Engineering program with a background of at least one semester in Usability and GUI design, they all formed groups of 3 people. Every group was dispersed putting the 3 students in a different classroom and they were asked to sketch collaboratively the GUIs using a different tool for every task. Some groups used MockFlow [4] for the first task, Teleplace [3] for the second task and WeSketch for the third task. To avoid 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 extracted. These measures were achieved by counting the messages transmitted, changes made to the sketches, idle time and time invested on design, and in general, Time Completion Task for the whole group.

Figure 5 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.

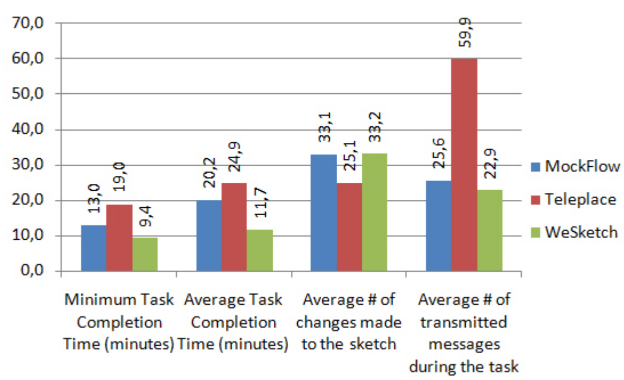

Figure 5. Measures related to Task Completion Time, Productivity, Iteration and Communication Fluidity

Figure 6 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 need to dedicate only to give written or spoken instructions and opinions while waiting to participate, and this makes the task to lasts less.

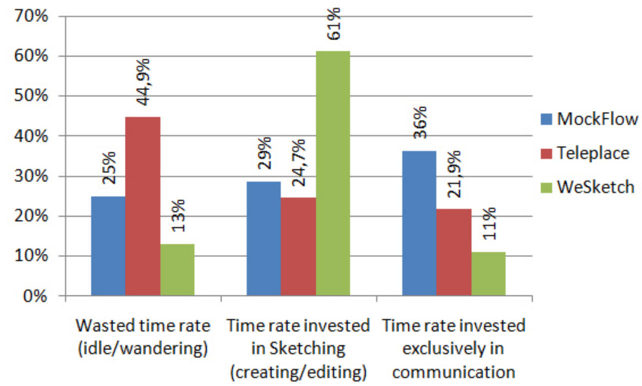

Figure 6. Measures related to Productivity

Once every task was completed, a survey with several Likert Scale questions and 2 open questions (Why? and What would I do better?) was also applied to the users. The Likdert Scale questions were chosen carefully to get feedback from every designer about satisfaction and comfort. The questions and user's scores are shown in Figure 7.

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

- 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"

Question	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%	48%	49%	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 7. Questions of the survey applied to users who used Teleplace, MockFlow and WeSketch and their scores in the Likert Scale.

- "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 very simple and easy to learn"

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.

Additional to the measures and surveys, we wanted to probe that the data obtained was statistically significant, so we made a Student's T Test [13] comparing WeSketch and MockFlow and then WeSketch and Teleplace. The result in both cases was p factor < 0.001, which means that the results obtained in our experiments had a significant statistical meaning and were not obtained by chance.

IV. 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, like the over the shoulder view [2].

Most of these mentioned improvements are being developed at this moment and will 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.

Although we identified some issues while inspecting the tool and running the experiments, we consider important to conduct formal usability evaluations on WeSketch so that we have a better understanding of what should be improved.

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 telepointer to the sketch, but we think that there may be a better way to illustrate this, like locating avatars behind the first avatar who took the sketch, for example, so we let this open to discussion.

We think that WeSketch's architecture could also be improved to fit other tasks where groups need to work concurrently and aware of their environment, their group and their resources.

V. CONCLUSIONS

A set of experiments were carried out and 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 than they do with current tools" 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 who's 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.

A more complete description of the WeSketch project can be found at http://www.sinfoci.net/wesketch

ACKNOWLEGMENT

WeSketch was born as an MSc thesis project in the Virtual Reality Lab for the Master in Engineering program of EAFIT University in partnership with University of Quindio. The authors want to thank both universities and the lab for their support and sponsorship.

We would also like to thank the people from SINFOCI research group and the Systems and Computers Engineering program students, especially David Cadena and Jesse López, from University of Quindío, for their contributions and active participation in the evaluation of Wesketch and the conducted experiments.

REFERENCES

[1] Granollers, T. and J. Lores. "Modelo de Proceso de la Ingeniería de la usabilidad y de la accesibilidad". 2005; Available from: http://griho.udl.es/mpiua/.

[2] Gutwin, C. and Greenberg, S. 2002. "A Descriptive Framework of Workspace Awareness for Real-Time Groupware". Comput. Supported Coop. Work 11, 3 (Nov. 2002), 411-446

[3] 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.

[4] T. Inc., Teleplace Whitepaper, http://www.teleplace.com/resources, 2008, visited on September 10, 2010

[5] Mockflow, Produle Systems, http://www.mockflow.com, 2009, visited on September 10, 2010

[6] 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.

[7] C. G. Saul, "Staying aware in groupware workspaces" 1996.

[8] JMF: Java Media Framework. http://www.oracle.com/technetwork/java/javase/tech/index-jsp-140239.html, August 2010.

[9] Kryonet Framework. Open Source Network Communication API. http://code.google.com/p/kryonet, August 2010.

[10] Java Persistence API. http://www.oracle.com/technetwork/articles/javaee/jpa-137156.html, August 2010.

[11] JMonkeyEngine. Open Source 3D Graphics Engine. http://code.google.com/p/jmonkeyengine, August 2010.

[12] 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.

[13] Zimmerman, Donald W. (1997). "A Note on Interpretation of the Paired-Samples t Test". Journal of Educational and Behavioral Statistics 22 (3): 349–360.