# Large Multi-Touch Vertical Displays in Multi-User Competitive Tasks

BY

DAVIDE TANTILLO
B.S, Politecnico di Milano, Milan, Italy, 2014

THESIS

Submitted as partial fulfillment of the requirements
for the degree of Master of Science in Computer Science
in the Graduate College of the
University of Illinois at Chicago, 2016

Chicago, Illinois

Defense Committee:

G. Elisabeta Marai, Chair and Advisor
Tom Moher
Andrew E. Johnson
Franca Garzotto, Politecnico di Milano

# ACKNOWLEDGMENTS

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# LIST OF ABBREVIATIONS

| | |
|---|---|
| D3 | Data-Driven Documents |
| DOM | Document Object Model |
| FTIR | Frustrated Total Internal Reflection |
| GUI | Graphic User Interface |
| HD | High Definition |
| IR | Infrared |
| EVL | Electronic Visualization Lab |
| LCS | Local Coordinate System |
| LCD | Liquid Crystal Display |
| LMVD | Large Multi-touch Vertical Display |
| LVD | Large Vertical Display |
| MVC | Model-View-Controller |
| OS | Operating System |
| PI | Principal Investigator |
| QHD | Quad High Definition |
| SAGE2 | Scalable Amplified Group Environment |
| SVG | Scalable Vector Graphics |
| UHD | Ultra High Definition |
| UIC | University of Illinois at Chicago |

# SUMMARY

Large vertical displays (LVDs) are starting to appear in different realities. There are many exhibition and fairs that are starting to employ LVDs to show information and commercials, dynamically and interactively thanks also to the LVDs' touch capabilities. LVDs are becoming more relevant also in the workplaces since they allow to host virtually remote participants and show much information at the same time. Schools and Universities began to test LVD capabilities to support professors in a lecture by allowing them to show documents, videos and take notes, all at the same time.

LVDs potentiality are in their very wide resolution which is capable of hosting much information at the same time and in the fact that they are connected to a computer, which means instant internet access, where a tremendous quantity of information is immediately usable. However, this huge resolution is not easy to exploit. Too much information might be confusing and the user might also not benefit from a simple piece of information presented in an inefficient way. Fortunately there are some research that studied how to present information on LVDs.

Another important LVDs topic is user interaction with these kind of devices. There are two main approaches to do this: through mouse and keyboard or using touch capabilities. Both approaches have pros and cons and their preference should depend on the used application. Actually, in the beginning touch detection systems suffered from being imprecise and unreliable. However with the introduction of the possibility to interact with LVDs through a suitable and more reliable multitouch input system, a new frontier of interaction between human and machine — or human-computer interaction — was possible and consequently new possibilities and challenges were revealed. During recent years, many studies were conducted on the combination of LVDs and multitouch systems related to their usage in science, visualization, schools and in the workplace, and to provide directives on how they should be employed.

However, there are still many questions and open problems that are preventing LVDs to enter in many realities where they may improve many tasks. One of these question is that there is no precise methodology to create an interface of such a resolution. Another question is that it is not clear yet which is the most efficient way to interact with LVDs. Eventually there is still a lot of research to do in the direction of how humans interact with these large devices either alone, in pairs or in a group.

There are several interesting studies about these open questions, but they are only starting points, as also stated within their conclusions.

This thesis gives a contribution to the human-computer interaction literature by studying the problem of a collaborative-competitive task using an LVD with multitouch capabilities. We conduct a user study comparing the behavior and the results of groups of users performing the same task using a traditional approach and then using an application for an LVD. We want to understand how people interact with the display and with one another as a group when each user has a personal goal in addition to a shared one. Furthermore, we want also understand if technology can help reduce the frustration of a collaborative-competitive task compared to the traditional approach.

At the same time we provide suggestions on how a scalable user interface for simple application can be implemented using a library to implement scalable vector graphics, or SVG.

# CHAPTER 1

## INTRODUCTION

People are generally enthusiastic of innovative technologies. One of the most important witnesses of this enthusiasm is the reaction of the audience in the iPhone 2007 Steve Jobs presentation when he showed the pinch gesture to zoom an image[1]. People literally said 'wow' and after a while clapped, completely fascinated by what they were seeing. LVDs or compositions of them are one of these technologies that people appreciated in the past and continue to appreciate in the current days. Just imagine how in living rooms the televisions are becoming every year larger. Also in office desks are present larger monitors or more of them side by side used in order to increase the number of open program windows.

The main characteristics of LVDs are a ratio where the horizontal dimension is three or four times the vertical one, and the width that is at least three meters. These dimensions make very inefficient their usage in everyday tasks, or at least using it through applications developed for a normal desktop pc. Nevertheless, many fields start to have a strong interest in this kind of device. Companies had an interest on employing them for meetings or presentation to customers [1]. Studies ([2], [3], [4], [5]) understood how their huge resolution was potentially able to show an enormous quantity of information in a more efficient way with respect to all the methods used in the past. A map is an interesting example of the higher efficiency of LVDs; a high-resolution map fits better in a display with very high resolution. Many scientists can simultaneously analyze the map, and easily discuss about points of the map having these points in front of themselves.

The efficiency of the scientists' actions may be further increased by the introduction of a system that allows the scientists to directly manipulate with their hands and fingers what they are observing. Here comes the need of a LVD with multi-touch capabilities. Such a feature, in addition to being fascinating and attracting, increases enormously the degrees of freedom of the interaction between the human and the LVD. Another interesting example useful to appreciate the potential usefulness of this

---

[1]https://youtu.be/vN4U5FqrOdQ?t=33m45s

Figure 1: Two scientists analyzing a map on a LVD.

device is a group of scientists during the analysis of a molecule model. The bigger size of a molecule model on a LVD and the possibility to rotate and zoom the model aid scientists on their analysis and discussions.

LVDs are a technology relatively new that bring numerous challenges together with their potentiality. Adding multi-touch interaction increases the number of the previous challenges that computer scientists and developers have to tackle and to find a solution for. Some of these challenges are related to:

- *Input Detection*: how users should interact with the display, which is the most efficient input device to interact with the display among single mouse, multiple mice, single-touch, multi-touch, pointer, if input devices influence the user collaboration;

- *Human Interaction*: how the usage of a normal operating system is comfortable on an LVD, how users dispose windows, how users exploit the display space and if there are less used area, how the position of the users with respect to the LVDs influences the previous factors, how users collaborate in a shared context;

- *Data Visualization*: how to show data and information, how to interact, navigate, consume and produce data;

- *Application Context*: in which context applications have to live, so if an application has to be a normal operating system program or if has to live in a dedicated context;

- *Application Development*: how to organize an application interface for this large resolution, which are the guidelines to follow for LVD application development, if the normal desktop application development has the same characteristics of LVD application development, if there are specific library dedicated to application for LVD.

The past and current research is directed mainly towards the first three points of the previous list. Data visualization on LVDs is probably the most researched field of the previous points. Different techniques dedicated to different kind of data can be employed to allow users to consume more efficiently the information. There is much research also in the human-interaction field with these devices that generated a interesting conclusions. However, these conclusions are somewhat sparse, sometimes divergent and consequently it is difficult to exploit efficiently the information. This is a clear symptom that LVDs are still a very young study subject and much effort has still to be done in order to create a consistent literature. The same phenomenon is observable for input detection for LVDs, even though there are less studies on this direction.

Instead, the research seems to be absent for the application context and the application development for LVDs. There are no directives or guidelines suggesting how to develop applications for very large and high resolution displays. This consequently leads developer to create a dedicated solution for each problem. Moreover, there are not any dedicated software environment to them. There are some attempts on making browsers the main container for LVD applications [6]. In fact, internet page interfaces can be adapted to whatever resolution without a big effort. However there are no common ways or guidelines to organize an interface or create functions for LVD applications, and every small piece has to be created from scratch by the developer.

In this thesis we will focus on the human interaction with LVDs.

## 1.1    Thesis Objective

This thesis has two objectives. The main is oriented towards the human interaction with large multi-touch vertical displays (LMVDs) in a multi-user environment, and so to the human-computer interaction discipline. We will analyze groups of people working together during the scheduling of a scientific conference, but with the underlying goal of reserving for their speeches a slot in dates and hours where they are available, considering that other people may be potentially interested in the same time slots. We will provide information about the people interaction with a LMVD, how they interact together, if they evenly share the resources, if the frustration that a competitive task generates is reduced using the technology with respect to a traditional approach using real post-it notes and a simulated whiteboard.

The second objective is the creation of a technique to develop a simple interface for small applications in the high resolution context of LVDs. Attention will be given also to multi-touch input management and a simple technique to manage them will be shown. We will make use of a library to implement scalable vector graphics, or SVG, components. The usage of SVG will avoid scaling resolution problems and will make the interface independent from the display resolution.

## 1.2    Thesis Structure

Chapter 2 introduces the reader to the world of LVDs and multitouch systems. It then gives background knowledge on the fusion of these two systems discussing about LMVDs. In the end, it analyzes exhaustively the conference scheduling problem.

Chapter 3 shows interesting studies on LVDs and LMVDs. At the beginning, it gives an overview of the state of the art of LVDs and LMVDs, describing the strengths and the weaknesses of these two technologies. Moreover, research on the human behavior using LVDs and LMVDs will be analyzed, showing what experts of the human-computer interaction field are interested in and how these user studies are conducted.

Chapter 4 illustrates how we developed the application we used for the user study we conducted – the Conference Scheduler – and the application components. The first part explains a simple, yet effective technique, used to realize scalable LMVD interfaces with the potentiality of a Scalable Vector Graphic (SVG) library. The second part shows how we exploited this technique to create the Conference Scheduler tool and illustrates some technical details.

Chapter 5 contains all the information related to the user study we conducted. At the beginning, we describe how we conducted the user study, such as the procedures, the employed apparatus, and the measures we decided to capture and analyze. The chapter then shows the results and the analysis we did. At the end, a comparison between our and the other's findings is shown.

Chapter 6 provides a summary of the thesis findings, making statements about the more consistent results, but also analyzing the weaker ones. At the same time, this chapter gives some suggestions on future work that might be done to reinforce the literature of LMVDs in the human-computer interaction field.

# CHAPTER 2

# BACKGROUND

This thesis tackles a problem that combines LVDs and touch detection systems immersed in a multi-user environment. LVDs per se are a vast topic and considering also touch detections systems increases the topic which is possible to discuss about. This chapter wants to take into account those aspects that are interesting for the purpose of this thesis. First, we analyze the main characteristics and problems of LVDs and the aspects that have characterized them over the previous years. A section is dedicated to the touch detection systems making a taxonomy of the available technologies. The section ends with a paragraph dedicated to the combination of a LVD and a touch detection systems. We discuss then about LVDs employment in a multi-user environment. Last, we talk about the conference scheduling problem that is the problem took into account to make our user study on LVDs in competitive environment.

## 2.1 Large Displays

There is no strict characterization of what a large display is. Generally the context define if a display is a large display. For example, in an office a large display can be considered a 27 inches display. Larger display are expensive. For this reason, it is preferred to increase the available screen space creating a composition of smaller displays, as shown in Figure 2. Although not completely correlated, also resolution changed and increased. The Full HD standard (1920 x 1080 pixels) is the most used, even though the new Quad HD (2560 x 1440 pixels) and 4K (3840 x 2160 pixels) resolutions start to be used for desktop displays and less often in laptop displays, even though the diagonal of laptop displays is often limited to 15.6 or 17 inches. Even for desktop, it seems that there will not be any more surprising increases, also because in many scenarios, usually workplaces, higher display dimension is reached combining two or three displays, as explained above.

### 2.1.1 The Large Display Paradox

Increasing the monitor size or number is not the most efficient thing to do in every context. This is well explained through 'The Large Display Paradox'[1]. This paradox states that small displays have

---

[1] http://blog.codinghorror.com/the-large-display-paradox

Figure 2: A composition of 3x3 displays in a workplace.

advantages such as the fact that people do not waste time on resizing, moving, or changing the vertical ordering of windows, but have only to deal with a single maximized application at a time. It is also easier to change the foreground window since the bottom bar is easier to reach. Obviously, it is harder to use more than one application at a time with small screen space. A large display makes the usage of multiple applications easier, but the usage of a larger display has some cons. In fact, as long as the size of a display remains limited operating system automatic windows disposition makes this task manageable, but when the display number increases or the size enhances drastically, maximizing, or smartly positioning windows become a real problem. The extreme case is reached especially by very large displays, composed by a unique or a cluster of displays, where windows maximization is an important problem. If no ad-hoc solutions are employed, they become very inefficient to be used for the most common tasks. For example, most websites do not support large resolutions, and zooming is often not very useful since interface's items tend to assume quite a random position on the screen. The lack of support for these kind of displays is justified by the fact that they are not so common and generally are used for specific purposes. However, since their usage is becoming more frequent, at a certain point at least the most important websites have to start to think about these new users. Moreover, these users may desire to access an email client or a cloud storage service. The only way to

benefit of these services on LVDs in a reasonable way is using a reduced browser window, positioning it in front of the user, zooming the view if necessary, and starting to perform the task. Also in this condition, the website usage is not comfortable. When LVDs will be employed more and by several other sectors a consideration to these resolution problems should be done.

### 2.1.2  Resolution Problems

A problem encountered both in normal displays and very large ones is the fact that increasing the resolution over the Full HD (1920x1080) makes the interface's items and texts too small. There are 15.6 inches laptop and also smaller ones that support Quad HD and 4K resolutions. Using these extreme resolutions on limited size display makes inevitably difficult the usage of an interface and the reading of texts. With the introduction of these resolutions, operating system manufacturers thought to insert the possibility to scale up everything so that texts and controls can be again intelligible. Nevertheless, on LVDs this solution is not enough. Simply scaling up an interface may work when screen proportion are respected, but when the standard proportion is not present, a whole reorganization of the interface following a different paradigm of human-interaction is necessary. Regarding the problem of windows disposition on large displays or on more of them, there are applications that tried to solve the problem in smart ways. One of this application is WinSplit Revolution that simply allows to create virtual grids over the monitor and assigns a number to each cell. Pressing the key combination ctrl + alt + (numpad) the foreground window is resized and placed in the corresponding cell of the virtual grid created before.

WinSplit Revolution in its earlier versions did not support more displays, but then integrated the idea. Figure 3 shows how a combination of 4x3 displays can be divided in three different virtual displays composed of different sets of displays. The virtual displays help to position the windows over the large display. This application has been discontinued, but alternatives perform the same task in an efficient way, such as MaxTo or UltraMon. This simple idea was introduced also in Microsoft Windows 7 in a slightly different way with the name of Snap[1] and then improved in the next versions. Pressing the window key + an arrow key, the foreground window is resized and positioned in a certain position depending its current state. For example, if a window is maximized, pressing window key + right

---

[1]http://windows.microsoft.com/en-us/windows7/arrange-windows-side-by-side-on-the-desktop-using-snap
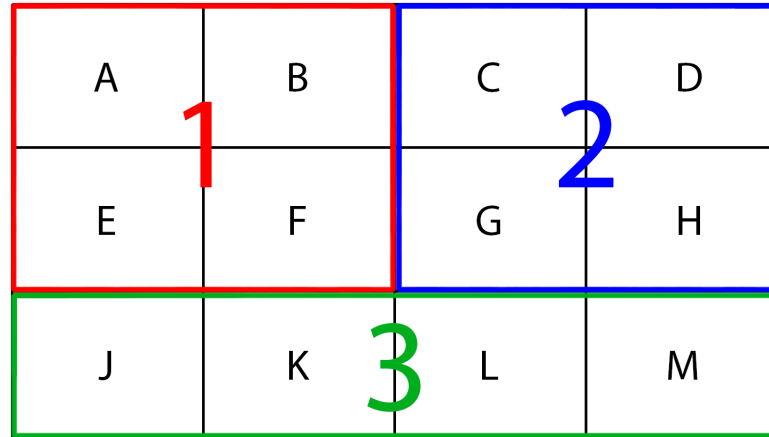
Figure 3: A schema of a combination of 4x3 displays divided in three different virtual displays. Virtual display 1 is a 2x2 display composed of display A, B, E, F. Virtual display 2 is a 2x2 display composed of display C, D, G, H. Virtual display 3 is a 1x4 display composed of display J, K, L, M.

arrow key will position and resize the application in the half-right side of the monitor. Now, pressing the window key + up arrow key will position and resize the application in the top right corner. It is possible to activate Snap also dragging a window toward a border of the display and touching it. With the exception of the bottom border, dragging a window on one of the seven remaining directions will position and resize the application in the corresponding part of the monitor, as shown in Figure 4.
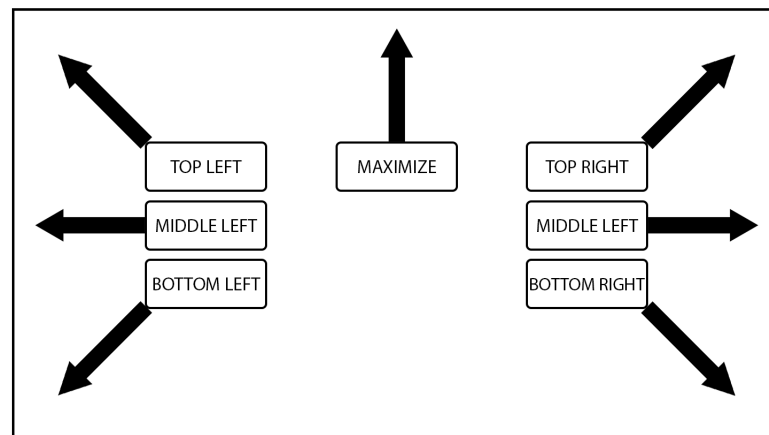


Figure 4: The figure shows in the rectangles the corresponding behavior of a window when is dragged on the border indicated by the respective arrow using a version of Microsoft Windows later than Windows 7.

For example, dragging it toward the left border of the display will position and resize the application in the half-left part of the display. While doing it toward the upper border, this will maximize the window. Even though this solution has limited way to position windows over a display with respect to application such as WinSplit Revolution, Snap was well accepted by the community as it is inserted in most of the articles[1] about the most useful Microsoft Windows keyboard shortcuts.

On Mac OS things are different. Mac users follow a different paradigm for windows disposition. With the introduction of Spaces[2], later called Mission Control, using multiple virtual desktop and the possibility to switch among them with the simplicity of a gesture on the trackpad, Mac users tend to maximize every windows dedicating roughly up to two applications per desktop. Moreover, they have the ability to place side by side desktops making possible also the visualization of more windows per desktop in an efficient way, even though the automatism of these gestures are reached after some usage. A similar feature is present in the OS Linux known as Workspaces[3]. As long as the problem of positioning windows remain limited to displays with a normal size, these features help and satisfy most of the users. Nevertheless, when LVDs come into play, these positioning and resizing ways are still not enough. The main problem is that with very high resolution, the automatic disposition of the interface items start to become meaningless: Google Gmail client is an example of this bad behavior.

When the *Compose button* is pressed, a *New message window* is created in the bottom right corner of the screen. This behavior may be useful for normal resolution, but it is completely meaningless for large resolutions. Figure 5 shows how much space there is between the *Compose button* and the *New message window*. It may also occur that the user does not realize that the *New Message window* was created since it is difficult to see the button feedback from the user position. Ones might think that providing an ad-hoc interface for this limited slice of users may result in an useless effort and probably they are right. Anyway, whoever want to develop an application for these kind of devices has to take into account this relevant problem. If someone is selling certain products, they should be worried about the possibility that their customers may desire ad-hoc features.

---

[1]http://www.howtogeek.com/198122/32-new-keyboard-shortcuts-in-the-windows-10-technical-preview/

[2]https://support.apple.com/kb/PH18757

[3]https://help.ubuntu.com/stable/ubuntu-help/shell-workspaces.html

Figure 5: Gmail interface does not scale up well for a LVD. When the user presses the compose top left button, he cannot notice the new window on the bottom right because it out of his field of view, as the red dashed line suggests.

### 2.1.3 The Missing Large Display Paradigm

There is no paradigm to follow for the creation of an LVD interface. This lack is well explained by Moreland [10] in his third lesson: "*Large-format display can impede interaction*". This lesson suggests that as smartphones and tablets introduced a new human-computer interaction paradigm, also LVDs should introduce their own. Some challenges with the interaction with this kind of displays are analyzed by Andrews et al. [11]. Until some interaction paradigms are not deeply studied, LVDs will not be used as frequently as people might expect. An indicator of this low usage is that a lot of research was done during the 90's and early 00's, starting from the necessity to visualize data on large screens as in the ASCI program [12] of the United States Department of Energy and the guidance offered by the DVC Workshop [13]. In the next years, the research conducted in LVD's field remained mostly stationary. It is a pity considering also the fact that the cost of installing LVDs is dramatically reduced. In fact, an LVD can be obtained by composition of smaller ones. In the past, several projectors were placed side by side in order to obtain a higher resolution screen. However, the problems related to alignment and color balancing make projectors usage undesirable for scientific purposes. Moreover, the cost and maintenance of projectors are usually high. However, single projectors are sometimes used in composition with touch technology [5]. A more interesting solution is represented by the usage of multiple LCD displays. Nowadays they are relatively cheap and easy to mount. Moreover, several companies that offer already complete and customizable solutions stimulate to go toward their employment. However, LVDs have an important disadvantage to consider. Their bezels [14] create

a perceivable discontinuity that users find troublesome at a first glance. Anyway, there are studies concluding that no differences are perceived by people using multiple displays [15] or that the black borders on adjacent displays are useful for windows disposition [16]. This findings mean that even though research should find a way to reduce or remove the presence of bezels on combination of displays, bezels do not represent an obstacle to the diffusion of multiple displays to compose a larger one.

Good news is that with the introduction of supercomputers and the possibility to compute huge quantities of data in less time, coupled with the need of scientists or interested people to visualize data in larger resolutions, research on LVDs received a new boost.

## 2.2    Touch Detection Systems

LVDs are one of the subjects of this thesis. Multi-touch detection systems are another important subject and can be considered like another innovative technology that people are enthusiastic of. Although we will discover multi-touch has a longer history than we can imagine, an effective example similar to the one of the pinch gesture to zoom, showed by Steve Jobs, is the live demonstration given by Jeff Han in a TED Conference of 2006[1]. In less than ten minutes, he showed how many things are possible to do adding the capability to use more than one detection point in touch systems. He zoomed, moved and rotated an interface, drew lines and figures, typed with a virtual keyboard and other amazing actions, zoomed photos (the gesture produced a similar effect on the audience to the one obtained by Steve Jobs with the same action, but a year earlier, mentioned in the introduction), each one executed with an incredible simplicity. From approximately that date, many electronic devices employed multi-touch detection systems, especially smartphones. In fact, posing some attention on the dates, multi-touch detection systems are those that introduced us in the era of smartphones marked probably by the first iPhone, the same one cited at the beginning of this thesis introduction, equipped with a fully functional multi-touch detection system. Since they are highly sold, smartphones and their kind of multi-touch systems received a proportional attention from the research field. For example, Apple recently introduced in its last device the 3d touch[2] that is able to distinguish between a soft

---

[1] https://youtu.be/QKh1Rv0PlOQ

[2] http://www.apple.com/iphone-6s/3d-touch

and a hard touch and consequently increasing the number of input ways provided to the user. As we can imagine, less attention was dedicated instead to those multi-touch systems used on LVDs, that use generally infrared overlay, in contraposition to the capacitive technology used on smartphones.

### 2.2.1 Touch Detection Systems Taxonomy

There are several touch detection system technologies. We describe here the most famous and explain at the end why the infrared (IR) technology is the most employed for LVDs.

**Resistive touch** technology (Figure 6) is probably the most employed technology for touch displays. This system is part of the display itself and is composed of the display glass and a polyester film. These two parts are internally covered with an electrode film separated by a gap. The users' touch is detected by the contact of the two internal films that implies an electrical flow. Resistive touch main advantages are that it is possible to perform a touch with whatever object, that is the cheapest technology and that has a low power consumption. Moreover it is resistant to water and dust. However, resistive touch main disadvantages are the low precision and that the four layers before the actual display decrease the image quality. Furthermore, the external film can be damaged by sharp objects. It does not allow multiple touches at the same time.
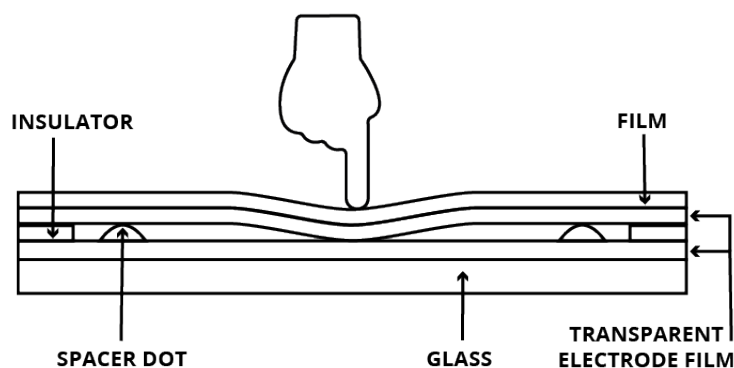
Figure 6: Resistive touch technology.

**Surface capacitive** technology (Figure 7) is another widely employed solution. This technology exploits the static electrical capacity of a human or a stylus with an electrical charge. The display is now composed of a glass covered by an electrode film and a protective cover. When a finger touches the display, some electrical charge is transferred from the screen to the human body. This capacitance

decreasing is detected by four electrodes placed on the display corners that are able to determine the touch point. The remotion of the plastic film with respect to the previous technology allows to obtain a better image quality and a more durable and resistant screen. However, a bare hand or a capacitive stylus is needed to detect the touch. For example, a hand covered by a glove does not constitute a perceivable touch for this technology. Last, it is susceptible to Radio Frequency Interference (RFI). Multi-touch is not possible using this technology.
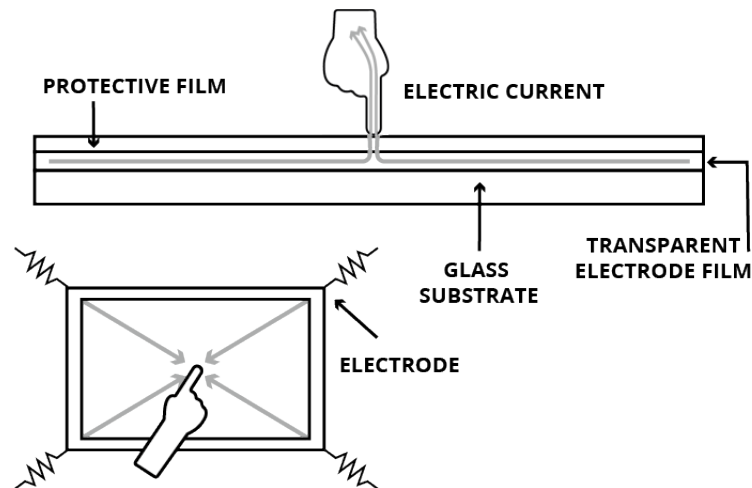


Figure 7: Surface capacitive touch technology.

**Projected capacitive** technology (Figure 8) is similar to the previous technology but has the big advantage to allow multi-touch interaction. The electrode film is now composed by two independent layers that are able to determine the X and Y coordinates of touches. This is generally the technology used in the current smartphones.

**Surface Acoustic Wave**, or SAW, touch detection systems (Figure 9) use a different approach with ultrasonic waves. Some piezoelectric transmitters and receivers are disposed on the display borders. When a touch occurs, parts of the waves are absorbed and the respective receivers are able to detect the touch location. SAW touch systems are precise enough (depending on the quality of the components) and allow to have a clear image on the display. However, they are susceptible to dust and other contaminants. Moreover, hard objects that do not absorb the waves cannot be used with this technology. A SAW touch detection system is able to detect multiple touches.
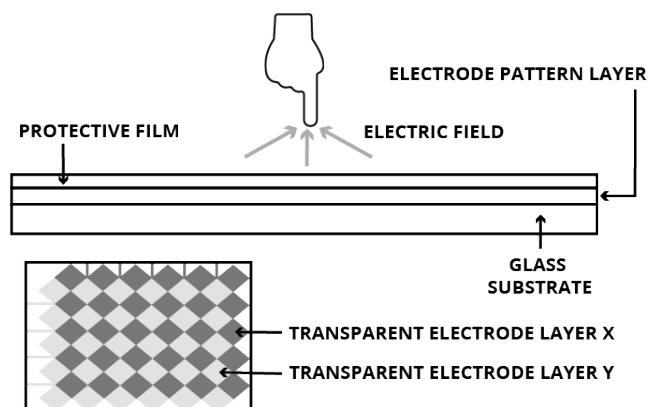
Figure 8: Projected capacitive touch technology.
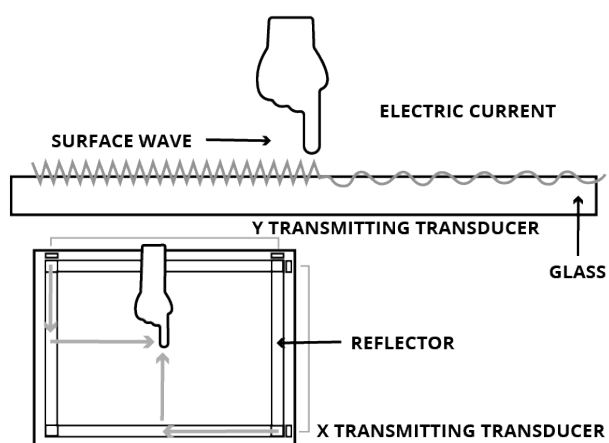


Figure 9: SAW touch technology.

**IR** technology (Figure 10) can be independent from the display. Indeed, this touch detection system is composed of a frame of IR transmitters and receivers composing an invisible grid of IR beams on the display surface. When whatever object interrupts the beams, the system is able to determine the X and Y coordinates of the touch. Moreover, some of this apparatus can detect also the size of the detected object. IR technology is accurate, but suffer some known problems. The first one is that the invisible IR grid is not thin enough and when a user quickly moves his finger from a point to another close point, the system might not be able to detect the release of the touch identifying the two touches as if originated by a unique finger. Another instance of the same problem is that the IR beams can be interrupted by objects that are very close, but are not touching the display surface. IR technology allows to detect more touches simultaneously. Their price is generally high, but they can be customizable. This last characteristic is very important in the field of LVDs. As we learned, they are realized by several monitors and there are no standard sizes. The absence of standard sizes for LVDs implies the need of personalized touch overlays. For this last reason, IR technology is the usual choice for commercial LVDs.
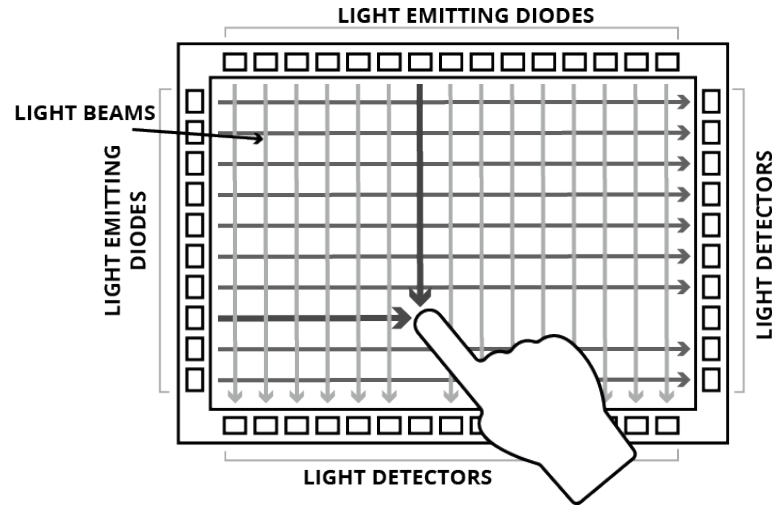


Figure 10: IR touch technology.

**FTIR** technology (Figure 11) is the technology invented by Hann [17]. Basically, FTIR exploits the *no refraction condition* occurring when the light enters from a material to another. This status is called *total internal reflection* and is verified when the value of the angle of incidence of the light beam exceeds a threshold. This threshold is function of the refractive indexes of the two materials. Hann

used an acrylic material flooded with IR beams to create this condition. When a touch occurs with the material, the angle of incidence changes and the rays become *frustrated*. In this state, the rays pass the material and an IR camera captures the point where the beams exit to detect the touch position. It has multi-touch capabilities and high precision. However, it is sensitive to ambient light and cannot recognize objects.
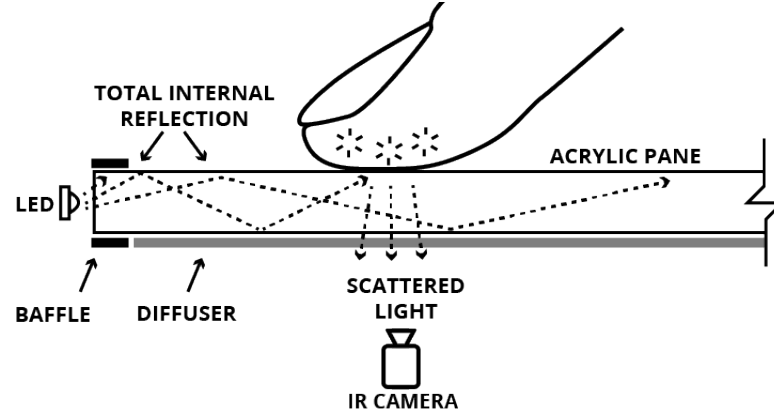
Figure 11: FTIR touch technology.

## 2.3    Large Display with Multi-touch

Returning to the Han conference, the most important message he gave about multi-touch displays was that this kind of technology could have changed the way how the human interacts with the machine. Han also faced the problem of data visualization saying that even though tons of data are present, we are not able to access and navigate data in an effective way and the introduction of a new way of interaction may help to drill down and explore data more effectively. LMVDs offers new degrees of freedom for the data navigation problem, and consequently represents an opportunity to face already studied problems in a different manner. In the past years, research on LMVDs was conducted mainly on horizontal large displays since they were more diffused. A push in the direction of vertical displays is noticeable from the various studies done in the recent years on how to interact with LMVDs, how to visualize and navigate data using LVDs, and how LMVDs usage can improve the way information is captured and learned by people and children. Knudsen et al. [3] understood the opportunity that LMVDs may give for data visualization and exploration, but recognized that the ways of interaction that these systems currently provide are limited to support the pointing and the windows movement.

They studied several groups interacting with a physical whiteboard, but imaging to interact with a LVD. Knudsen et al. proposed in the study new interaction techniques and design ideas such as to dedicate the central part to the user thinking, to use peripheral display areas for persistent data, to enable large gestures and the possibility to interact when the user is not at a touch distance with the display. Bezerianos and Isenberg [18] analyzed how variables visualization, such as length, angle and area, is affected by the proximity of the user to the display, albeit in a different level depending on the analyzed variable, and that perception accuracy has the same level when the user is free to move or stands in a unique place, but is less time consuming in the second case. There are many studies on how LMVD technology can improve learning and generally, when used in the right way, it improves several factors of the learning sphere, such as attention and memorization [19]. An interesting study about LMVDs for learning was conducted by Agostini, Di Biase [5] concluding that for digital natives [20] is important to include technology in their learning path.

### 2.3.1    <u>Multi-user Environments and Large Displays</u>

The last important topic this thesis adds to LMVDs is the presence of more than one user in the environment. This addiction increases the difficulty of using a software application on the display. The presence of more users should be considered in the development of certain applications, even though the management of the multi-touch capability is already able to support more than one user.

The large displays studied in this thesis are vertical ones, called also wall-displays. Most of the studies on multi-user environment are done on horizontal large displays, or tabletop displays, and generally only in collaborative environments [21], [22]. The study results can still be used, but not completely since some assumptions do not hold for wall-displays. For example, users using tabletops have fixed locations and are generally sitting down; or windows orientation is not a problem on wall-display since the windows have all the same orientation. Recent research demonstrated how the employment of LVDs with high resolution that gives people the opportunity to perform physical movement in front of the screen in place of virtual movement [2], increases the performance and productivity obtained by users performing a task and their satisfaction too [23], [4]. With the addition of the cost reduction and the easiness of composing and mounting displays, vertical displays are becoming popular also in the research field and multi-user collaborative environments are starting to be researched. One of the most important contributions to this kind of environment is given by Jakobsen and Hornbæk[24] which

studied the problem of collaboration using a LMVD, giving important statements and suggestions on how LMVD technology should be employed. Moreover, this study confirms how the interaction with tabletops and LVDs differs. Then, it shows important results on several aspects such as how much space people tread during a task, how people are available to share part of the space on the screen and the percentage of area used of the screen. Probably the most important conclusion of the research is that high space and resolution offered by this LVDs are enough to host many users at the same time and various collaboration styles. However, this is only an important starting point for the research on the interaction with LMVD, as stated by the authors. LMVDs employment in every field is slowed down by the absence of directives and a development paradigm. This makes the development of useful, usable and more interesting applications harder since developers have to employ ad-hoc solutions for every new application. It is not enough to extend an operating system to work with these big resolutions and this particular input mode for the problems exposed in the previous paragraphs. Dedicated environments offering basic interaction items may be a solution, even though similar solutions tend to spread into too many proprietary systems once the product is well settled into the market. One open source solution that tries to deal with the problem of very LVDs is SAGE2 [6]. SAGE2 is the system employed by this thesis to develop its application.

This thesis poses its attention on a multi-user, but competitive environment. As said above, Jakobsen and Hornbæk [24] gave a huge contribution in the world of LMVD in collaborative environments. The word *competitive* in this scenario is used to indicate a situation when at least a user follows a personal goal, which is not shared with the other users. Even though there may be other shared goals among the users, in such a case the multi-user environment cannot be considered only collaborative since following a distinct personal goal would probably bring a user to make different decisions compared to the situation in which a user has only common goals. Different situations can be extrapolated from this scenario. The main two are when a single entity is at the center of the attention of the users, and when every user controls a single entity (obviously these are not reduced only to competitive environments). In the first case, the only way to maintain order among the user is giving some rules to respect. For example, the interaction with the single entity may be mutually exclusive and may follow a simple round robin style to assign the control to a user, the interaction may be limited to a maximum of users per time, or a non-partisan user may direct how the users interact with the entity. In every

case, the users must coordinate themselves following some rules in order to avoid an unproductive and difficult interaction with the entity. In the second case, a simple solution may be to create a separate context for each user and eventually bringing the information of each separate context into a unique place. An extreme, but interesting, solution for the first case would be to transform it similarly to the second one, so as to replicate the single entity, one for each user. Nevertheless, this solution is not enough since the interaction of the other users have to be reproduced in real time. This might lead to higher confusion since a user interacting with the replicated entity may be confused by the reproduction of the interaction of the other users. One way to mitigate this problem might be introducing an avatar, or the stylized part that is actually interacting with the entity, for the other users performing their same actions. This last solution probably would require many resources, but might be an interesting way to tackle the problem. We have to pay attention that the paragraph above is not related to a specific context, such as the one of LMVDs, and the problem found in a context can be easily extended to others. Virtual reality is one of these context. Actually, the presence of a technological context, such as the one of LMVD, may be an advantage since replication and communication among entities is easier, but a technological context has also disadvantages. For example, in the LVD context there is the disadvantage that an item is confined within the two dimensions of the display, or there are people who are not able to interact nimbly with a multi-touch display or technology in general. These are all interesting challenges that may weaken the effectiveness of vertical LMVDs in multi-user environments.

## 2.4 The Conference Scheduling Problem

We want introduce here the problem that is the user study object of this thesis: the time scheduling problem. Actually, in our study we used a precise instance of the problem that is the *session scheduling process at a scientific conference*. However, before starting to talk precisely about our problem instance, we will describe and discuss in this paragraph the aspects that make the time scheduling problem a non-trivial problem. To explain it, we will take into account another instance of the time scheduling problem: the time table design problem. This problem is analyzed by Garey and Johnson [25] and is shown to be an NP-complete problem. Being NP-complete means that no algorithm can solve the problem in polynomial time. We talked about the time table design problem even though it is not the object of our study, but the previous mention was necessary to understand the complexity of a typical

time scheduling problem. We took into account this instance for two reasons. The former is because our goal here is not to give a mathematical representation of the conference scheduling problem. The latter is because the time table design problem is a known problem and plenty of material is already present in the literature about it. Indeed, the goal of this thesis is not to create an automatic tool to find a good solution of a hard problem. We want to study humans behavior during the tentative of finding a solution to a complex problem with the support of a LMVD influenced by the presence of other users acting for both a shared and a personal goal.

### 2.4.1  Why People Still Organize Conferences Without Technology

There is another explanation why conference session scheduling is the object of our user study. Although there are several tools dedicated on finding a good scheduling for a conference (or a time table in general), people are discouraged by the goodness of the configuration found by these tools. This discouragement is caused by the fact that it is hard to express the importance of constraints. For example, for some people certain time slots can be more valuable than others, and also trying to capture this importance through a number between a minimum and a maximum, someone might not be able to express a time slot importance coherently with respect the others thought. For this reason, people still prefer to organize conferences by themselves without the usage of the artificial intelligence.

There is another reason that justify the fact that people still organize a conference in a traditional way. This reason goes over a mathematical analysis and explanation. To explain it, we want to take into account the thought of some experts about the organization of conferences. An interesting article[1] related to conference organization collects in eighteen paragraphs with advices given by people who already organized a conference. The one we want to point out is called "Programming is Curation and Design" that contains some precious witnesses of why people still uses a real whiteboard to organize a conference. A conference is not only a succession of talks, but an event that should entertain, interest and inform people of the field about the last discoveries. A conference generally incorporates many branches of a more general one. Someone interested to a specific branch should be able to follow at least the most interesting presentations of the field of his interest. Scheduling connected talks on the same time slot, but in different rooms might be a damage for that specific branch while might unfairly

---

[1]https://medium.com/tedx-experience/how-to-organize-a-conference-567fb50ccdbd

advantage some other fields. Andy Budd of 'UX London'[1] and Jeffrey Zeldman of 'An Event Apart'[2] make an interesting comparison between a conference and a narrative. The second one states that a conference "is a designed experience"and "an educational and emotionally considered narrative". He tries to explain how people affiliated to each conference sub-topic attending the conference should be able to share evenly the same experience throughout the days. Probably, the most important statement he says is that "The order in which sessions take place is critical. (...) related ideas should be presented in blocks that help attendees see connections across sessions and topics". To conclude the digression on this article, we want to cite also Steve Baty of 'UX Australia'[3] and 'Interaction'[4]. He states that "it's difficult to overstate the importance of a well-curated and scheduled programme versus an ad hoc collection of talks."And this thought is aligned with the previous ones. Baty also says that a schedule per se "(...) takes just a few minutes (...), but can take a few days to get the flow, tempo, and structure right. The bigger the programme (days, tracks) the more time it can take". The last sentence is very meaningful for our treatment of the problem because it is evenly aligned with the mathematical view of the problem. Indeed, we learned before how increasing the number of the presentations, increases the complexity of a good schedule.

### 2.4.2 The elements of the Conference Scheduling Problem

There are nine main entities in this problem. They are listed and explained here:

- Days (d): the days along the conference will be held;

- Rooms (r): the available room in which the paper presentations will be done;

- Hours (h): the temporal intervals in which a day is subdivided;

- Topics: the main areas in which papers are categorized;

- Plenary sessions: sessions that unavoidably prevent the schedule of other presentation in the same hours.

---

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

[2]http://aneventapart.com

[3]http://www.uxaustralia.com.au

[4]http://interaction16.ixda.org

- Collapsible rooms: the rooms that can be collapsed in a bigger one to increase the room capacity;

- Room capacity: the number of attendees that can be contained in a room;

- Papers: the papers to present in the conference;

- Expected attendance: the expected attendance of each paper presentation;

- Constraints the constraints that prevent the scheduling of a paper presentation in certain slot hours.

| | DAY 1 | | | | DAY 2 | | | | DAY 3 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ROOM A | ROOM B | ROOM C | ROOM D | ROOM A | ROOM B | ROOM C | ROOM D | ROOM A | ROOM B | ROOM C | ROOM D |
| HOUR X HOUR Y | | | | | | | | | | | | |
| HOUR Z HOUR W | | | | | | | | | | | | |
| HOUR P HOUR Q | | | | | | | | | | | | |
| HOUR R HOUR S | | | | | | | | | | | | |
| HOUR T HOUR V | | | | | | | | | | | | |

Figure 12: A typical sample table used during a schedule conference session. We can observe three available days, four available room and five available time slots. There are a total amount of sixty potential time slots.

These characteristics generates an availability of d x r x h potential time slots that can be assigned to paper presentation. This can be observed in figure X where a typical table used for a conference schedule is represented. However, some details make the schedule a task not easy to perform.

- Plenary sessions, by definition, request the participation of all the attendees, or prevent anyway to schedule other presentations in the same time slots of the same day. They are like keynote, lunch and banquets. To simplify the representation, they can be intended as special topics with constraints that force their schedule in unique hours and in all the rooms of a day.

- The expected attendance of some presentation can be higher than the higher room capacity available. In such situation, some adjacent rooms belonging to the set of the Collapsible Room have to be merged in a unique and bigger one. This situation allows overcoming those situations in which expected attendance is higher than the higher room capacity, but at the same time reduces the room availability by two or more in a time slot. This leads to increase the conflict level among the other presentations.

- Paper presentations are bonded by several constraints. Indeed, different people present different papers and consequently their presentation underlies to the personal availability of each speaker. Other constraint might be that some paper cannot be presented in the same time slot since they are relevant papers of the same topic and attendees should be able to follow both of them.

To obtain an admissible schedule it is necessary to schedule all the paper presentations in the available time slots respecting the room availability, the constraints, and avoiding presentations overlapping. Nevertheless, this is only a simplistic view of the problem, even though enough effective to show the main mechanisms of the task. Indeed, as we pointed out the previous paragraphs an admissible solution is usually not sufficient. The final result has to be a quality schedule and, as we learned from the witness of some experts, several factors influence its quality. Actually, there is a factor in which we are more interested. The different topics presented in a conference should have papers presented evenly among the more willed available days and hours. Indeed, along the available days some hours and some days are generally more valuable. Nobody wants to have a presentation early in the morning while generally everybody desires the central hours. We pointed out this aspect because this is what makes people negotiate during the creation of the conference scheduling and this is basically the object of our study.

### 2.4.3    The Traditional Approach

Now that we know how a schedule can be admissible and how it can be considered a good schedule, we describe in this paragraph what is the procedure used to schedule the paper presentation in a conference. Actually, this process is independent by the technology used, but we will take into account the traditional approach to facilitate the explanation. From now on, we will use traditional approach and whiteboard approach as synonyms. The conference organizer selects a location and a room that is

in accordance with all the participants to the conference session. Participants are composed of a chair for each topic present in the conference. Each chair is conscious of the particular constraints of each paper presenter belonging to his area. One or more conference organizers participate to the session to oversee the scheduling session. A table is drawn on the whiteboard in accordance with the given days, hours and rooms, and some time slots are occupied by some plenary session. Each paper title and author is transcribed on a colored post-it where the color represents a particular topic. To each chair are given the post-it belonging the topic he represents. Then two phases composing the heart of the task are performed:

1. *First phase* or *Initial assignment phase*: this is a turn-based phase. Chairs approach the whiteboard all together or in small groups. Following a round robin procedure, each chair places N post-it on the table, occupying time slots. N is a number defined previously that generally varies between one and three. In this phase, even though somebody may desire an already occupied slot, he cannot mark it. The chair will be able to negotiate that slot in the next phase.

2. *Second phase* or *Conflict Resolution phase* or *Negotiation phase*: this is the phase where conflicts are resolved and where the negotiation comes into the field. If a chair desire to move one of his presentations to an already occupied slot, he has to negotiate it with the chair who is occupying the slot. Multiple private conversations can happen at the same time. A typical situation is the following one: chair A desires the slot occupied by chair B, but chair B cannot find any other interesting free slots. However, chair B is interested to the slot occupied by chair C. Consequently, chair A and B talk with chair C trying to negotiate the last slot. This behavior brings to a domino effect. Indeed, changing the time slot of some presentations, might lead to move several other presentations. Moreover, in this phase are solved also other conflicts and constraints, such as the presence of two incompatible presentations in the same time slot.

The process has to be a democratic process and one topic has not to be advantaged over the others. One of the main problems is that in this process, some chairs tend to be more interested exclusively to their topic. However, it is worth to remember that every participant has two goals. The one shared among all the participants, that is to maximize the goodness of the final schedule, and the personal one to maximize the goodness schedule of his specific area. Thinking exclusively to maximize the personal goal will lead to satisfy only half of the participants overall goal while damaging also the other

participants overall goal. Reaching an admissible configuration that satisfies the interests of every chair is not an easy task.

# CHAPTER 3

# RELATED WORKS

We will show in this chapter interesting studies on LVDs and LMVDs. First, we describe two studies that talk about the state of the art of LVDs. Then, we deeply analyze two studies on LVDs: the first one on a collaborative environment, the second one on a competitive environment.

## 3.1    Studies on Large Displays

Although with some difficulties and without having found a precise role into any field, LVDs are still actively used and researched after roughly twenty years from their introduction. Lots of studies have been conducted and many applications were realized based on LVDs. Some assumptions resulted faulty and consequently the applications, but this is the normal history course of a new and emerging technology. In fact, often the committed errors are caused by missing knowledge acquired later in the years or by not yet discovered technologies. Moreland [10] examined the problems emerged in these years about LVDs and analyzed the four most important lessons learned from studies and applications on LVDs. For each of these lessons, Moreland proposes a redirection. Here we briefly go through these lessons to understand some LVD barriers and suggestions on how to overcome them:

- *"We know how to build tiled display"*: after many studies, we are now able to build LVDs as tiled displays. There are mainly two techniques. The former is using adjacent projectors with front or back projection techniques on a white panel. Moreover, thanks to advanced techniques we are now able to tackle color balancing [26] and image alignment and calibration [27]. The latter technique is becoming the leader nowadays thanks to the price reducing and the ease of assembly. This techniques consists of creating a mosaic of displays. Their main disadvantage is the presence of bezels, as we already explained in the background chapter.

  Redirection: Now that we acquired this knowledge, industry should be able to provide large tiled displays ready to be installed. Many companies in fact started to sell these solutions in the recent years.

- *"A pixel is not the same thing as a datum"*: at the beginning we were brought to think that increasing the pixels in an image, increases the quantity on information acquired by a human.

Studies [28] demonstrated how a human acquire information mostly on a small screen part and everything around is not assimilated. Moreover, increasing the cognitive workload can reduce the observer attention causing a further information loss [29], [30].

Redirection: we are not experts on how to use LVDs and how to effectively develop application on them. There are many open questions such as how to effectively visualize data, how to collaborate using these devices, what are the right size and resolution for a certain task, and many others of them. Some challenges posed by the previous questions are tackled through research and user studies and some of them will be discussed later in this paragraph. However, even though they make some important conclusions, they are still and only a starting point and many others study should be conducted.

- *"Large format displays can impede interaction"*: this problem is examined in the previous section enough accurately, but here is addressed more under the input device aspect. As shown by some studies [31], [32], keyboard-mouse interaction has many problems with display size larger than usual. Moreover, they implies a fixed position and, depending on the used application, it might be inefficient. Instead, remote devices are often imprecise.

  Redirection: As suggested in the previous chapters, a new interaction paradigm has to be developed for LVDs. Obviously the usage's purpose guides the way of interacting with the application, but here the matter is that LVDs has been used with input devices specialized for other purposes. There are many studies related to human and LVD interaction with various input devices such as touch screen, wand, and pointers [33], [34], [35], [36], [37], [38], but until appropriate interaction mode will be discovered and made standard, they will result very inefficient.

- *"Display technology is a means, not an end"*: LVD can aid people in many tasks such as data visualization or group collaboration. In fact, they are usually employed in these situations and others. However, they are often inserted in a context as they are – as a huge display – without implementing any specific solution on them. For example, porting an application on a so large display might increase the involvement of a group of users, but slow down the application used at the same time, since it is difficult to control its interface with a high resolution.

  Redirection: As the title of this point states, LVDs are not a finished product ready to be employed in every context, but a mean to improve some factors of that context. Actually, we do not really

know if LVDs can improve some factors in every kind of context. For example, it seems that they are useful in dynamic map analysis such as path-finding and context awareness in military situations [39], [40], [41].

There is still a lot to learn on LVDs and user studies are a precious way to understand how to make LVDs more usable. For this reason, we want to take in consideration some studies that examine the user behavior and the advantages while they perform different kind of tasks.

To understand the human necessities from a LVD, Knudsen at al. [3] performed a very interesting study on groups of users performing tasks from various domain imaging to interact with a LVD, but actually interacting with a whiteboard where attached papers represented the data. This choice was done to avoid that users behavior were influenced by an imperfect data or environment simulation since, as we learned previously, LVDs are a technology that still need to find interaction models. Moreover, taking into account eleven different domains can bring to make specialized conclusions and also generalized ones, if more characteristics come up from the different domains. They performed workshops related to business, research, and arts. Here we want to examine only the general results to understand the human necessities pointed out by this study. They are:

- *Persistency*: this is the most frequent use of the whiteboard space and consists of creating partition areas and assign to each of them a particular purpose. Some of them were dedicated to host data for long periods of time;

- *Showing data and interaction*: two main ways of showing and consuming data were used. The former consists of putting side-by-side different form to represent the same data or a drill down on a specific dimension with an existing graph. The latter consists of having a one-by-one data representation where filtering was enabling by a simple interaction.

- *Space to spread out data*: typically, users needed of an abundant space to spread out a certain category of data to have them temporary side-by-side and to select or modify them.

- *Data processing flow*: users desired to have the possibility to be able to take track of data processing flow and the ability to see previous data states or previous snapshots, for example with points connected by a path where points represent the data states.

- *Movements*: as easy to intend, users tended to approach the whiteboard to get more details from the data, then tended to move away to get an overview of them. However three main positions where assumed by the users with respect to the whiteboard:

  - The closer one, to interact and deeply observe data;
  - The middle one, giving the back to the whiteboard and to interact with other users;
  - The farthest one, facing the display to observe data in their entirety.

- *Gestures*: Users performed many gestures and most of them are already classified by literature. They were categorized in three main categories as on-screen, in-air, and in front of the screen.

All these factors are interesting points to take in to consideration when LVD application are created. Even better, their implementation feasibility should be studies in the hypothetic development of interaction paradigms or an underlying framework able to offer these capabilities.

A study conducted by Bezenarios et al. [18] tried to understand the way users can visualize and consume data on LVDs in an efficient way. They compared the accuracy and the time consumed to acquire and interpret data visualized on a LVD in two situations, static and moving. The former tested two fixed position at left-most of the screen to stress the most extreme distance and since a conjecture on symmetric result from right and left was posed. The two position where at a close distance, at 60cm, and at a far distance, at 240cm. It resulted that both positions were affected by errors and that tasks in full view were performed better. Moreover, they noticed how the length variable is relatively unaffected, while area and angles are consistently affected. The latter situations allowed users to move. It surprising resulted that the information perception was accurate as the static situation, but required twice the time consumed by fixed location. This results are to take in consideration in the occasion an application showing data is developed.

## 3.2 Collaboration on Large Displays

LVDs allow operating in different kind of scenarios considering the presence of one or more users, the type of task that can be collaborative, competitive or independent, and the physical presence of the individuals. Figure 13 represents graphically the possible scenarios for large vertical displays.

There are many studies concluding that LVDs offers several benefits when used by a single user. These benefits occur for tasks requiring high cognitive load [23] also when the interfaces was not
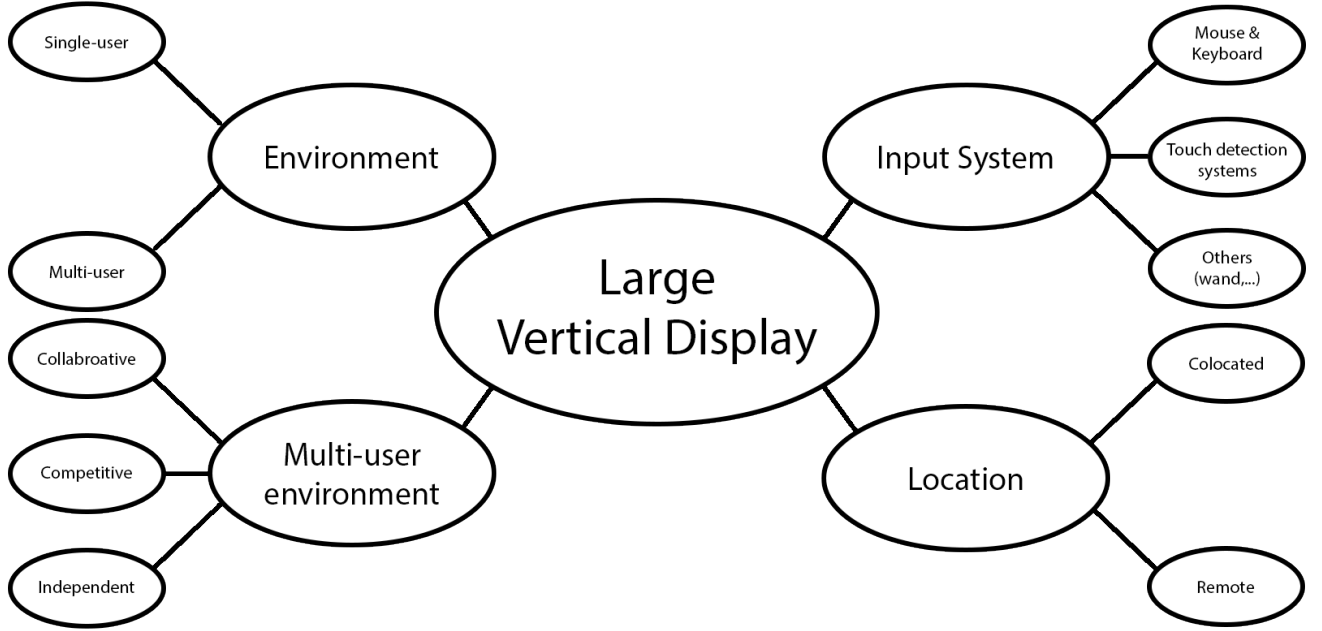
Figure 13: Possible scenarios for large vertical displays.

properly redefined for large resolution, in immersive 3D navigation [42], [43] and data visualization [3]. These studies show how there are interesting performance advantages using LVDs in comparison with a normal screen. It is curious how in [44] it was evicted how a large screen can be a cheaper substitute of a Head-Mounted Display for virtual environment obtaining approximately same results, but with a lower cost.

Regarding single user using a LMVD, there are not many studies and no one contains user study proving the effectiveness of the system. Consequently, it is not possible to state if a LMVD improves a single user task. Although not completely related to LVDs, it is interesting observe the study done by [45] where mouse, single-touch, two finger touch and multi-touch were compared to accomplish a task. Users in front of an oblique screen have to press or touch more blue circles as possible avoiding orange circles. It resulted how multi-touch interaction brought to significant improvement with respect to the mouse interaction, even though the highest miss rate was obtained by multi-touch interaction.

When scenarios where more than one user interacts with a LVD are taken into account, the number of studies and conclusions increases. An interesting study in the direction of collocated collaborative tasks has been conducted by [46] where it investigated the spatial strategies employed by pairs of users during a high cognitive load task, such as extracting information from several text documents. Furthermore, the study states how the LVD is comfortably used as an external memory, like if it was

a virtual whiteboard, where they tended to organize the information in clusters. Moreover, the study compares the usage of data-centric tools and function-centric tools concluding that the first ones suit better a large screen and allow to extract more information within a document. Despite these tools obtained better results in the study, is worth to mention how function-centric tools are useful on a LVD to filters set of information and to understand the useful ones. Therefore, when the number of information increases also the high resolution offered by a LVD is not enough and function-centric tools have to be used to organize efficiently the data. Talking about LVDs and collaboration, it worth to mention the SAGE2 [6]. SAGE2 is an open-source middle-ware offering the capabilities to multiple users to share an operating environment where is possible to display and consume different varieties of data intensive information. Moreover, SAGE2 offers the capabilities to display the shared environment on displays of not fixed resolution and in a remote way. This system has also the capability to take in input touch information, even though at its current state the touch input manager offers only the possibility to multiple users to open and access in every context the user interface where it is possible to open applications and go through a collection media files. The EVL at University of Illinois at Chicago created SAGE2. EVL uses SAGE2 mainly in the Cyber-Commons room on a LVD equipped with an infrared multi-touch overlay allowing multiple users to interact with the wall display simultaneously. SAGE2 offers interesting way of tackling data intensive problem and three interesting use cases are presented in [6]. One of this consist of judging photos of a contest where judging panels are distributed in two different campuses. The judges are able to share a context within a web page where they can visualize the photos an leave comments in a respective shared notepad. Images can be enlarged, moved and the same can be done for every notepad application. Moreover, a video-conference allows judges to communicate as if they were in the same place. This use case is a very interesting example of collaborative task that is assisted by LVD technology.

### 3.2.1   Two Studies on Collaboration Using Large Display

Here will be showed two interesting studies on LVDs in multi-user environment. The first study is a strictly collaborative task were pairs has to work together in order to solve a load intensive task. The second study is a collaboration task where a competitive component influence the users' behavior.

### 3.2.1.1    A collaborative study

Another important contribute regarding collaborative is given by Jakobsen and Hornbæk [24]. This article offers interesting conclusions on characteristic of LVDs with multi-touch when used by more than one users. First of all, this study provides some contrasting results with previous research on wall displays where difficulties on sharing and collaboration where found. Instead, it concludes that several users can effectively use them at a time in an efficient way. Other conclusions regard the fact that the very high resolution offered by these displays allow people to work up close and that there is enough space to employ different collaboration styles. All these interesting conclusions are obtained performing a user study on a collaborative task and more specifically a problem-solving task that involves a set of documents. It worth to mention the way this study was conducted since it makes conclusions on most of the same characteristics regarding the human-interaction with displays this thesis wants to analyze. These characteristic are related on how users are willing on sharing screen space, on the space distribution utilized by the users, on the distance with the display, on their movement, on the type of verbal communication, on their proximity to each other, on the employed collaboration style and similar ones. To capture all these characteristic, the study used the Stegosaurus dataset from the VAST 2006 challenge [47] that consists in a collection of document describing a certain scenario. The documents are new articles, images, reference documents, a map and a spreadsheet. The goal of the task was to find a hidden plot within all the dataset information. As a VAST's dataset, it was used in other studies such as the one of [48] but for different purpose. For example, Andrews studied the performances obtained using two different sized displays. In the one of Jakobsen and Hornbæk the goal is the human behavior and interaction respectively to the other participant and to the display. The apparatus used by this study was a 7,680 x 3,240 vertical multi-touch display composed of a grid of 4 x 3 displays back projected with 12 projectors each with a 1,920 x 1,080 resolution occupying a wall area of 2.8 meters wide by 1.2. To begin the study, users received a brief introduction to the LVD usage and interface followed by 10 minutes per user to take confidence with the apparatus with a sample and different dataset. Then users were explained on the task and about the type of documents at their disposition. After the introduction, the task briefing and a break, the task started. At the end of the task a summary of their findings were asked them to completely conclude the study. Eventually a questionnaire regarding their personal tracts and their collaboration style was given them. During

the whole experiment, users were video and audio recorded to analyze deeply the characteristics listed before.

### 3.2.1.2    A competitive study

Another study that is worth to mention is the one of Birnholtz et al [49] where a competitive task using a LVD is studied. The difference between this study and the one proposed by this thesis is the input method. In the Birnholtz et al. study, it is examined the behavior of groups of people interacting with a LVD in two different condition: using a single shared mouse and one mouse per participant. They examined three main characteristics regarding the perceived competitive behavior, how the task performance was conditioned by the users' seating position and the group behaviors. Regarding the first characteristic, their initial hypothesis were that using the single mouse interaction would encouraged a more competitive behavior because who hold the mouse probably would act more in his own interests. Instead, it was statistically significant that users acted more competitively in the multiple mice condition. This is caused by the fact that in general when the input device is unique, all participants monitor what is happening in the main part of the screen, while using multiple input devices users rarely monitor the events happening on the main part. Instead resulted not statistically significant the users seating position since users acted evenly on the screen space independently on their position. There are also interesting results related to the group behaviors. The usage of multiple mice actually resulted in a more parallelized activity as hypothesized by them. However, the task progress seemed to be similar using the two input methods, even though multiple mice usage resulted in a time interval to be more efficient. Regarding group discussion, users talked more during the single mouse interaction. One important things to signal is that more frustration was evinced by the single mouse interaction. Simple request to the mouse owner were easily accomplished, while it was harder to understand by the mouse owner complex or vague directives and this events generally resulted in a change of mouse control. An interesting discussion displayed in this study, shows the three participants strongly arguing about the fairness of an action of one of them. It resulted in a slight throw of the mouse on the table. Then, another participant grabbed the mouse and firmly stated that the action he just performed was fair. This is an important example of frustration that can occurs during a competitive task. These emotional demonstrations generally happen because it is not easy to reach the common goal that would define the task termination when each user is also trying to maximize

his personal goal. Considering that frustration can increase with the time passing and non-functional problem, such as an exhausted whiteboard marker during a competitive task using a whiteboard, we bet that a consistent and efficient electronic application can reduce the frustration spreads among the participants.

The study conducted by Birnholtz et al. [49] has a little different purpose with respect to the one of Jakobsen and Hornbæk [24]. While the latter were interested to study also characteristics more related to the human interaction with a LVD using a multi-touch input approach, such as territoriality, the former is designed only to analyze the behavior among users using a LVD with mice input. In fact, participants simulated to be editors from separate sections of an agriculture newspaper. The common shared goal was to lay out the first page of the newspaper with a selection of different news. Constraints such as a maximum of fourteen news and not overlapped news were present. Every participant acts to be an editor representing a different section, such as fruit or protein, and has the underling goal to maximize the number of keyword related to his section appearing on the final first page. The participants were able to practice with the interface before the actual study started. After this, twelve minutes where assigned to groups of three people to complete the task, before with a single mouse and then with multiple mice, or vice versa. The task were repeated twice and alternated with both input devices. To analyze the characteristic listed above, the users where video and audio recorded and many data log records by the application regarding clicks and interaction with the news are saved. At the end of the study, a questionnaire is given to the user in order to capture more information relative to their experience. The used apparatus consumed a wall area of 5 x 1.8 meters with a LVD composed of a grid of 4 x 3 displays back where eighteen projectors back projected 1024 x 768 pixel each with a total resolution of 6144 x 2304 pixels. We will see how both of these configurations has dimensions similar to the configuration employed in the study of this thesis.

For completeness, or just curiosity, other interesting studies related to collaboration on LVD are the followings [50], [51], [52].

# CHAPTER 4

# IMPLEMENTATION

This section describes the implementation of the Conference Scheduler application and the techniques used to realize it. They are the result of several attempts to implement an effective simple interface for a LMVD. This interface has the capability to be activated either by a mouse or by a touch input. Since the application is designed for a LVD, its interface has been realized using a library for SVG graphic, that is D3. In such a way, the graphic quality is not affected by the resolution since scaling SVG images allows to preserve the quality. Other SVG libraries can be used obtaining the same effect, such as SnapSVG. However, we will use D3 basic commands that are generally contained in every SVG library. Although the application runs within the SAGE2 system, all the techniques shown here are completely independent of SAGE2.

This section starts describing a technique used to create a simple interface for a LVD using an SVG library. The technique will be demonstrated through three toy examples. The first example shows some words depending on the button pressed. The next two examples will add more interaction capabilities by showing how to differentiate multiple touches, and how to manage each single touch lifetime. This interaction enhancement will be shown creating an interface for an image gallery. At the end of the chapter, we dedicate three different sections to explain how the previous techniques are applied to realize the Conference Scheduler interface, how the application is structured and how it works.

## 4.1    Simple Techniques to Implement Large Display Interfaces

The LVDs treated in this thesis are those whose resolution is not predefined since they are composed by several displays with a homogeneous resolution. Indeed, the interfaces we want to realize are independent from the resolution where they are visualized. In order to obtain clear and sharp interfaces, we employ an SVG library to realize SVG graphic.

SVG graphic has the main capability to do not lose quality when scaled. Indeed, an SVG image is composed of shapes. These shapes are described by mathematical formulas. Thanks to these formulas, it is possible to scale the image without losing quality contrarily to raster images composed by bitmap. Indeed, scaling a raster image brings to show the pixels composing the bitmap. An SVG file is defined

as an XML file where shapes are described through tags and properties. It is possible to specify paths or lines, basic shapes (rectangle, circle, ...), text, colors, filling colors, gradient, animation, and several other features. However, it is unusual to realize SVG graphic specifying those features directly writing the XML file. There are two common ways to realize SVG graphic: using drawing software, such as Adobe Illustrator or Inkscape, or using an SVG library, such as the javascript library D3. Our application is completely written in javascript using the D3 library as support to realize SVG graphic.

### 4.1.1 Toy Example 1

We demonstrate the effectiveness of our interface for LVDs using a toy example. A display and some button compose this application. Pressing the buttons will result in reproduce a text within the display, that is initially empty. In this and the following paragraphs, we suppose the knowledge of basic javascript instructions. Once imported the D3 library, the first thing we need to realize is a container for our SVG interface. This can be specifying the following command:

```
var container = d3.select(body).append("svg").attr("id", "theContainer");
```

This will append to the DOM an SVG container with the id *theContainer* within the interface will be realized. We also save within the variable *container* a reference to the SVG container; we need now to set its position and its size. Since we suppose our application is the only protagonist within the LVD, we want that the application occupies the entire display surface. The following command positions the SVG on the top left of the screen allowing it to occupy all the web page, and sets a neutral background color:

```
container.style("position", "absolute").style("background", "#D2BD8E").style("left",
    0).style("top", 0).attr("width", 100%).attr("height", 100%);
```

The toy example is composed of a single entity containing the buttons and the display. However, others application interface can be composed by several components sparse over the interface. For this reason, we want to create a sub-container. This can be done using the group function offered by SVG libraries. Groups allow obtaining a LCS (Local Coordinate System) with respect to the one of the containing SVG. We can create and append it to the DOM using a similar command to the one used to create the SVG:

```
var interface = container.append("g").attr("id", "theInterface");
interface.model = [];
```

The last line of code attaches an array that will serve as model for our interface. Indeed, before to insert and visualize the elements within the SVG, we want to create a consistent model that allows to specify the shape of the interface. Furthermore, having a model allows the possibility to easily specify properties that can be later transformed in desired visual styles and behavior. Obviously, a sophisticated behavior has to be created from scratch and we will discover later how.

The core of the model creation is composed by four functions:

- *createItems* is a tricky function that allows to create homogeneous items models in a grid style. For example, if we need a display and three buttons with the same size, we can create their base model by invoking twice this function, one call for the display, and another one for the three homogeneous buttons;

- *addProperty* simply adds a property to a item model. This property can be used later to set the graphic style of the items or assign them a behavior;

- *createGUI* is used to interpret the model and the properties of the items models to realize the GUI of the application;

- *within* checks if an input event on the interface is within an item. This function will be used to check if a touch occurred on the buttons of the toy example.

For each section of homogeneous items we want to insert in the interface, we have to determine the following measures from a manual or mental sketch. These measures are the parameters used to invoke the *createItems* function:

- *amount*: the quantity items within the section;

- *limit*: the quantity items per line or per column within the section. The horizontality or verticality of the limit can be specified with another parameter;

- *verticalOffset*: it is the value in percentage of top displacement with respect to the top of the SVG group;

- *horizontalOffset*: it is the value in percentage of left displacement with respect to the left of the SVG group;

- *verticalSpan*: it is the value in percentage of the height of the section;

- *horizontalSpan*: it is the value in percentage of the width of the section;

- *limitPerColumn*: specify if the limit parameter has to act per line or per column. If true, the limit is intended per column, if false or not specified, it is intended per line;

- *container*: the SVG group representing this section.

The measures are expressed in percentage of width and height. To derive them, it is possible to follow the same method used in the Figure 14 and Figure 15. In the toy example interface, we identified two homogeneous sections: the display and the buttons ones.
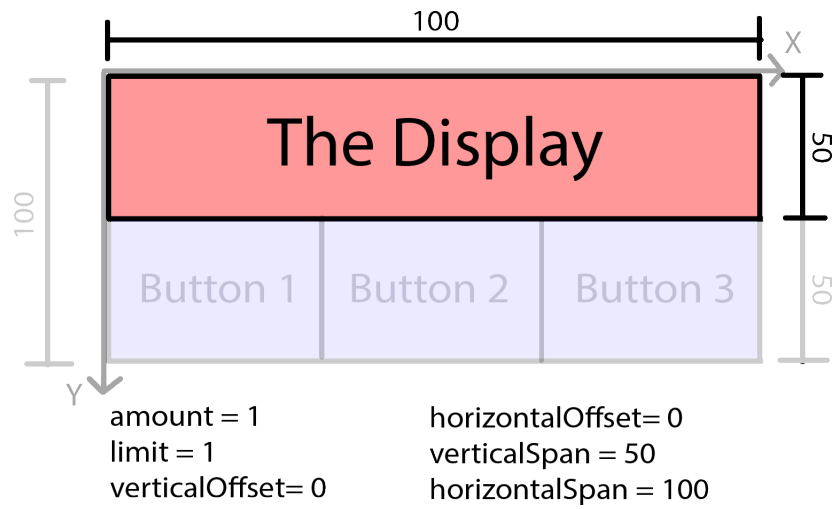


Figure 14: Toy example 1: the guidelines explain how to derive the information for the display section.

As we can observe in Figure 14, a single item composes the display section and we set *amount* to one. Also *limit* is one, since the single line of this section contains an item only. The display is on the top left of the LCS of the group and this implies that both *verticalOffset* and *horizontalOffset* are set to zero. This section occupies all the interface width and half of the interface height that corresponds to set respectively *horizontalSpan* and *verticalSpan* to 100 and 50.

Following the same guidelines, we can derive the measures for the button section too. In Figure 15, we can observe the sketch of this section. Here we have three items, the three buttons. For this reason, we set *amount* to three. In this section too, we want that all the three items are on the same line and this corresponds to set *limit* to three. As we can observe from the left arrow in Figure 15, this section has a vertical displacement of half of the interface height. Hence, we set *verticalOffset* to 50. No
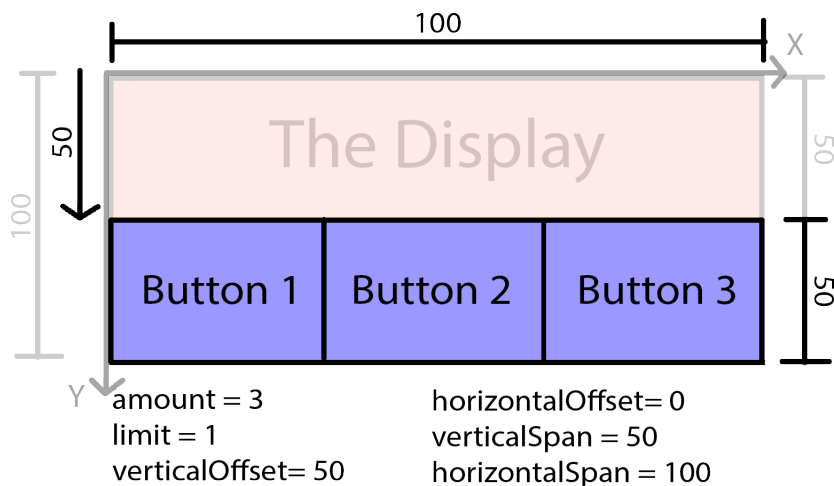
Figure 15: Toy example 1: the guidelines explain how to derive the information for the buttons section.

horizontal displacement is present and we set *horizontalOffset* to zero. Finally, since this section has the same size of the previous one, *verticalSpan* and *horizontalSpan* are set to 50 and 100 respectively.

We show now how the *createItems* function works. To avoid confusion and to simplify the explanation and the comprehension, we do not show the part of the function dedicated to the case of vertical disposition of the element, like if the *limitPerColumn* parameter was not specified or set to false.

```
function createItems (amount, limit, verticalOffset, horizontalOffset, verticalSpan,
    horizontalSpan, container) {
  var horizontalLimit = limit;
  var verticalLimit = Math.ceil(amount / limit);
  var items = [];
  for (var y = 0; y < verticalLimit; y++) {
    for (var x = 0; x < horizontalLimit; x++) {
      if (y * limit + x >= amount) {
        break;
      }
      var item = {
        r: y * verticalSpan / verticalLimit + verticalOffset,
        c: x * horizontalSpan / horizontalLimit + horizontalOffset,
        cSpan: horizontalSpan / horizontalLimit,
        rSpan: verticalSpan / verticalLimit,
        container: container
      };

      items.push(item);
      container.model.push(item);
    };
  }
  return items;
}
```

The function *createItem* creates with a tricky mechanism the models of all the homogeneous items in a section, and insert them into an array that is returned to the function invoker. This model will be later interpreted to be effectively visualized as a GUI. With the parameter *amount* and *limit*, we have specified the dimensions of a grid that will contain the items of the section. The number of items per line is the value of the *limit* parameter, while the number of the items per column is calculated with a simple mathematical operation. The first two lines assign these two values to the *horizontalLimit* and *verticalLimit* variables respectively. Then, the array *items* is initialized to contain all the items created by this function. Two for loops are used to iterate over the grid to create the items models and interrupted only when the desired amount of model has been created. For each iteration of the deeper loop, an item model is created and inserted into the *item* array. An item model is an object containing the following measures where the value are expressed as percentage of the group LCS:

- *row (r)*: the top displacement of the item;

- *column (c)*: the left displacement of the item;

- *row span (rSpan)*: the width of the item;

- *column span (rSpan)*: the height of the item.

At the end of the iteration, the items are added into the model of the passed group and returned to the invoker in an array. Now that we defined the *createItems* function, we can use it to create the items and the sections of the toy example interface. Since we already calculated the parameter for our display and buttons sections, we simply have to invoke our function.

```
var display = createItems(1, 1, 0, 0, 50, 100, interface);
var buttons = createItems(3, 3, 50, 0, 50, 100, interface);
```

These two lines of code simply create the base model of our interface. The next step is to add properties to our items models. There are infinite way to add, give name and interpret properties and behaviors. We create the *addProperty* function at this purpose. We want to add to the display an empty text, to color differently the three buttons, and to assign to the buttons a hidden word to visualize in the display once clicked. The *addProperty* function adds the specified property and value to the items passed as parameter. The items can be passed in an array or singularly. It works in this way:

```
function addProperty (element, name, value) {
    if (element.constructor === Array) {
        for (var i in element) {
            element[i][name] = value;
        }
    } else {
        element[name] = value;
    }
}
```

IN the above code is checked if the passed element is an array or a variable. Then, in the first case the property is added to all the contained elements, otherwise it is added only to the single passed element. In this way, we can add properties that will be later translated into a corresponding graphical style or behavior. We want to make our buttons *clickable*, to change their background colors, and to assign to them a hidden word. These lines of code perform these operations:

```
addProperty(display, "displayText", true);
addProperty(buttons, "clickable", true);
addProperty(buttons[0], "bgColor", "green");
addProperty(buttons[1], "bgColor", "white");
addProperty(buttons[2], "bgColor", "red");
addProperty(buttons[0], "hiddenWord", "Good Morning!");
addProperty(buttons[1], "hiddenWord", "Good Evening!");
addProperty(buttons[2], "hiddenWord", "Good Night!");
```

The next step is to generate the GUI. We delegate this task to a function that will interpret the model. This function has to be expanded in the case new graphical styles are desired. We call it *createGUI*. This function needs as parameter the group where to insert the SVG graphical elements. The function signature is the following:

```
function createGUI(theGroup);
```

Within the function body, we have to calculate the group dimensions with respect to the main interface, which is the main SVG. Since we specified the main SVG in percentage, the function will derive the real quantity in pixel of its width and length of it, using a javascript built-in function. Then, before to calculate the items measures, we want to define a padding around every element so that the interface results clearer. We can do this with the following lines of code:

```
var gWidth = theGroup.node().parentNode.clientWidth / 100;
var gHeigth = theGroup.node().parentNode.clientHeight / 100;
var padding = 5;
```

Now that we know the values of the width and height of our interface, we can iterate over all the items models and generate their respective SVG graphical elements. At the beginning, for each item we calculate its coordinates and sizes with some simple calculations, as we can see from this portion of code:

```
for (var i in theGroup.model) {
var item = theGroup.model[i];
item.x = item.c * gWidth + padding;
item.y = item.r * gHeigth + padding;
item.w = item.cSpan * gWidth - padding * 2;
item.h = item.rSpan * gHeigth - padding * 2;
...
}
```

For each item, we want to create an area representing it with proper dimensions and colors. Moreover, for the display, we have to add an empty text that will be populated later by the button actions. To achieve this goal, we use an SVG rectangle and a SVG text. We want to remember that all the D3 functions employed here are offered also by others SVG libraries. Within the previous for-loop, we add these two pieces of code:

```
var bg = item.bgColor || "gray";
item.rect = theGroup.append("rect")
.attr("x", item.x).attr("y", item.y).attr("width", item.w).attr("height",
    item.h).attr("fill", bg);
```

```
if (item.displayText) {
item.text = theGroup.append("text")
.attr("x", item.x + item.w / 2).attr("y", item.y + item.h / 2)
.style("dominant-baseline", "central").style("text-anchor","middle").style("fill", "black");
}
```

We separate them because we want to emphasize their independence. Indeed, we can add to the *createGUI* function as many styles as we want that will respond to properties assigned previously by using the *addProperty* function. In the first portion, a background color is searched within the item. If it is not present, a default color is set. Then, it is simply created an SVG rectangle with a D3 function that has as parameter the information previously calculated. In the last portion of code, an SVG text is added only if the *displayText* property is present. Since it is present only in the display model, the SVG text is added in the center of its SVG rectangle specifying some alignment styles. To add a behavior to a button that responds to a touch, we need to define an input entry point for our code. D3 offers the *selection.on(click, function)* method that allows to capture input clicks within the

SVG element specified by *selection*. However, our case study is a LMVD. We prefer the case where the touch event does not emulate a mouse click, but is simply an event containing the touch absolute coordinates, a unique touch id and the type of touch, such as in the SAGE2 environment. A touch event generated by the same user touch action can be of different type:

- *Touch down*: a new touch action occurs and a finger or object has just touched the screen;

- *Touch move*: a touch down occurred and the touch action continues with movement on the display surface;

- *Touch release*: the touch action ends and the finger or object leaves the screen.

For this reasons, we hypothesize that our application runs in a system dedicated to LVD, such as SAGE2, where inputs have a precise entry point. Our entry point will be a function with the following signature: *inputEvent(id, x, y, type)*. However, the techniques used here are compatible also with normal displays, but some operations, like the touched item recognition, might result superfluous since specific library functions are offered at this purpose, such as the *selection.on* D3 function. To let our interface works in whatever browser, we can simply call the *inputEvent* function whenever a click event occurs in our interface.

In our toy example, the *inputEvent* function check if whatever touch event is within a *clickable* item (we defined the *clickable* property previiously).

```
function inputEvent (id, x, y, type) {
   if (type != touchDown) {
      return;
   }
   for (var i in interface.model) {
      var item = interface.model[i];
      if (item.clickable && within(item, x, y)) {
         if (item.hiddenWord) {
            display.text.text(text);
            maximizeText(display, text)
         }
         break;
      }
   }
}
```

The *inputEvent* function iterates over all the models, discard those that are not *clickable* and then check if the click is within it. We can obtain this information with the support of the *within* function, as we can see from the code. This function check if the x and y coordinates are within the extremes of

the current item. Its code is shown below. We did not managed here the other touch types since we only need to press a button.

```
function within (element, x, y) {
   return (y >= element.y &&
   y <= (element.y + element.h) &&
   x >= element.x &&
   x <= (element.x + element.w));
}
```

Once identified that a button has been clicked, we can express its behavior in infinite ways. Probably the most elegant one is to attach to the item model the identifier of a function that executes the designed behavior and to invoke the function. To simplify the explanation, we directly perform the button operation within the context of the *inputEvent* function. Indeed, if the item has an hidden word, this one is shown in the display and then maximized with another function that we will not explain here. Now, our interface is ready and functional for a LVD. The result can be viewed in Figure 16. Although some parts of the explanation might result complicated, it is worth to mention that, once define the *createItems*, *addProperty*, *createGUI*, *inputEvent*, and *within* functions, we need a limited number of lines of code to create a simple, but effective interface. Indeed, grouping the needed lines of code to create our interface, we obtain the code below.

```
var container = d3.select("body").append("svg").attr("id", "theContainer");
container.style("left", 0).style("top", 0).style("background", "#D2BD8E")
   .attr("width", "100%").attr("height", "100%");
var interface = container.append("g").attr("id", "theInterface");
interface.model = [];
var display = createItems(1, 1, 0, 0, 50, 100, interface)[0];
var buttons = createItems(3, 3, 50, 0, 50, 100, interface);
addProperty(display, "displayText", true);
addProperty(buttons, "clickable", true);
addProperty(buttons[0], "bgColor", "green");
addProperty(buttons[1], "bgColor", "white");
addProperty(buttons[2], "bgColor", "red");
addProperty(buttons[0], "hiddenWord", "Good Morning!");
addProperty(buttons[1], "hiddenWord", "Good Evening!");
addProperty(buttons[2], "hiddenWord", "Good Night!");
```

## 4.2    Increasing the Expressiveness of the Large Display Interface

The technique demonstrated above has the capability to create simple SVG interfaces in limited time. For applications with few goals, this techniques is probably a good choice. For example, we

Figure 16: The interface of the first toy example. A button pressure invokes a certain action.

created with this technique a timer/countdown application in few lines of code that is used in both remote and collocated meeting to respect the time window every participant has for talking.

In this paragraph, we show how to create more flexible interfaces using more groups of items. We want also to demonstrate how adding properties to items can be effectively exploited to generate whatever kind of desired behavior. We show how it is possible to differentiate among different touches and how to manage the lifecycle of a touch action. At the end, we increase the number of interaction possibilities differentiating between a single and a double touch.

### 4.2.1   Toy Example 2: Increasing the Interface Flexibility

To avoid that the timer/countdown interface we created occupied all the LVD, we simply set the SVG container size to be smaller. In such a way, the other space of the LVD can be used to display documents, images and videos. However, here we are interested to create interfaces for more complex and sophisticate applications that are the center of the attention for the users. For this reason, we want that our application occupies effectively the whole display surface. We will realize another toy example, but without entering too much in the details. This time we realize an image visualizer that shows at the center of the page the images selected from a side menu. At the top of the page there is an independent banner welcoming the users. We identify three independent groups. With the same method used for the first example, we draw a sketch of the interface using a rectangle with similar proportion to a LVD. In Figure 17, we can see our sketch.
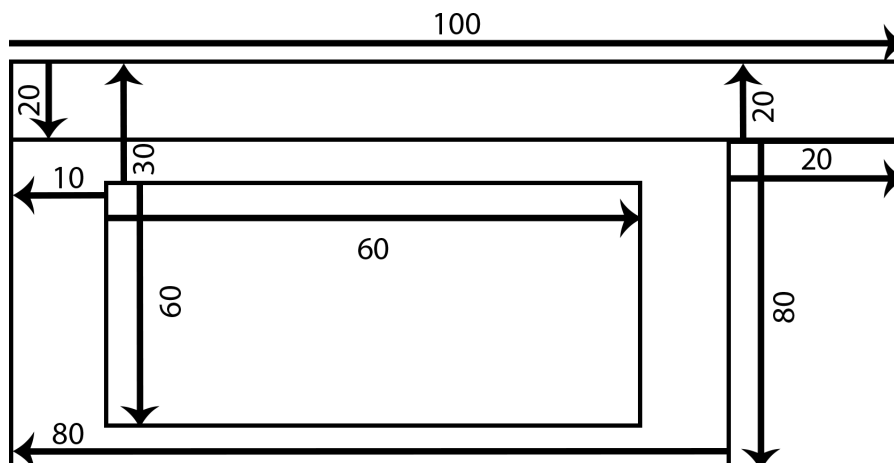
Figure 17: Toy example 2: how to derive the information for the different containers.

For each of these groups, we have to create an SVG group with the method used before, but adding the measures identified in Figure 17. Moreover, we want to add to the SVG container all the groups to reach their model in an easier way.

```
container.interfaces = [];
var sectionTop = container.append("g");
var sectionCenter = container.append("g");
var sectionRight = container.append("g");
sectionTop.measures = {x: 0, y: 0, w: 100, h: 20};
sectionCenter.measures = {x: 10, y: 30, w: 60, h: 60};
sectionRight.measures = {x: 80, y: 20, w: 20, h: 80};
container.interfaces.push(sectionTop, sectionCenter, sectionRight);
```

To position these groups in the right place, we have to slightly modify the *createGUI* function. We will serve of the *transform* function offered by all the SVG libraries. We will translates the groups in the desired position and scale it to the right dimension. The *createGUI* function now begins in the following way:

```
function createGUI (theGroup) {
  var svgWidth = theGroup.node().parentNode.clientWidth / 100;
  var svgHeigth = theGroup.node().parentNode.clientHeight / 100;
  var translateX = (svgWidth * theGroup.measures.x)
  var translateY = (svgHeigth * theGroup.measures.y)
  var scaleX = (theGroup.measures.w / 100)
  var scaleY = (theGroup.measures.h / 100)
  theGroup.attr("transform", "translate(" + translateX + ", " + translateY + ")");
  ...
  for (var i in theGroup.model) {
    var item = theGroup.model[i];
    item.x = item.c * svgWidth * scaleX + padding;
    item.y = item.r * svgHeigth * scaleY + padding;
```

```
        item.w = item.cSpan * svgWidth * scaleX - padding * 2;
        item.h = item.rSpan * svgHeigth * scaleY - padding * 2;
        ...
    }
}
```

In the first part, we calculate the translation and the scale factors. The translation factors are the left and top displacement in pixel calculated multiplying the percentage units of width and height and the desired displacement in percentage respectively. The scale factors allow to obtain the actual pixel size of the coordinates and the dimensions, and is calculated as the ratio between the desired width and height in percentage occupied by the group and 100, that is the maximum percentage of the two dimensions.

The translation factors are directly used within the transform function offered by D3. It simply translates horizontally and vertically the specified group. Because of this operation, the LCS of the group is translated too. Consequently the (0,0) coordinates represent the top-left of our translated group.

The x and y scale factors could be also used within the transform. Nevertheless, if the two scale are different, the ratio of the elements contained within the group is modified. We do not want to change the aspect ratio of our images, neither the one of the text. For this reason, we modify only the ratio of the x, y, w, h measures of the item. In such a way, the proportion of text, images, and similar contained within the group is preserved, while the proportion of the group measures is affected as desired.

At the end, we have to slightly modify also the within function. Indeed, the items models measure are expressed with respect the LCS of the group where they are contained. To restore the functioning of the *within* function, each time it checks if a touch is within an item, we have to bring the touch coordinates in the group LCS. This can be obtained adding to the *within* function the following lines of code:

```
var translate = d3.transform(element.container.attr("transform")).translate;
x -= translate[0];
y -= translate[1];
```

Last thing to modify is the *inputEvent* function. Indeed, we need to iterate over all the groups models, hence we need to perform two for loop:

```
...
for (var i in container.interfaces) {
```

```
for (var j in container.interfaces[i].model) {
var item = container.interfaces[i].model[j];
...
```

Now, we have all the resources to create a more flexible interface. As stated before, we do not want to enter in the details on how disposing the elements for this toy example. For completeness and to demonstrate the versatility of this method, we only show how to add an image in an element. We add to the buttons models, contained in the side menu, a property to express the fact that they contain an image. We suppose here to have an array *imgs* containing the link of the images.

```
var head = createItems(1, 1, 0, 0, 20, 100, sectionRight)[0];
var buttons = createItems(imgs.length, 3, 20, 0, 80, 100, sectionRight);
addProperty(head, "text", "Side Menu - Images")
addProperty(buttons, "clickable", true);
for (var i in buttons) {
   addProperty(buttons[i], "img", imgs[i]);
}
```

To interpret this property, we add to the respective part of the *createGUI* function the following lines of code that add an image to the buttons, if the img property is in a model item.

```
if (item.img) {
   theGroup.append("image").attr("x", item.x).attr("y", item.y).attr("width",
      item.w).attr("height", item.h).attr("xlink:href", item.img)
}
```

We do not explain how to display the image on the central section of the application since it is only a pure implementation detail and it is similar to the text display of the previous toy example. Instead, we prefer to show how adding properties to items is effectively a powerful way to create interesting behaviors of the items of our interface. Figure 18 shows the result of the second toy example's interface.

### 4.2.2    Toy Example 3: The Touch Action Workflow

We anticipated in the previous paragraph the possible states, or types, of a touch event and how it is transmitted to an application by the touch detection system. A touch event has attached a unique id, distinguishing it from touch events generated by others touch actions, a state, identifying if the event is a touch down, touch move or touch release, and the x, y absolute coordinates with respect to the display surface where the touch occurred. To enhance the expressiveness of an interface, we need to exploit all these information. First, we have to be able to distinguish among different touches. Then

Figure 18: The interface of the second toy example. A pressure on an image on the sidebar displays the image on the central container.

we need to perform different actions depending the type of the touch event. For example, we probably want to avoid that a button is activated with a touch move, but only with a touch down.

The easiest way to differentiate among the touch types is to define three different functions within the *inputEvent* function, completely redefining it:

```javascript
function inputEvent (id, x, y, type) {
   switch(type) {
      case "touchDown":
         touchDown(id, x, y);
         break;
      case "touchMove":
         touchMove(id, x, y);
         break;
      case "touchRelease":
         touchRelease(id, x, y);
         break;
   }
}
```

This simple way is convenient to maintain a certain order in the code and to distinguish the touch action lifecycle. Now we see how actually exploit the different touch states. To show it, we reuse the second toy example allowing the image visualization with a dragging operation. To show an image on the display, the user has to touch an image and drag it within the display area.

In order to exploit efficiently a user gesture on a LVD and to avoid confusion among touches of different users or different touches of the same one, we use a simple technique that makes use of the unique id provided by the touch detection system. First, we need to implement the *touchDown* function.

For each new touch event, we need to figure out if its position is valid. We do exactly what we did in the old *inputEvent* function iterating over all the known models.

```
function touchDown(id, x, y) {
   for (var i in container.interfaces) {
      for (var j in container.interfaces[i].model) {
         var item = container.interfaces[i].model[j];
         if (within(item, x, y)) {
            ...
         }
      }
   }
}
```

Once identified the validity of the touch and the touched item, we need to take track of the touch coordinates and the touched item associating this information to the unique touch id. We store this information in a dictionary declared at the beginning of our code, called *activeTouch*. We can infer the potential actions the touch is going to perform querying the touched item properties. In our case, we want to understand if the image can be dragged.

```
if (item.draggable) {
   activeTouch[id] = {};
   activeTouch[id].lastMousePosition = {x: x, y: y};
   activeTouch[id].item = item;
   break;
}
```

We did not added the *draggable* property to the image yet and actually the image has not even a model. There are many ways to achieve this goal. Once might be to add this information in the *createGUI* function. Previously we had buttons in the right section, now they are only passive element to host *draggable* images. We can add to them a *draggableImage* property. Then in the *createGUI* function, we reflect this property creating the image, its model and adding to the model some useful property.

```
if (item.draggableImage) {
   var img = theGroup.append("image").attr("x", item.x).attr("y", item.y).attr("width",
      item.w).attr("height", item.h).attr("xlink:href", item.draggableImg)
   var newItem = {x: item.x, y: item.y, w: item.w, h: item.h, img: img, container: theGroup};
   addProperty(newItem, "originalPosition", {x: item.x, y: item.y, w: item.w, h: item.h,
      container: theGroup});
   addProperty(newItem, "draggable", true);
   theGroup.model.push(newItem);
}
```

Now in our context, a touch action has started and we know it since this information is stored in the *activeTouch* dictionary. Hence, in the case an active touch performs a movement over the LVD surface, we want to reflect this behavior in our interface. This leads us to implement the *touchMove* function.

```
function mouseMove(id, x, y) {
   var f = activeTouch[id];
   if (f && f.item.draggable) {
      var newX = (x - f.lastMousePosition.x) + f.item.x;
      var newY = (y - f.lastMousePosition.y) + f.item.y;
      f.lastMousePosition = {x: x, y: y};
      moveImage(f.item, {x: newX, y: newY});
   }
}
```

Each time a mouse move occurs, we read from the *activeTouch* dictionary if there is a touch action with the same id. Then, we infer the corresponding action from the property assigned to the touched item. It is important to check the existence of an active touch with the same id because some touch actions, depending on the touched item, might not be stored in the *activeTouch* dictionary. A button is an example of it. Indeed, a button is generally activated with a touch down and the next touch states of the same action are meaningless. In the *touchDown* function we do not want to store its information in *activeTouch*, unless special behavior are assigned to a specific button.

Once retrieved the stored information bound to the touch id, we perform a certain action depending the property of the involved item. In this case, we want that the image follows the touch, so the user's finger. We calculate the new position as sum of the old one and the difference between the new position of the mouse and its last one. Then we update the last position of the mouse in the respective *activeTouch* object. The image movement task is given to another function that only update the model and the SVG image coordinates with the ones passed as parameter.

The image now follows the user's touch. Now, we want that the image is placed into the display, if the touch ends within it, otherwise it is placed in the previous position. Furthermore, if an image already occupies the display, it has to be placed into its original place. We implement here the *touchRelease* function, hypothesizing that the display model is contained in the display variable. If an image occupies the display, we associate the image to the display field *displaying*.

```
function mouseUp(id, x, y) {
   var f = activeTouch[id];
   if (f && f.item.draggable) {
```

```
      var foundPlace = within(display, f.lastMousePosition.x, f.lastMousePosition.y);
      if (foundPlace) {
         moveImage(f.item, display);
         if (display.displaying && display.displaying !== f.item) {
            moveImage(display.displaying, display.displaying.originalPosition);
         }
         display.displaying = f.item;
      } else {
         moveImage(f.item, f.item.originalPosition);
         if (display.displaying === f.item) {
            display.displaying = null;
         }
      }
   }
   delete activeTouch[id];
}
```

With the same technique used in the *touchMove* function, we retrieve the stored information bound to the corresponding touch id and infer the action to perform. We understand if the released occurred within the display using the *within* function. In such a case, the image is moved in the *display* position using the *moveImage* function. It is worth to mention that now, the *moveImage* changes the coordinates, the size and also the group of the image. This allows maintaining coherency with all the function defined previously. If the display is already occupied by another image, it is simply moved in its original position. Eventually, the new image is associated to the *displaying* field of display. In the case the image is dragged outside the display, it is placed in its original position and if it was in the display, the *displaying* field is nullified. Eventually, since the *touchRelease* determines the end of a touch lifecycle, its touch information stored in *activeTouch* are deleted. We can see the result of the third toy example on Figure 19, even though the image does not show the multi-touch capability of the interface.

In this paragraph, we demonstrated how to create flexible and adaptable interfaces for LVD exploiting the information offered by a touch detection system. We showed how it is possible to reflect different behaviors, depending the property assigned in the phase of the model generation and that the application model can be modified and expanded during the application lifecycle.

### 4.2.3   Increasing the Touch Interaction: Double Touch

There is still one thing we want to show. We want increase the way a user can interact with a LVD using a touch overlay. For example with a mouse, a user has at least the left and the right click, but a touch overlay is not able to distinguish two different fingers. Allowing the user to interact with
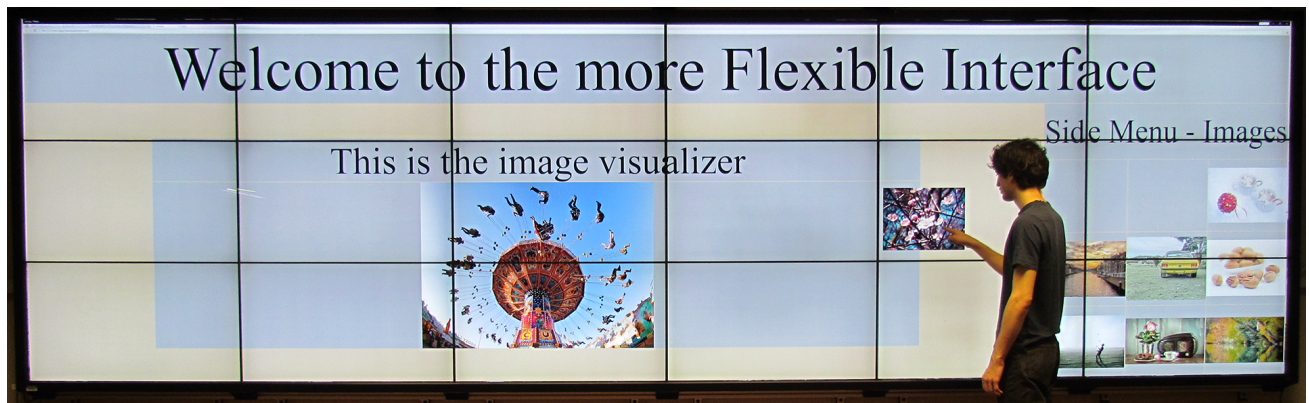
Figure 19: The interface of the third toy example. Now the interface supports also the drag function.

interface items in a unique way might be limiting. If the user would visualize image information in the previous example, how could he do? There is not a unique way. For example, an area dedicated to the information of the touched image might be a functioning idea. Another way might be increasing the interaction possibilities of a user. They could be a touch with two, three or more fingers, a long press, a double touch, and so on.

Also in this case we want to remember that there are many ways to implement such behavior. Ours is only a dedicated approach for LVD interfaces. We do not only want to detect a simple double touch. Our scenario is a multi-user environment and two subsequent touches are frequent. Instead, we want to detect if a certain item has been touched twice in a short time interval. To achieve this goal, we modify the *touchDown* function and implement a new one, the *doubleTouchDown*.

```
...
if (within(item, x, y)) {
   if (item.clickReceived) {
      doubleTouchDown(x, y, id, item);
      item.clickReceived = null;
      return;
   } else {
   item.clickReceived = true;
   setTimeout(deleteClick, 500, item);
}
}
...
```

Adding this portion of code allows to recognize if an item has been touched twice. The first click occurring on an item activates the *clickReceived* field and set a timeout that executes the *deleteClick* function after half of a second. This function simply nullifies the *clickReceived* field of the just clicked

item, if another touch does not occurs during the defined interval within the item area. Instead, if it occurs, this new touch invoke the *doubleTouchDown* function that will manage the initial phase of the double touch lifecycle. Indeed, to further increase the interaction possibilities it is possible to implement a *doubleTouchMove* and a *doubleTouchRelease* function that act in a similar way of their single touch counterpart. Last thing to do is to nullify the *clickedReceived* field and to interrupt the *touchDown* function with a return statement, to avoid the invocation of a not desired behavior.

There are other important considerations to do on adding new interaction possibilities. If an item is big, two user could interact with it at the same time and undesired double touch might happen. To avoid this problem, it is possible to invoke the *doubleTouchDown* function only if the two touches occur close enough. Moreover, the increase of the number of interaction possibilities, results in the increase of the interface expressiveness, but at the same time increases the detection errors possibility. Indeed, until now we acted as if the touch detection system would not have any reliability issue. Nevertheless, in real situations some detection errors can occur. We need to pay attention to these reliability problems, analyzing the situation and understanding if they affects too much the interaction with a particular item. In such a way, we should change the way to interact with that item.

## 4.3   The Conference Scheduler Application

This paragraph shows the application we realized for the user study done in this thesis. We exploited the same techniques exposed previously by implementing some useful and advanced behaviors that allow the users to exploit a high interaction level with the application interface. The user will be able to move items from one container to another, to increase their size, to change the position and the dimension of the containers, and to show additional information by activating some buttons. Nevertheless, we will not show implementation details regarding these new behaviors since they can be implemented in many ways. Indeed, we have already demonstrated the flexibility of the method exposed before, as it comes from the mechanism similarity to the MVC paradigm. Instead, we will show the application architecture to a high level by exposing the components and their interaction workflow.

### 4.3.1    Application Interface and Functions

The Conference Scheduler application offers all the functions that people generally perform during the scheduling of a conference using a whiteboard. In that scenario, users need to perform the following operations before starting the scheduling:

- Draw a table on the whiteboard where columns contain the available days, days contain columns for the available rooms, and rows contain the available slot hours. The table is similar to the one in Figure 12;

- Assign a color to each topic and write down on colored post-it notes the paper titles and authors.

Colored post-it notes will be used to reserve slots for the presentation of the paper written on it. In case a paper presentation needs a room with a capacity higher than the one with the highest capacity, it is possible to reserve multiple rooms that have the special capability to be collapsible. This can be done using copies of the post-it notes and occupying the desired collapsible rooms. For simplicity, collapsible rooms are placed one beside the other on the scheduling table. Visual cues are used to indicate which rooms are collapsible. Once the material is ready to be used, participants can start the scheduling task. We summarize in the following list the operations a participant should be able to perform:

- *Slot reservation*: the participant reserves a slot by occupying the corresponding cell with a colored post-it note;

- *Multiple slot reservation*: the participant reserves more than a slot by occupying multiple collapsible rooms with a post-it note and copies of it;

- *Change reserved slot*: the participant has the intention to change the slot that he has previously reserved by occupying another free slot. This might be the result of a negotiation with another participant or an error due to the non-compliance of some constraints;

- *Get room information*: the participant desires to know more information about the available rooms, such as their capacity, the available instruments and their characteristics, and the distance between the other rooms;

- *Get location floorplan information*: the participant desires to view the location floorplan.

These operations are all that is needed to perform the scheduling of a conference. The Conference Scheduler offers the possibility to perform all these operations. In addition, the applications tries to facilitate this complex task and to limit the constraints posed by the virtual 2D environment offered by this technology. On the one hand, the whiteboard approach has some problems and limitations, such as a substantial initial delay for the setup, the post-it note stick persistency problem, and so on. On the other hand, it has some advantages such as the possibility to move post-it notes in the 3D space of the room. There are several advantages on using a LVD for a similar task – even though they do not imply any obvious facilitations of the task – but there are also drawbacks, such as the limitation that all the items belonging to its virtual environment are unavoidably bound into it. For this reason, a user study was necessary to understand the effectiveness of LVDs for similar tasks.

We illustrate now how the Conference Scheduler interface is composed and we will list how a user is able to perform the previous operation needed for this task.



Figure 20: The Conference Scheduler initial interface shows the scheduling table at the top left, the paper containers at the right, and the commands at the bottom.

In Figure 20, we can observe how the Conference Schedule interface is organized. On the left side the application generates a scheduling table similar to the one in Figure 12. The paper containers are generated on the right side. A small portion of space below the table is dedicated to two command buttons (Map and Information). Active areas are present on the top-left of the scheduling table and on the top of the paper containers. These areas are called handles and they enable additional operations. The base operations are offered to the user through single touches, double touches and dragging actions. This is how they are activated:

- *Slot reservation*: the user drags the desired colored post-it note from its container to the corresponding cell by occupying it. The post-it note will follow the user's finger;

- *Multiple slot reservation*: the user drags the desired colored post-it note from its container to the corresponding cells by occupying it. Then, he performs a double touch on it which enlarges its size by occupying the other rooms;

- *Change reserved slot*: the user drags the post-it note from the occupied cell to another free one;

- *Get room information*: the user touches the Information button;

- *Get location floorplan information*: the user touches the Map button.

We enable two additional operations to facilitate the task and to reduce the effect of the post-it notes, which are constrained into the LVD 2D virtual environment. In this way, the user can also:

- *Change table and containers position*: the user touches the handle of the container he wants to move and drags it to the desired position;

- *Change table and containers size*: the user performs a double touch on the handle of the container he wants to resize and modifies its dimension by dragging his finger in the corresponding direction. The dimensions are dynamically updated.

Another feature we added is the possibility to save the current session. This operation can be useful in two situations. The former is when it is necessary to temporarily stop the scheduling session. Saving the session and restoring it later might be useful in the case others need to use the LVD. The latter is because sometimes is useful to look at the scheduling of the previous year conference. The save function was not directly provided in the user study, but the application state is saved each time a user performs an action with a post-it note.

### 4.3.2    Application Initialization

Before showing the modules composing our application, we will explain how it is initialized, therefore what happens when the Conference Scheduler application is opened. We used the techniques exposed in the previous paragraphs to develop our application. It is possible to identify two phases for the applications initialization analyzing the toy examples: the model creation and the GUI generation.
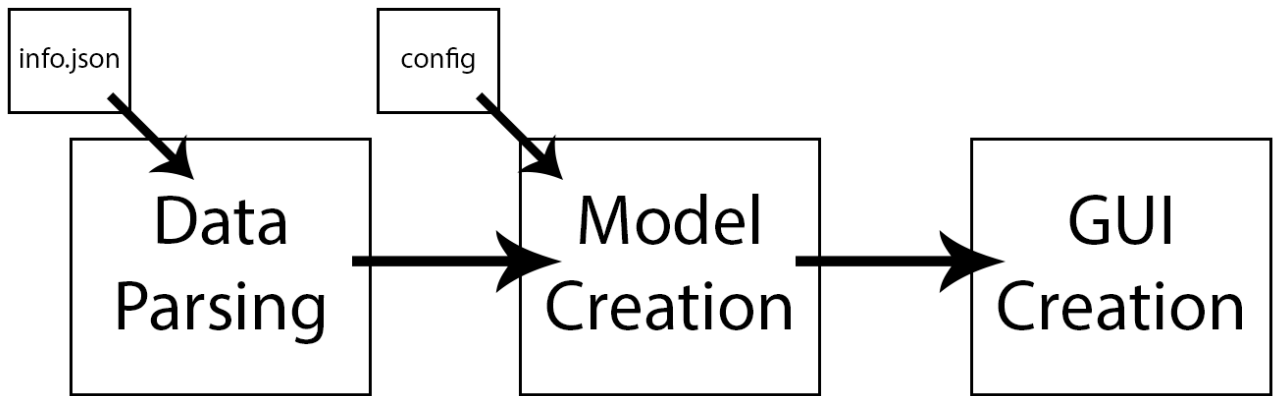
Figure 21: The application initialization is composed of three steps: data parsing, model creation and GUI creation.

This also occurs in the initialization of the Conference Scheduler. Actually, the data parsing phase comes before these two, as we can observe in Figure 21.

In the data parsing phase, a .json information file is parsed. The information related to the slot hours, the rooms, the days, the topics, the papers, and the association between paper and topic are extracted from the .json file. This information is memorized and passed to the next phase for the model creation. This phase is characterized by the use of multiple invocations of the *createModel* function. The application is able to generate the right amount of items models with the information read from the data parsing. The initial items and container dimensions can be specified in a different configuration file (Figure 21 shows it). By default, they are maximized within the available space. However, it is not really important the initial dimension of the containers since users have the possibility to move and resize them within the LVD surface.

The last initialization phase is the GUI creation. Once the model is created, the *createGUI* function is called and the SVG graphical elements are consequently created following the measures and the properties generated by the model creation phase. Eventual graphical preferences are read from the optional configuration file.

### 4.3.3   <u>Application Model</u>

We want now to go through the model creation phase showing the models and the properties assigned to them in the Conference Scheduler application, in order to have a clear overview of it.

The first section we will analyze is the scheduling table, which is composed by the handle, a line for the days, a line for the rooms, a column for the hours, and a subtable for the slots. We show the property assigned to each of these items in the following list:

- *Handle*: it is used to change the size and the position of the container. We assign to it the property *grabbable*;

- *Days, rooms, hours*: they are passive elements that only show a textual information. Hence, we assign them a property *text* containing the text of the corresponding day, room, or hour. This property specifies also the color, the thickness, and other textual styles later interpreted by the *createGUI* function.

- *Cells*: they are active elements since they are used to indicate a conference slot and to host the post-it notes. We add to them the following properties:

  - *isEmtpy*: if true it indicates that the slot is currently empty. All cells are empty at the beginning of the scheduling session, except for the ones that host plenary sessions;

  - *type*: it is set to *cell* and it is used to distinguish between cells and post-it notes. This is necessary since during the scheduling session this two items will be overlapped and the touch functions need to distinguish between them;

  - *point*: it expresses the value of a certain slot. This characterizes the higher value of some slots than others. This piece of information is used to calculate the quality of the schedule.

The command button section has a fixed position and three properties. We assign the three following property to the buttons:

- *text*: it is the same property assigned to days, rooms, and hours;

- *button*: it specifies that the item is a button and consequently *clickable*;

- *buttonOpen*: it indicates if the information bound to the button is currently shown;

- *image*: it contains the path to the image to show.

The paper containers are created depending on the parameter received from the parsing phase and they are independent from on another. Nevertheless, their models are created all together. A header,

a body to contain the post-it notes, and the post-it notes compose each paper container. Below are present the properties assigned to each of them:

- *Header*: it contains the topic name and it also acts as handle to move and resize the group. Hence, we assign the properties *text* and*grabbable* to it.

- *Post-it notes*: they are the items that allow the performance of more interactions. Consequently, more properties are needed to reflect their behavior. We summarize the properties in the following list:

  - *draggable*: it indicates that the item can be dragged and enlarged on the display interface;
  - *color*: it indicates the post-it note color depending on the topic to which it belongs;
  - *id*: it is a unique id that allows to one paper from another;
  - *points*: it sets the value of the corresponding paper. This is used to characterize the paper value;
  - *author, title*: they are set respectively with the corresponding authors and title of the paper.

In general, all the items are assigned to a *color* property to specify their background color following a neutral scheme. We will not talk about how the GUI has been realized and how the *createGUI* has been implemented since it is only an implementation detail of JavaScript and of the used SVG library.

### 4.3.4    Application Workflow

We will illustrate in this paragraph the application components and how they interact with each other. The touch of a user initiates a sequence of component interactions. For completeness we need to explain how the touch input information ies provided to the application. The Conference Scheduler runs within the SAGE2 environment. This provides the capability to use the application in a client server architecture and, by consequence, remotely. Obviously, the remoteness capability has to be implemented in the application to be functional, but its implementation was facilitated by the SAGE2 system that easily allows to synchronize variables among different clients. Nevertheless, touch input data are not directly provided to the applications, but to the SAGE2 UI. Indeed, SAGE2 use them to provide a higher level interface as described in the work of Nishimoto [38]. To send touch data to our

application, we created a small mechanism on the server side that redirects input touch data to our application, if it is running. With the help of Figure 22, we analyze the input touch workflow.
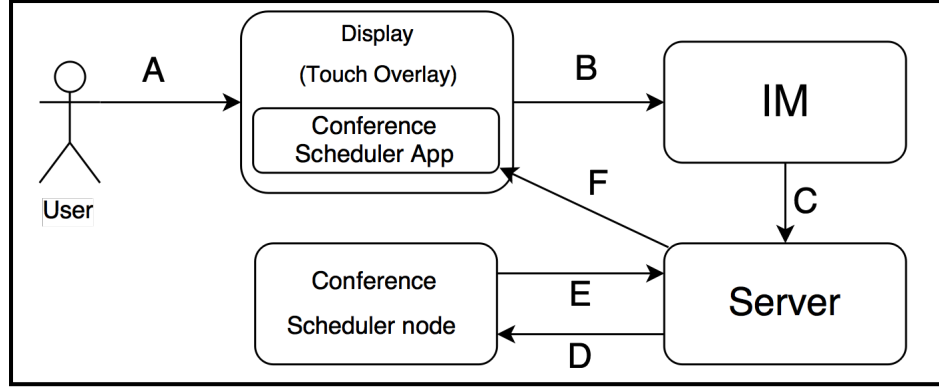


Figure 22: The Conference Scheduler application touch input workflow within the context of SAGE2.

We hypothesize that the Conference Scheduler application is running. When this occurs, the server does not send touch data to the SAGE2 UI. The touch overlay and the LVD are located in the same place, but connected to two different machines. The touch overlay is connected to a machine (IM) independent from the server. When the user touches the screen and interrupts the infrared barrier of the touch overlay (A) the information is sent to the IM that transforms the raw data to the format needed by the SAGE2 (B). Once the IM generates the elaborated data, it sends these data to the SAGE2 server and more precisely to the omicron component (C). Here the information is further elaborated and since omicron knows that the Conference Scheduler is running, it sends the new information to the Conference Scheduler node (D). This node simply prepares the data for the Conference Scheduler application in a more appropriate format. Once the data are ready for the application, they are passed to the server (E) that will redirect the information to all the clients connected to the current SAGE2 session (F), so that all of them can update the GUI.

Now that we know how the input touch data arrives to the Conference Scheduler application, we will neglect these details in the following explanation since they do not add any additional information. The application can be decomposed in the component presented by Figure 23 to a high level.

The application changes and evolves when the user touches the LVD in order to perform an action after the user acquires (sees) the information showed by the GUI. By consequence, the touch input
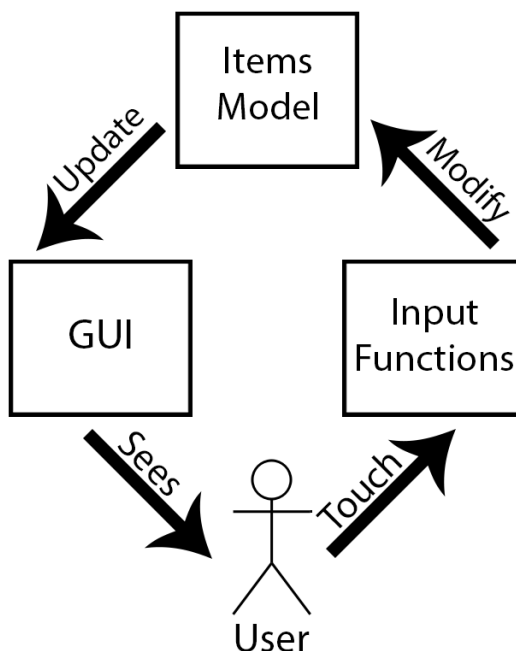
Figure 23: A high level representation of the Conference Scheduler application.

functions are invoked. They generally modify the model and these modifications are reflected as an update of the SVG elements contained into the GUI. It is possible to notice a certain adherence of this model with the famous MVC pattern. The touch input functions are the controller, the GUI is the view and the model is trivially the model. We will see in depth two operations the user can perform, and how these operations modify the model and update the GUI.

### 4.3.4.1 Button Touch

The easiest scenario we can represent is the activation of one of the two available buttons. In this situation only the *touchDown* function input acts while the others are neglected since a button does not answer to a *touch move* or to a *touch release*. When the user touches the display, the input data are sent to the input entry point function. This function recognizes the *touch down* type of the touch and invokes the *touchDown* function that iterates over all the items models to figure out which item has been involved. Since it is a button, its model is updated by setting the property *buttonOpen* to true. Then, the GUI is updated generating a new SVG element containing the map or the information. This one is a temporary item that is not added to the model and that is destroyed once the same button is pressed again. Figure 24 shows this workflow in a sequence diagram.
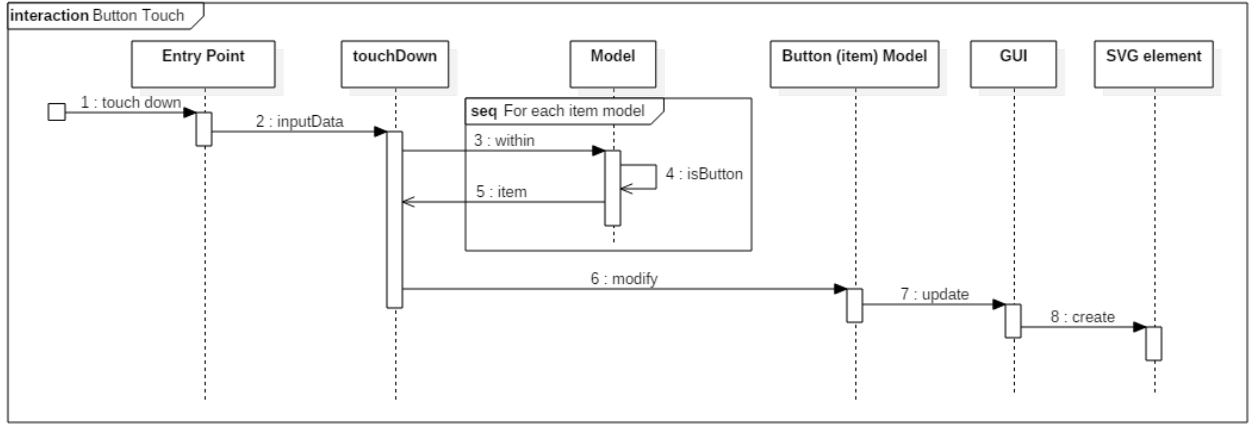
Figure 24: Sequence diagram for the button touch interaction. The only *touchDown* function is invlved in this operation.

### 4.3.4.2   <u>Post-it Notes Drag</u>

The scenario is more complicated. We want to represent the situation in which a user touches a post-it note and drags it to a cell. All the single touch input functions are now involved. The initial part of this use case is the same of the previous one. We start the explanation from the point in which the item model is retrieved and returned to the *touchDown* function. This function increases the number of clicks received by the item. Consequently, if another *touch down* is received in a short time interval, the double touch workflow is started while the single touch workflow is stopped. Then the *activeTouch* array is populated with an object containing the same information used in the last toy example.

When the user wants to drag a note and consequently performs a touch move, the input entry point calls the *touchMove* function. This one retrieves the object associated to the *touch id* within the *activeTouch* array and calculates the new position of the element, as did in the last toy example. The item model is modified with the new position and the GUI is respectively updated. This is done for each *touch move* that is generated by the same touch action.

When the user releases the touch, a *touch release* is generated and the input entry point calls the *touchRelease* function. The object associated to the same touch action is retrieved using the *touch id* on the *activeTouch* array. Using the *within* function and iterating over all the items models, it is retrieved the cell within the *touch release* occurred. The new size and position of the post-it note model
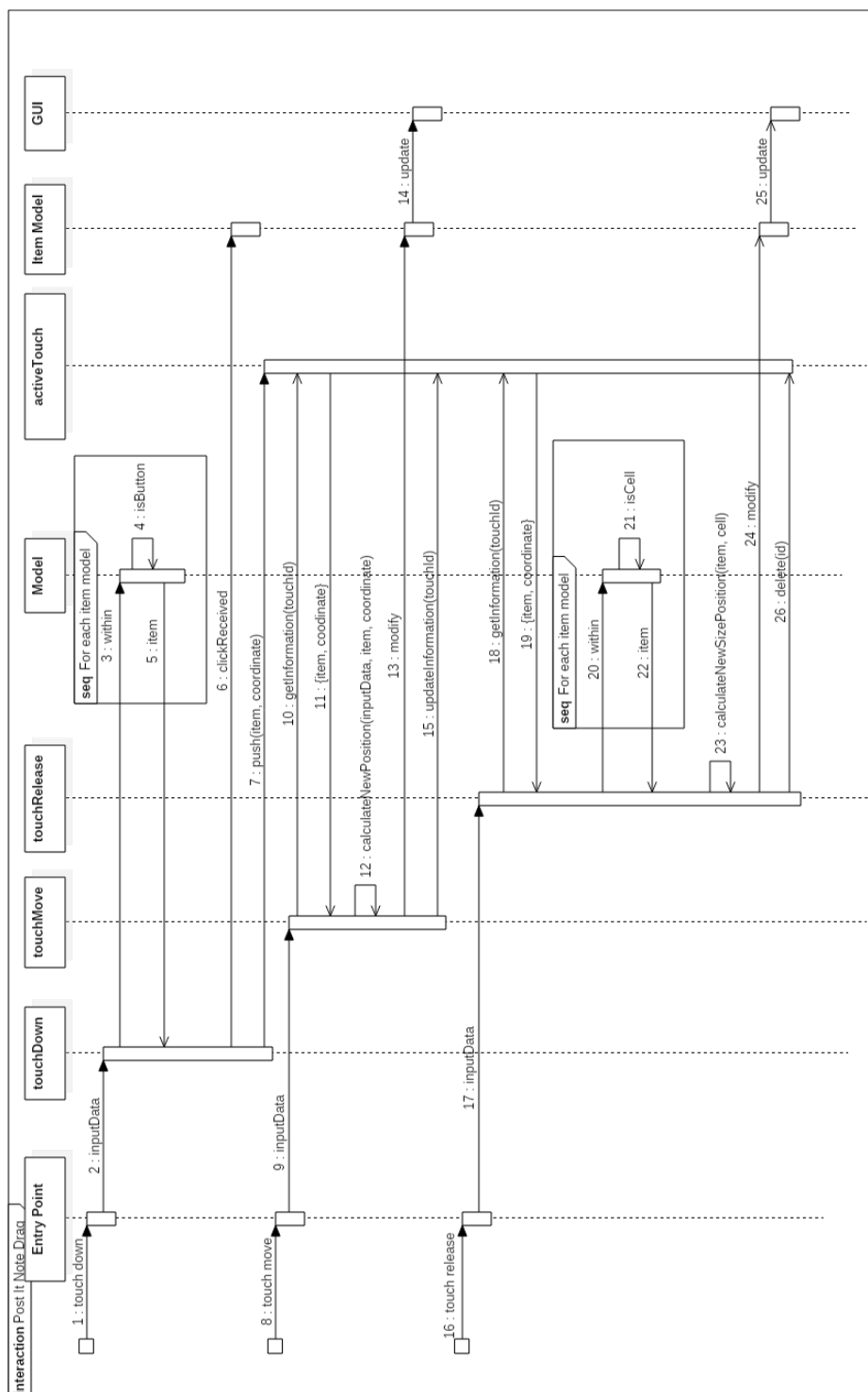
Figure 25: Sequence diagram for the post-it note drag interaction. All the touch input functions are involved in this operation.

is calculated as did in the third toy example. The item model is modified and the GUI updated. The *activeTouch* array is freed by the object associated to the *touch id* of the touch action just terminated.

# CHAPTER 5

## USER STUDY

In the previous chapters, we showed how many studies have demonstrated how the usage of LVDs produces several benefits in different kind of tasks, from navigation of a 3D environment [42], [43], sense-making [48], to complicated tasks requiring high cognitive load [23]. LVDs usage is not trivial and their effectiveness is not a foregone conclusion since there are no specific paradigms for producing applications dedicated to these displays [10]. Nevertheless, other studies have demonstrated how LVDs usage in a multi-user collaborative environment can increase productivity and the quality of the work [24], [2], [23], [4]. However, research on LVDs has been conducted only in multi-user environments where people have a unique common goal. Beyond these studies, certain tasks require working with other individuals to reach a final goal, while at the same time, every individual tries to maximize his/her own objective. We define these scenarios as collaborative with a competitive component. For simplicity, we will refer at this scenario as competitive, even though we do not want to neglect the presence of collaborative component. An example of this scenario is the one presented in [49], where a user study is conducted on a collaborative task with a negotiation component, as we showed in the related works chapter. In our user study, we want to test the effectiveness of LVDs with multi-touch capabilities to assist people in a multi-user competitive environment and the users' behavior.

We will study the general problem of using a LMVD in a multi-user competitive environment with a specific task: paper presentation scheduling for a scientific conference. In this specific problem, there are common constraints that prevent the event scheduling at certain hours, and personal constraints for every participant. The presence of these constraints limit the configurations, since an admissible configuration has to respect the needs of everyone. It is difficult to reach an admissible configuration because the participants are not typically interested in the other participants' constraints. This aspect makes conference scheduling a competitive task, since every participant tries to maximize his/her objective in reaching an admissible configuration. However, we do not have to forget that the main goal of the task is to produce a good scheduling and that a good scheduling constitute the second half of the users' goal. Maximizing only the personal goal generally lead to neglect the shared goal and

consequently to a bad scheduling. We thoroughly discussed about this problem at the end of Chapter 2. We consider this problem as a perfect candidate to discover the effects that LMVDs can have in a multi-user competitive environment. The LMVD also allows us to mimic the traditional mode of doing this scheduling with post-it notes.

The developed application, the Conference Scheduler, runs in the context of SAGE2 [6]. We will use the Cyber-Commons wall display at the Electronic Visualization Laboratory at University of Illinois at Chicago that is equipped with an infrared multi-touch overlay. The large size of the display will allow multiple users to interact with the application simultaneously. This equipment enables our study to analyze if a LMVD can aid people in dealing with a multi-user competitive task compared to the traditional way of dealing with the problem, and how group of people interact with the LMVD in this particular situation.

## 5.1     Apparatus and Used Material

The LMVD in the Cyber-Commons room at EVL is a wall display composed of 18 panel tiled display with a total size of 21.9 by 6.6 foot. The display are disposed to compose a 6 wide by 3 high display matrix. Each display has a resolution of 1360 by 780 pixel for a total resolution of 8160 by 2340 pixel. The touch overlay is mounted around this display and connected to a different machine with the unique task of receiving, elaborating, preparing and sending touch input data to the machine where the SAGE2 server runs. The Cyber-Commons room is slightly larger than the display and this does not give opportunities on staying at the side of the LVD. Since a study run with each approach generally lasts approximately 30 minutes, some chairs and tables where provided in the room to let participants sit and relax for some instants. The tables were disposed horizontally to the display roughly 1 meter far from the display, while the chairs where behind these tables. However, no user sat on chairs, but ever preferred to sit on the tables. This was probably caused by the fact that the table forced the user to take a longer path to reach the chairs.

Since we want to do a comparison between the traditional approach and the use of a LMVD, we decided to normalize as much as possible the scenario. We did not employed an actual whiteboard, but we used a snapshot of the same scheduling table produced by the Conference Scheduler with appropriate dimensions. Participants will stick the post it notes directly on the large screen. This is possible since

the large screen is covered by a Plexiglas surface. Colored post-it notes were provided to conduct the test.

Participants did not interact with any computer, but only with the LMVD using their fingers and sticking the post-it notes on the wall display.

## 5.2 Participants

In the user study participated 20 people, 1 female and 19 male. We conducted 5 study runs with a group of 4 people each. Some of the participants applied to the user study already in small groups. Since we did not impose any constraint on the knowledge among participants, we did not divide them. Indeed, in a scheduling session chairs participants might both knew or no the others. To test the effectiveness of the employed device and application, each group performed the scheduling with two different datasets both performing the traditional and the display approach. All groups completed the scheduling session in the given time. No reimbursement or compensation are given to participants.

## 5.3 Task

The groups of user involved in the user study have to perform a scheduling session per each approach. We already explained how a scheduling session works, in the background chapter. However, we slightly simplify it in order to do not confuse too much the user since our goal is to understand how users behave using a LMVD; not only comparing the two approaches.

Each user impersonates a chair for a topic in a scientific conference. The topics have mock names represented by colors. The same happens for the papers titles. Each paper has a name composed of two character. The first one is the first letter of the topic color to which it belongs, while the second is an incremental number starting from one.

The constraints sheet has the following information presented in a table:

- *Paper*: the name of the paper. It has the form of $XY$, where $X$ is the color of the own topic and $Y$ is an incremental number starting from one.

- *Is after*: it is precedence constraints for the paper in the *Paper* column in the same row. The papers in this cell have to be in a slot hour before the constrained paper.

- *Room*: it is a space constraints for the paper in the *Paper* column in the same row. This paper has to occupy two rooms.

TABLE I: CONSTRAINTS TABLE SAMPLE CONTAINED IN THE MATERIAL GIVEN TO THE PARTICIPANTS

| Paper | Is after | Room | Days | Slot Hours | Points |
|:---:|:---:|:---:|:---:|:---:|:---:|
| **R1** | | | Thu, Fri | | 1 |
| **R2** | R1 | Double | Thu, Fri | | 2 |
| **R3** | | | Fri, Sat | | 1 |
| **R4** | R3 | | Fri, Sat | Second, Third | 2 |
| **R5** | | | Fri, Sat, Sun | Second, Third | 1 |
| **R6** | R5 | | Fri, Sat, Sun | | 2 |
| **R7** | R5 | | Fri, Sat, Sun | | 3 |
| **R8** | | Double | | | 1 |
| **R9** | | | | | 1 |

- *Days*: it is a temporal constraints for the paper in the *Paper* column in the same row. This paper must be inserted in slots belonging to the days contained within this cell.

- *Hours*: it is a temporal constraints for the paper in the *Paper* column in the same row. This paper must be inserted in slots belonging to the hours contained within this cell.

- *Points*: the value of the presentation for the paper in the *Paper* column in the same row.

In Table I is presented a sample of table contained in the material given to the participants.

The conference used for this user study has four days, four rooms and four slot hours. However, not all the slots are available for the topics presentation. This is done to simulate the existence of plenary sessions and events that occupy some rooms. The unavailable slots are the ones in the first day on the firsts two slot hours and the ones in the last day on the lasts two slot hours. In such a way, participants will concentrate their attention on the central days that are actually the more researched by chairs. By consequence, there are forty-eight available slots. Furthermore, in this conference only two rooms are collapsible. These are the last twos on the table and this knowledge is shown inserting a red rectangle around their name. Figure 26 shows the exact table that the Conference Scheduler application presents to the participants.

To make desirable the slots in the central hours and in the central days, a value is given also to every available slot. This works like a multiplier for the value of the paper inserted within it. For example, if a paper with value of 2 is inserted in a cell with value of 3, the paper's owner obtains 6 points. Collapsing a room allows participant to obtain the sum of the multiplier for the inserted paper.

Figure 26: This is the exact table that the Conference Scheduler shows in our user study. There are four days, four rooms and four slot hours. The black cell are the unavailable ones while the light brown cell are available. The collapsible rooms has a red rectangle around their name.

This information is contained in the same material containing the constraints of each user. Figure 27 shows the image presented to the participants. It is actually a snapshot of the table presented by the application, but with the addition of the values for each slots.



Figure 27: The numbers within the slots represent the value of the slot. They work as a multiplier for the paper occupying the corresponding cell. This image is the same given to each participant in the same paper sheet containing the constrains.

The only difference between the two approaches are the post it notes. Indeed, in the traditional approach participants are given another paper sheet with all the needed post it notes. For presentation requiring two rooms, two post it notes with the same title are provided. A sample of this material is shown in Figure 28. In the large display approach, the post it notes are emulated by virtual items in the display and are contained in specific containers, one for each participant, as show in Figure 29.

## 5.4    Procedure

In a study run, a group composed of 4 people goes through the task of scheduling a conference with both the traditional and the proposed approach while an observation study occurs. In a first phase,
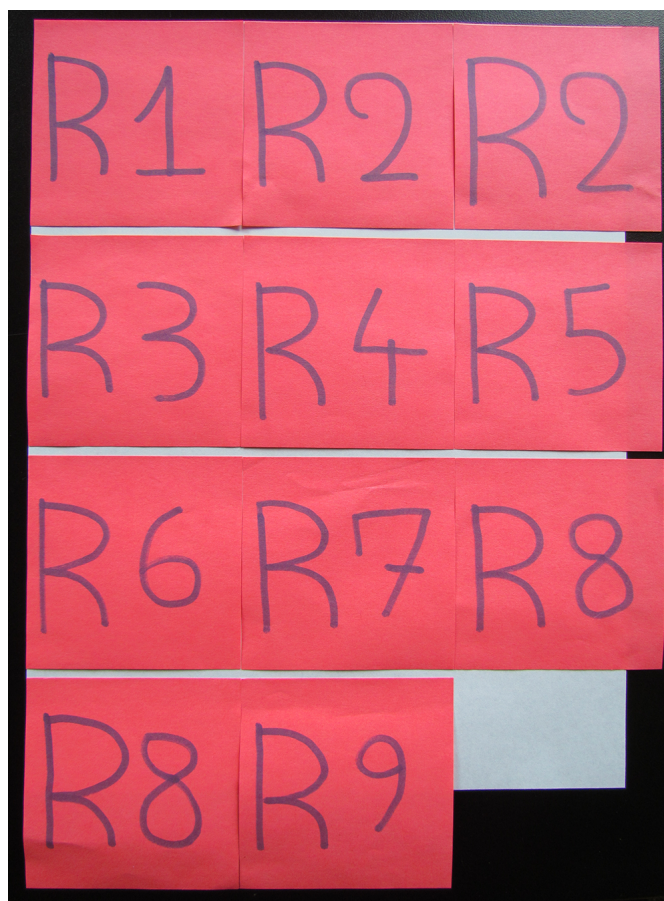
Figure 28: For the traditional approach, each participant receives a paper sheet similar to this one containing one or two post it note of his color, for each topic he has to schedule.



Figure 29: These are the papers containers presented by the Conference Scheduler application, one per each participant. The image shows that they can be resized as desired. At the begging, the application presents all of them at the same size.

the group members take turns in adjusting the schedule, followed by a group negotiation phase, as explained in the background chapter. The first employed approach will be alternated in each run to normalize the effect of the problem learning on the second one. Since some aspects of the problem might be not completely understood, a facilitator represented by the PI is present in the room during the study and will be able to answer whatever questions are raised up regarding the Conference Scheduler. The exact steps of the user study are the followings:

1. The PI will explain the purpose of the study to the subject, describe the procedures, inform them of their rights, answer any questions that the subjects might have, and ask for their informed consent.

2. The PI will explain how to use the materials that will be given to them. The material consists of:

   - A sheet per each participant containing the constraints they have to meet. Each participant has unique personal constraints;
   - A sheet per each participant containing the post-it notes of a color assigned to the participant. The post-it notes contain the mock names of the respective papers.

3. The PI will explain how a conference scheduling session is done using a whiteboard.

4. The subjects will be given a 15 minutes training using the mock whiteboard and real post-it notes.

5. A set of material is given to the participants and the session using the mock whiteboard starts.

6. A 15 minutes break is given to participants.

7. The PI will explain how the Conference Scheduler tool works to perform a conference scheduling session.

8. The subjects will be given a 15 minutes training using the Conference Scheduler tool.

9. A new and different set of material is given to the participants and the session using the Conference Scheduler tool starts.

10. At the end of the study, the participants are asked to fill out a questionnaire related to the tasks they performed. This information will be used to understand which approach has been preferred by participant, and to get suggestions on how to improve the Conference Scheduler application.

Steps from 3 to 5 and 7 to 9 will be alternated in each run to normalize the effect of the problem learning on the second employed approach.

## 5.5  Analysis and Hypotheses

Computer tools for complicated tasks have been developed to aid people in solving specific problems. Many of them have succeeded and completely replaced the traditional way of dealing with the problem. Other tools, belonging to the CSCW (Computer Supported Cooperative Work) research field, have been demonstrated to be very effective in solving collaborative tasks. LMVD are an instrument becoming more widely used to aid teamwork. We want to understand the effectiveness level of this technology, with the addition of a multi-touch overlay, when people operate together to reach a certain goal, but at the same time want to maximize a personal objective. Our intent is that the information obtained in this study could be used in future work on the design of applications in such environments.

We want to understand if the information displayed by a LVD and the multi-touch interaction mode aids people in solving a task in a multi-user competitive environment, where each user tries to maximize his objective. We think that this goal can be reached thanks to the facts that the quality of the displayed information is higher on a LVD. In fact, the information shown on the wall display is larger, brighter, better defined and more ordered. Nevertheless, there are also negatives in using the Conference Scheduler. Although the touch capabilities are enough reliable, the movement of a virtual post-it notes does not have the same flexibility as a real one. We are interested in understanding:

- If the employed technology makes a multi-user competitive task easier compared to its traditional approach;

- If the proposed application allows participants to have a better understanding of the information contained in the scheduling table;

- If the proposed application is missing critical features or is counterproductive compared to the traditional approach.

and more generals such as:

- How groups of people interact with a LMVD and one with the others, in terms of time consumed on interacting with the display, verbal communication, visual attention, and display proximity;

- If some people assume a leadership position within the group and others stay more passive.

Our hypothesis are that in general a LMVD facilitate a multi-user competitive task. Indeed, we expect that people prefer using a LMVD for the clearness and the better ordering of the information, and for the easiness of performing some operation. This should lead to slightly better scheduling, even though the little improvement of this result will not probably be statistically significant. We expect that it is more frequent that using LMVD, there will be a person in the group that will assume the leadership. However, we think that the inherent difficulty of the task will not lead to a consistent decrease of the consumed time. Despite of this, we suppose that negotiation will be easier and frustration lower. In general, our main hypothesis are that participants prefer a LMVD and that this technology can make such task less frustrating. Furthermore, since the conference scheduling task is divided in two different phases, we expect that participants will behave differently depending the phase. We suppose that in the first phase, participants will work more independently thinking mainly to their strategies and on the constraints satisfaction. Meanwhile, participants will talk more in the second phase trying to negotiate, but also to collaborate by finding an agreement for the task conclusion. Hence, we suppose that in the second phase both the competitive and the collaborative variable will live together, while the turn based nature of the first phase will not leave room for collaboration.

## 5.6    Data Gathering

Our experiment had two purposes. First, we wanted to understand if a LMVD is an effective instrument to support the execution of a collaborative/competitive task in multi-user environment. Second, we studied how people behave in such environment. To reach these purposes, we needed to capture three different type of data.

The first type consists in record and log data automatically with background procedure in the Conference Scheduler application. With this system, we were able to record the points evolution and the users' interaction with the display while placing post-it notes. Regarding the whiteboard approach, we employed a person that replicated all the users' post-it notes movement in order to visualize the data evolution over the time. This data were double-checked with the video recordings.

Video and audio recording are the second type of data we recorded and analyzed to capture the needed information. We used one *GoPro HERO3+* camera in the middle of the room video recording

the whole LVD and another Full HD video camera on the side of the LVD video recording the study horizontally. We obtained 4.6 hours of useful recordings, that are the 81% of the actual recorded data. These recordings do not include the time used to instruct the users, to fill the questionnaires and obviously the break moments. We used these data to extract the users' position, the shape assumed by the groups, and their visual attention. To record audio, we used a digital recorder (*Zoom H2 Handy Recorder*). The audio recordings have obviously the same duration of the video ones. We analyzed audio recordings to extract information regarding the verbal communication and frustration signals. Video and audio recordings were used also to confirm and validate the results obtained by the other type of data.

Last type of collected data are the questionnaires that resulted useful to understand the users' thought and to confirm some deduction we did from other data. We took also some video frame to show the difference on how users decided to organize the containers within the LVD.

### 5.6.1    Application Logs

The application recorded three types of logs. The first one is related to the evolution of the points of each user during the task. Each time the user moved a post-it notes from a position to a valid one, an event is record. For this event, the application saves the owner of the moved post-it note, the time in which this event occurred in milliseconds and the point obtained by this action. In such a way, we were able to reconstruct the point evolution of the users and consequently the point evolution of the scheduling. Second type of data log describes the points of interaction of the user with the display. For each touch on a post it notes, the application saves the owner of the post-it note, the coordinate and the type of the touch.

### 5.6.2    Video and Audio Recordings

As did by Jakobsen and Hornbæk [24], we preferred to code in different ways the visual attention and verbal communication. Other studies, such as the ones of Isenberg et al. [53] and Tang et al. [54], take in consideration a unique coding combining the visual attention and verbal communication. In such a way, we were able to extract also characteristics that can be defined more as physical base codes, rather than only socially based codes. Indeed, the first one are easier to extract, even though more technical, while the others require inference and abstraction from the situation.

DISPLAY          DOCUMENT          MIXED          EACH OTHER          DISENGAGED
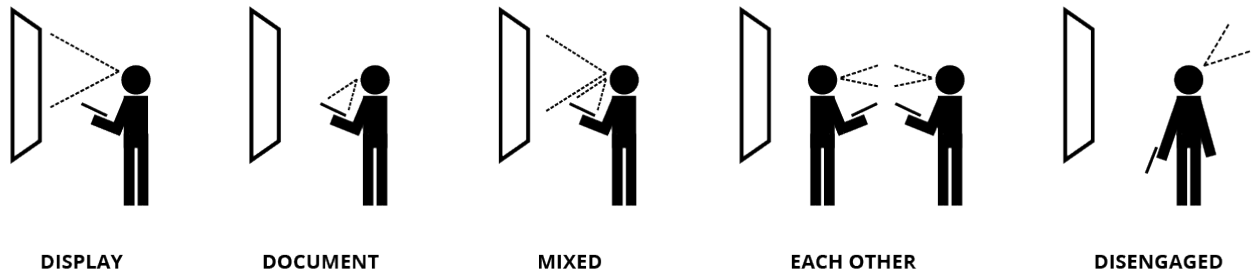
Figure 30: The set of code for the group visual attention is composed of five states: display, document, mixed, each other and disengaged. The groups enter in one of this state when most of the users perform the corresponding action for at least five seconds.

The coding that combines visual attention and verbal communication is slightly different from the three studies cited before. We preferred to simplify this categorization since the one used by the other studies are thought for pairs and generalizing them would imply a too large partitioning of this domain. Unfortunately, this simplification is more approximated since uses terms like 'most of the users' rather than 'user A does something, while user B does some other'. However, as the others coding, it remains still a mutually exclusive and exhaustive coding since we request that most of the participants do the respective action for at least 5 seconds. Figure 30 displays the set of codes for the group visual attention, while the list explains the categorization more in detail.

- Looking at display (*display*): all users, or mostly, are looking at the display.
  Example: The participants are checking the available slots while thinking to their next slot reservation.

- Looking at each other (*each other*): all users, or mostly, are looking at another user.
  Example: A negotiation is in act and a participant is asking others if he agrees to trade a slot for another.

- Looking at the document (*document*): all users, or mostly, are looking at their documents.
  Example: The participants are studying the constraints of their talks and are predominantly to their document.

- *Mixed*: most of the users are looking at something belonging to the previous three categories.
  Example: Two users are having a discussion, so they are looking at each other, while another is

checking the respects of his constraints. The last user is disengaged from the task or is looking at the others participants passively.

- *Disengaged*: most of the user are not looking at anything related to the scheduling or has a passive attitude to the current situation.

  Example: One user is trying to satisfy a violated constraint while the others are not interested to the situation for whatever reason.

We used the video recordings to analyze the users proximity to the display. We created four zones that are close, medium, and far from the display, distinguishing by central and lateral far. To measure more accurately the users proximity to the display, we slightly modified the video recordings adding virtual lines to delimit those zones. Figure 31 illustrates and explains these zones.
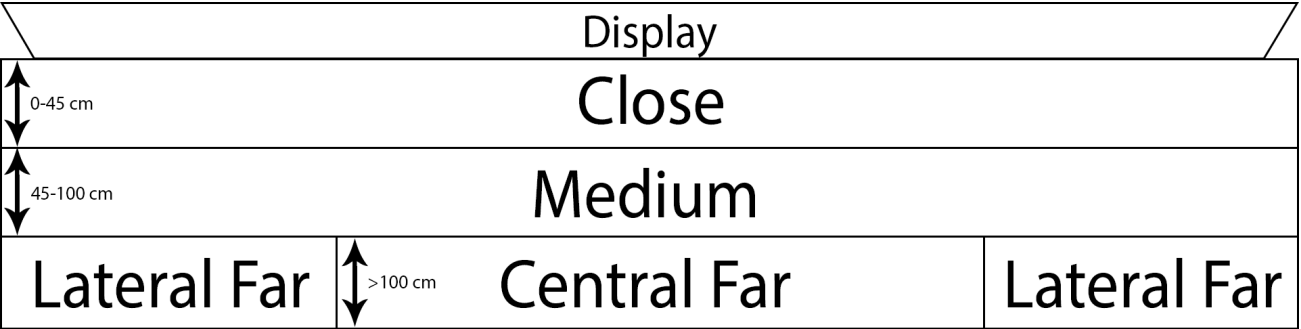


Figure 31: There are four display proximity codes: close, medium, central far and lateral far.

Furthermore, we found interesting to code also the shapes assumed by the groups. We identified four type of shapes and Figure 32 helps us to describe them.
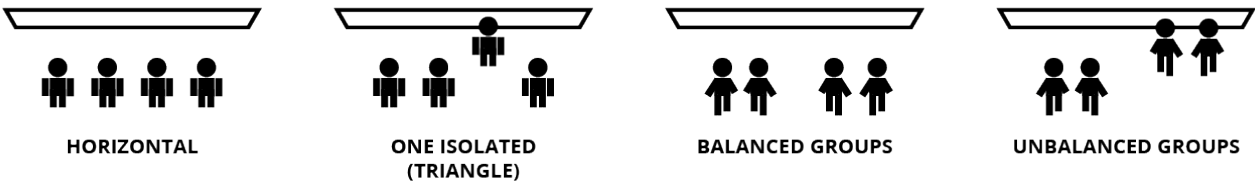


Figure 32: We coded the shape assumed by groups in a set of four codes: horizontal, one isolated (triangle), balanced and unbalanced groups.

- *Horizontal shape*: when the users are parallel aligned to the display;

- *Balanced groups*: when the groups are divided in two sub groups with a similar distance to the display. This generally happens when the two groups are both active and probably negotiating;

- *Unbalanced groups*: when the groups are divided in two sub groups, but with a different distance to the display. This generally happens when one group is active, while the other is inactive or doing something that does not require proximity to the display;

- *One isolated* or *Triangle*: when one user detaches from the group performing an action on the display and the others are inactive.

The categorization obtained using the audio recording are simpler and clearer, but still very significant. We use codes similar to the ones of Jakobsen and Hornbæk [24], but slightly revising them to obtain codes more adapt to the size of our scenario.
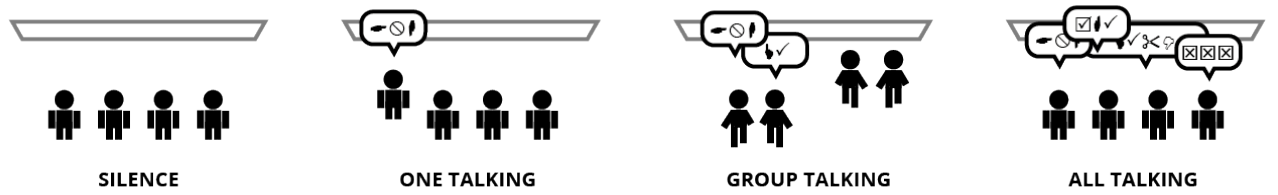


Figure 33: Verbal communication is divided in four different codes: silence, one talking, group talking, and all talking.

We show in Figure 33 the categories we coded for the verbal communication. Moreover, we describe the categories accurately in the list below.

- *Silence*: no participant is talking. Low sounds due to reasoning are admitted in this category. Example: The participants are in the first phase of the experiment waiting their turn, looking at their documents or the participant that is performing the action in the current turn.

- *One talking*: A participant only is talking while the others are silent or are making short or not significant comments, such as simple acknowledgment.
  Example: A participant is asking if it possible to make an exchange in the schedule, since he was not able to satisfy all his constraints during the first phase.

TABLE II: ALL THE SETS OF CODES WE USED IN OUR USER STUDY

| Code | Categories |
|---|---|
| **Visual Attention** | Display, Each other, Document, Mixed, Disengaged |
| **Display Distance** | Close, Medium, Central far, Lateral far |
| **Group Shape** | Horizontal, Balanced groups, Unbalance groups, One isolated |
| **Verbal Communication** | Silence, One talking, Group talking, All talking |

- *Group talking*: Most of the user are talking while at least one is silent or is making not significant comments, such as simple acknowledgment.

  Example: Two users are having a negotiation for a slot. The others are not interested to that slot and consequently are not participating to the conversation.

- *All talking*: All the users are engaged in a conversation. There might be a unique conversation involving all the users or multiple conversation in pairs.

  Example: A user is not able to satisfy all his constraints and ask if someone is able to trade a certain slot. All are interested to the trade and consequently participate to the conversation.

Each one of this sets of code has to last at least 5 seconds to be recognized as valid. In Table II, we synthesized the codes we used in our user study.

### 5.6.3    Questionnaires

Questionnaires helped us to understand if our application helped participants to tackle the scheduling of a conference – so, if the Conference Scheduler – is a well implemented application. Otherwise, our results would be consequently biased by the fact that the application had not negligible problems. We used a Likert scale for some of the questions, while for other questions the users commented in free text. We show now in some tables the questions contained in the questionnaire. We already shown the results when numeric value has to be inserted. However the questionnaire analysis will be done in the result and discussion paragraph.

The first group of questions (Table III) are related to the Conference Scheduler application. We were interested in understanding if the user liked the application, and what he thought about its interface. Regarding the interface we wanted to understand if the information visualized and the functions offered by the Conference Scheduler were enough complete.

Next group of questions (Table IV) aimed to make a comparison between the whiteboard approach and the Conference Scheduler. The questions asked to the user what approach he preferred and why, if one was preferred, what are the pros of the Conference Scheduler and if it has some lacks. Three questions that are more general concluded this section. We asked to the user what he thinks about the introduction of LMVDs for manual problems and why he has this thought. This last question is important to understand if negative answers on our implementation can be in part associated to some negative thought and feeling about technology.

We then posed two questions (Table V) about leadership within the group in which the user had performed the tasks. We compared this result with the video recordings to understand if one or more users tried to assume a leadership position within the group.

### TABLE III: QUESTIONS RELATED TO THE CONFERENCE SCHEDULER APPLICATION

| Question | M | SD |
|---|---|---|
| Please rate the **ease of use** of the Conference Scheduler on a scale of 1 to 10, with 1 being difficult and 10 being easy | 8.40 | 0.52 |
| Please rate the Conference Scheduler **interface** with respect to the **visualized information**, such as their clearness and completeness, on a scale of 1 to 10, with 1 being low informative and 10 being high informative | 8.95 | 0.46 |
| Please rate the Conference Scheduler **interface** with respect to the **offered functionalities** on a scale of 1 to 10, with 1 being low functionalities and 10 high functionalities | 8.65 | 0.71 |
| **Question** | | |
| Did you felt any lack or redundancy on the proposed **interface** with respect to the **visualized information**? If yes, list them in the space below | | |
| Did you felt any lack or redundancy on the proposed **interface** with respect to the **offered functionalities**? If yes, list them in the space below | | |

Finally, a free text question asked general comments on the Conference Scheduler application. We collected this data to improve the application for further study and to actually let users to schedule a conference with this instrument. The question was *Please write in the space below any further comments about the Conference Scheduler used during this study.*

## 5.7 Results and Discussion

In this section, we present the results obtained in our user study and contemporaneously discuss them. First, we make a comparison between the results of the two approaches. This is done initially

TABLE IV: QUESTIONS RELATED TO THE COMPARISON BETWEEN THE WHITEBOARD APPROACH AND THE CONFERENCE SCHEDULER APPLICATION

| Question | M | SD |
|---|---|---|
| Please rate the **level of improvement** offered by the Conference Scheduler with respect to the traditional way of tackling the problem on a scale of -5 to 5, with -5 being the Conference Scheduler is worse than the traditional way and 5 being Conference Scheduler is better than the traditional way | 2.81 | 0.50 |

| Question |
|---|
| Why did you prefer one approach to the other? |
| What pros did you find in the Conference Scheduler with respect to the traditional way of tackling the conference scheduling problem? |
| How could the Conference Scheduler be improved? Are there any features that you felt are missing? |
| Did you felt any lack or redundancy on the proposed interface with respect to the offered functionalities? If yes, list them in the space below |
| In general, do you think that technology, such as large displays with multi-touch, can help to tackle manual problems? Please indicate your choice by making a tick mark against the appropriate option (Yes/No) |
| If you answered No to question number 11, can you explain why technology can not help to tackle manual problems? |
| If you answered Yes to question number 11, can you explain how the technology can help to tackle manual problems? |

TABLE V: QUESTIONS RELATED TO THE LEADERSHIP

| Question |
|---|
| Do you think a leader (a person that generally talked more, tried to solve conflicts of other people, used more the displays, and so on) emerged during the test using the [**traditional approach—Conference Scheduler**] ? If yes, mark with a tick the box corresponding to the person you think emerged as a leader. Otherwise, leave the boxes empty. You can ask to the PI which number corresponds to a certain person, if you do not remember. You can also mark you as leader. |

TABLE VI: THE TABLE PRESENTS THE SCORE OBTAINED BY EACH PARTICIPANT IN THEIR GROUPS FOR BOTH THE APPROACHES, AND THE TIME CONSUMED TO PERFORM THE TASK AND THE TWO PHASES. THE WINNER FOR EACH GROUP IS IN BOLD. THE COLOR OF THE PLAYER CORRESPONDS TO THE COLOR OF THE LINE IN THE NEXT GRAPHS REGARDING THE SCORE

| Task results for each group and approach | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Group | Approach | Overall Score | P1 | P2 | P3 | P4 | Time | 1st ph. time | 2nd ph. time |
| G1 | Trad. | 143 | 35 | 35 | 35 | **38** | 19m55s | 11m17s | 8m38s |
| G2 | Trad. | 142 | **42** | 38 | 27 | 35 | 16m37s | 10m05s | 6m32s |
| G3 | Trad. | 140 | 33 | 34 | 36 | **37** | 31m27s | 12m21s | 19m06s |
| G4 | Trad. | 141 | 36 | 39 | 29 | **37** | 28m35s | 9m06s | 19m29s |
| G5 | Trad. | 141 | 37 | **38** | 34 | 32 | 39m57s | 11m41s | 28m16s |
| G1 | Display | 142 | 34 | 35 | 33 | **42** | 10m39s | 9m26s | 1m13s |
| G2 | Display | 143 | 38 | 38 | **39** | 28 | 29m27s | 10m26s | 19m01s |
| G3 | Display | 143 | **37** | 36 | 35 | 35 | 32m07s | 11m26s | 20m41s |
| G4 | Display | 139 | 29 | **38** | 36 | 36 | 30m22s | 10m25s | 20m07s |
| G5 | Display | 145 | 36 | 35 | **37** | **37** | 35m25s | 13m57s | 21m28s |

by analyzing the overall score of the final schedules and then analyzing the score obtained by each participant. We discover from the collected data that there is no winner between the traditional and the display approach. Then, we move our analysis and discussion onto some interesting characteristics and statistically relevant deduction on the use of LMVDs inferred from the data collected. The characteristics we studied are the user proximity to the display, the visual attention, the verbal communication, the shape assumed by the groups and the display usage. We also analyze the semantic of the communication distinguishing among negotiation and non-negotiation. A final paragraph will summarize our findings trying to connect the results and our deduction all together.

### 5.7.1 Scores

We gathered the information related to the experiment and their results in Table VI. As we can see from the *overall score* column, the results are similar for both approaches. There is also a very high similarity between the scores of the single participants.

In terms of overall score, it is easy to observe how there are no particular differences between the two approaches. At the end of the schedule session, each group obtained for both approaches a similar result in term of overall score. Indeed, the overall score average for the whiteboard approach and the display approach are respectively 141.4 and 142.4 with a variance respectively of 1.3 and 4.8. Although the variance for the display approach is higher, there is no statistical significance to assume them

different. There is no need to run a t-test to show that the two approaches do not differ by the overall score. To analyze this measure better, we want first to observe the evolution of the overall score of the schedule in the time. The overall score is obtained as sum of the single scores obtained by each participant. The graphs in Figure 34 show this information. In each graph, the red curve represents the overall score for the traditional approach while the black curve does it for the display approach.
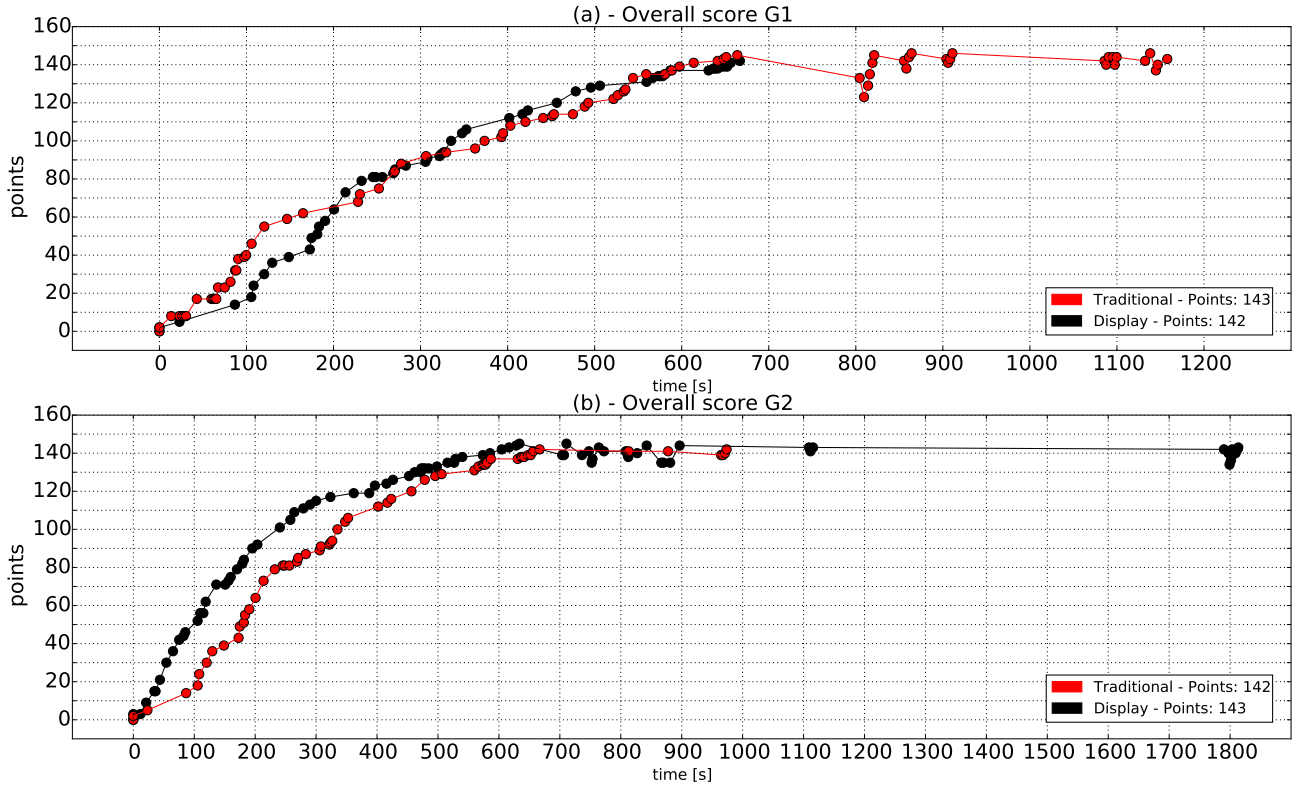


Figure 34: The graphs represent the overall score of the scheduling over the time for G1 (a) and G2 (b). Each point represents a participant action of placing a post-it in a cell. The increase/decrease of the score corresponds to a participant action that resulted in increasing/decreasing his personal score. The traditional and the display approach are represented respectively by the red and the black line.

Another important factor is the time consumed by the session. However, it also happens in this situation that the averages are very similar ($\mu_{display} = 27m26s$, $\mu_{traditional} = 26m55s$) and the variances too. This information is very important to understand why data seems to not communicate any improvement using the more technological approach. Independently from the higher information level shown by the LVD (implied by its dimension, brightness and definition), LVDs advantages are mitigated

by the inherently difficulty of the problem. In the background chapter, we showed how this problem is very similar to a NP-Complete one. By consequence, finding a good schedule in a rational time is not an easy task by definition of this family of problems. A factor that particularly influences the time extent of each session for each approach is the configuration reached at the end of the first phase. Indeed, there were situations in which the conflict resolution phase lasted less than ten minutes ($G1_{trad}$, $G2_{trad}$, $G1_{disp}$), while in others it lasted more than the double. There is also a particular case in which the second phase lasted approximately less than a minute ($G1_{disp}$). It is simply to understand that this is mainly due to the difficulty of finding an admissible schedule imposed by the configuration reached after the first phase. However, this is not the only cause of this difference in time. In certain situations, some participants argued more to obtain or maintain more valuable slots, while in others participants traded their most valuable slots to respect their constraints quickly. In all the study runs, there was no case when participants respected all their constraints in the first phase. Indeed during this phase, participants tend only to maximize their personal score and to respect the given constraints. No one thinks to leave a valuable slot empty to allow others participants to fill it up with one of their post-it notes. In such a way, the configuration reached at the end of the first phase is independent from the used approach, but it is more due to chance. We can also observe how the duration of the first phase is very similar in all the runs. While there is a higher variation on the values of the second phase.
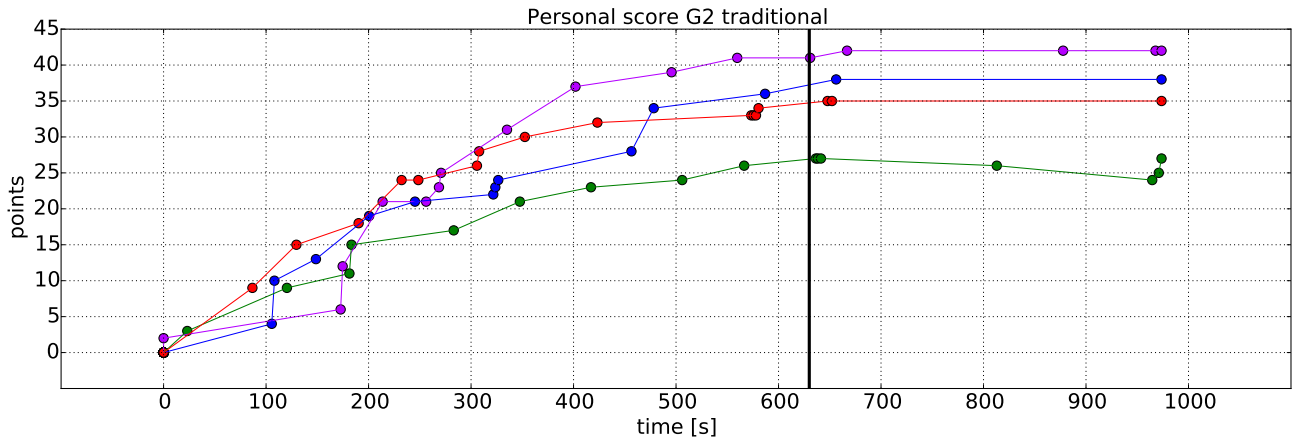


Figure 35: The graphs represent the score evolution of each participant. The line colors are coherent with the ones used in Table VI.

Another interesting way of analyzing the quality of the produced schedule is by making observation on the scores obtained by each participant. A good schedule should have a reduced variance among the single scores. It is evident that these variances in the two approaches are also very similar and no statistical test are needed to show this. Having a high variance implies having individual scores too sparse around the mean and we discovered that this situation is present in two possible case. In the first case, some participants might have many post-it notes placed in valuable slots, while others occupy low quality slots. In the second case, some participants might have placed their most valuable post-it notes in low valuable slots without negotiation in the second phase for a better position. This last situation is evident in $G2_{Traditional}$, as we can see from Figure 35. The green participant at approximately 300 seconds started to have a score lower than the others. Analyzing the videos, he did not place his most valuable paper in one of the most valuable slots while all the others did it. The purple participant exploited this situation by occupying the most valuable slots left free by the green participant. Indeed, approximately from the same time instant, the purple participant started to have a higher score than the others. Actually, this big variance of the scores is not a relevant problem for the final schedule quality. The second phase, whose start is indicated by the vertical line in Figure 35, is the place for a participant to negotiate better slots. However, in $G2_{Traditional}$ the green participant did not negotiate to improve his score at all, but only switched among his post-it notes to satisfy the violated constraints. Consequently, we are convinced that the quality of the schedule is independent from the employed approach, but it is affected by other variables, such as the type of participants and the configuration reached at the end of the first phase.

### 5.7.2 Display Proximity

Differently from the previous paragraph, in this one we only take into account the study runs with the display approach, since we want to understand how people behave with a LMVD. The conference scheduling problem is a task divided in two different phases. As expected, users behaved differently within the phases. Moreover, it does not require an intense interaction with the display since part of the required knowledge is user's knowledge-in-the-head. Furthermore, in our case this particular knowledge was contained in the constraint documents given to the participants. The usage of a physical document affected other measures, such as the visual attention, but not the display proximity. Our analysis on display proximity begins studying the entire task, then we will decouple the two phases discovering

substantial differences between them. We will find this tendency for each measure. The bar-charts in Figure 36 and Figure 37 summarize the participants' display proximity relative frequencies and their averages for the whole extent of the experiment (a) and for the two different phases (b) and (c). Even though a central far position was preferred by participants (35.74%), we summed it with the lateral far position (20.1%) since participants tended to oscillate between these two position and no significant difference was observed on the scores of the ones that occupied these two positions.

Due to the particular nature of the task, participants mainly stayed far from the display. We can notice this behavior from Figure 36-(a) looking at the horizontal line indicating the 50% of the relative frequency. If lighter colors cross it, a participant mostly stayed far from the display (14 participants over 20). This confirms our initial hypothesis: users needed to stay mainly far from the display to have a better overview of the current situation and to plan their strategies (55.84%). There are only rare situations in which a user preferred to mainly stay in a medium distance, as for $P_3G_1$ and $P_1G_4$. If we divide the two phases of the task, we will discover more precisely how participants acted. The first phase is turn based. Each participant is free to stay in the position he desires, but can place his post-it notes only when it is his turn. To have a better overview of the current situation, participants preferred to take a far place from the screen and to approach the display only when it was their turn. This behavior is confirmed by the collected data. As we can see, in the first phase people stayed preponderantly in a far position (69.31%). In the second phase, participants preferred to shorten their distance to the display (55.16%), as we hypothesized. This is the negotiation phase where participants have to discuss with the others to obtain better slots and to respect the violated constraints. Also in this situation, participants infrequently occupies a close position to the display (12.18%), but doing it only when an interaction with the display was needed. Differently from the first phase, participants used two main strategies in the second phase: 15 participants stayed in a medium position more frequently while 5 participants stayed in a far position more frequently. Although there is a clear preference in occupying the *medium* position, it is interesting to notice the tendency of someone to stay farther in the negotiation phase. Instead, in the first phase only two people ($P_3G_1, P_1G_4$) broke the tendency to stay in a farther position. Actually, there is an interesting correlation between these two participants and their scores. They are the only two participants obtaining a score lower than 30 while using the display
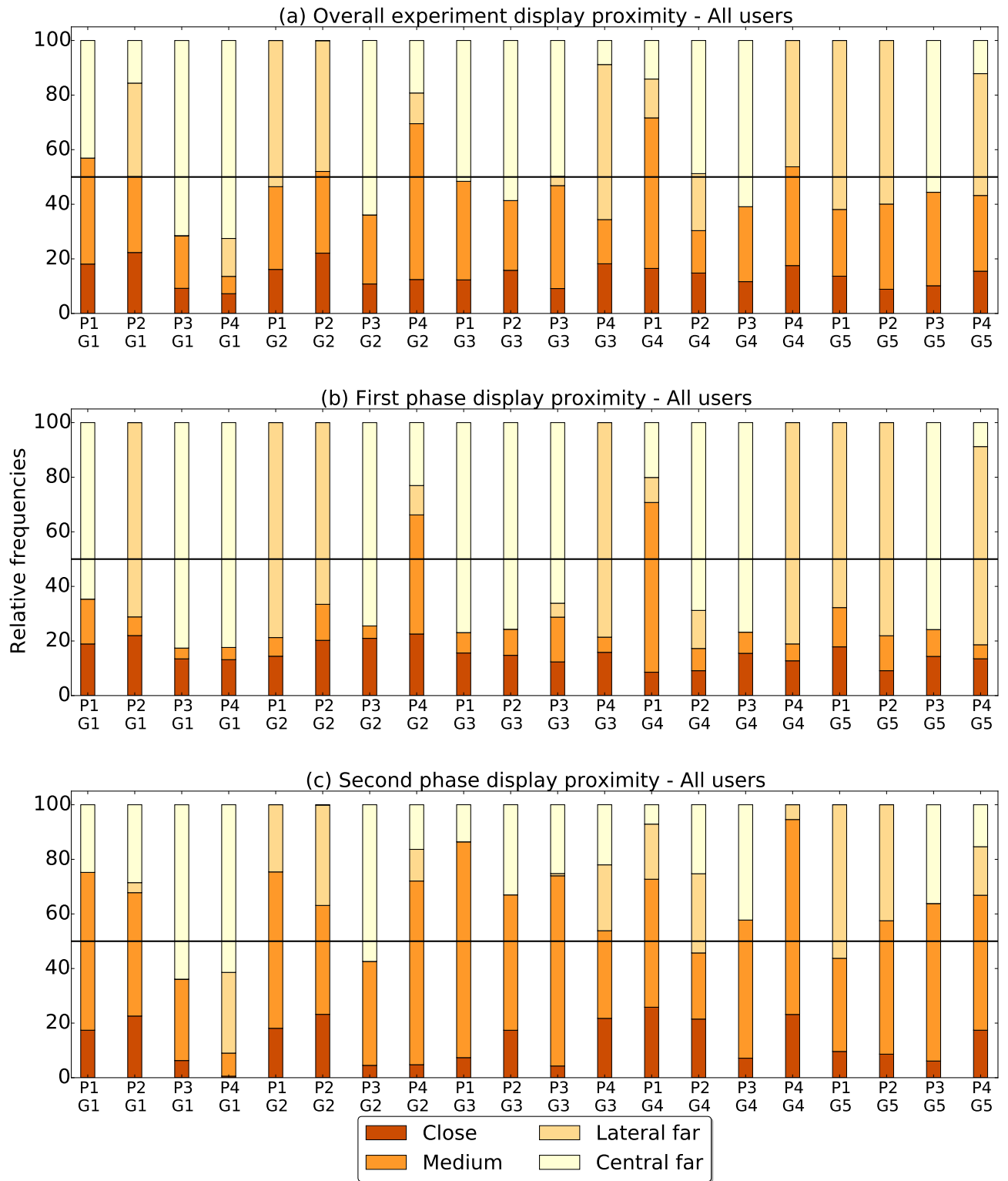
Figure 36: Stacked bar charts representing the relative frequencies of each participants' proximity to the display for the whole extent of the experiment (a) and for the two different phases (b) and (c). The horizontal lines help to understand if a user occupied more a far distance or a closer distance. For each bar, Px Gy indicates that the information is related to the participant x of the group y.
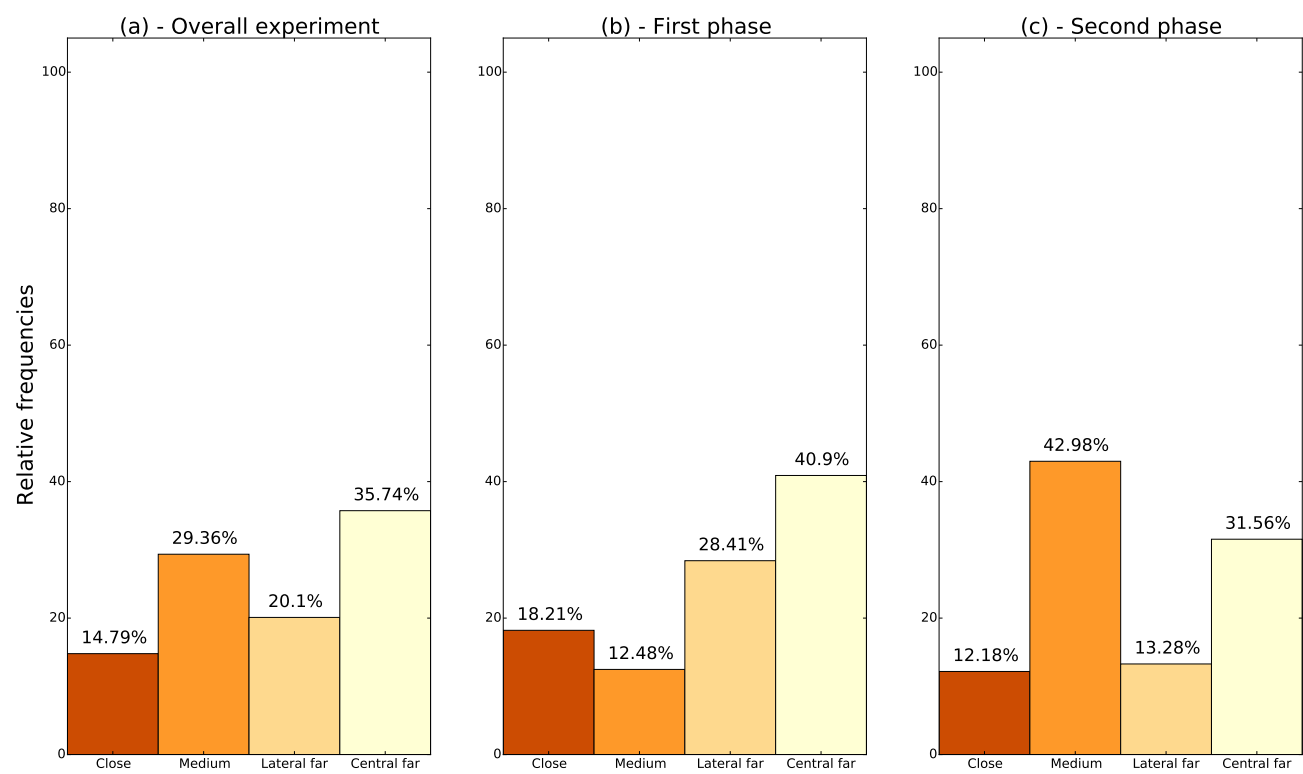
Figure 37: These three bar charts represent the relative frequencies of the verbal communication coding in the whole experiment duration (a) and in the two phases (b) and (c). We can appreciate here how these frequencies change depending the phase.

approach. However, there is not enough data to state anything of significant about the correlation between the preference of occupying a *medium* position in the first phase and obtaining a low score.

### 5.7.3 Verbal Communication

The verbal communication code frequencies depended substantially on the schedule configuration reached at the end of the first phase. We can observe this phenomenon looking at the stacked bar chart on verbal communication in Figure 38. There are two different ways in which verbal communication behaves; when *silence* is higher and when the *talking* states are higher. It is possible to understand this phenomenon by analyzing the verbal communication in the two task phases. In the first phase, there is not a lot of space to talk (16.69% sum of talking states) since everyone is focused on respecting his constraints and maximize his score. When participants talked in this occasion, they only made general comments without the need of long conversations (maximum time of talking during first phase is 26.01s). Here participants preferred to stay silent (83.31%), as shown in Figure 39-(a). On the other phase (Figure 39-(b)), people talked much more in groups (37.83%) or all together (45.25%).

TABLE VII: STATISTICS FOR VERBAL COMMUNICATION CODES

| Category | Absolute freq. | Relative freq. | Mean [s] | Median [s] | Max [s] |
|---|---|---|---|---|---|
| Silence | 142 | 41.72% | 24.31 | 13.89 | 134.11 |
| One Talking | 55 | 5.06% | 7.61 | 6.67 | 14.4 |
| Group Talking | 130 | 25.96% | 16.48 | 13.21 | 42.26 |
| All Talking | 87 | 27.27% | 25.96 | 20.49 | 87.85 |

Rare are the occasions in which only one participant talked (6.02%) or everyone stayed silent (10.9%). By consequence, the percentages of the verbal communication for the whole extent of a run is highly dependent on the duration of the second task. In those occasions in which the negotiation phase lasted less, silence is preponderant as in G1 and G3. Otherwise, when the negotiation phase lasted more, the effects of the talking categories have more influence, as in G2, G4 and G5.

We want now to analyze the frequencies and the duration of these periods. Table VII summarizes the statistics for the verbal communication codes. As we can see from the median, the periods of these categories have generally shorter duration than the mean. This is especially observed for the *all talking* and *silence* codes that have respectively a mean of 25.96s and 24.31s and a median of of 13.89s and 20.49s. Moreover, it is interesting to notice how the maximum duration of the silence has a value of 134.11s. Long periods of silence where observed mostly in the first phase, but they were often interrupted by someones' voice. To perform a more accurate analysis, we want to understand if participants communicated differently depending on the phase. For all the categories, we performed an F-test for the variance and a T-test for the mean to understand if the samples belong to the same or a different population. The analysis we did brought us to the following observations:

- *silence*: in the first phase when participants are silent, they maintain this condition longer (M=25.73s) than in the negotiation phase (M=12.94s)

  $(F(97, 45) = 1.563, \hat{F} = 8.4, p < 0.001 \text{ --- } t(133) = 1.656; \hat{t} = 4.717, p < 0.001)$;

- *group talking*: participants have longer group conversations in the second phase (M=18.91s) than in the first phase (M=12.56s)

  $(F(94, 34) = 1.650, \hat{F} = 2.809, p < 0.001 \text{ --- } t(101) = 1.66; \hat{t} = 3.859, p < 0.001)$;
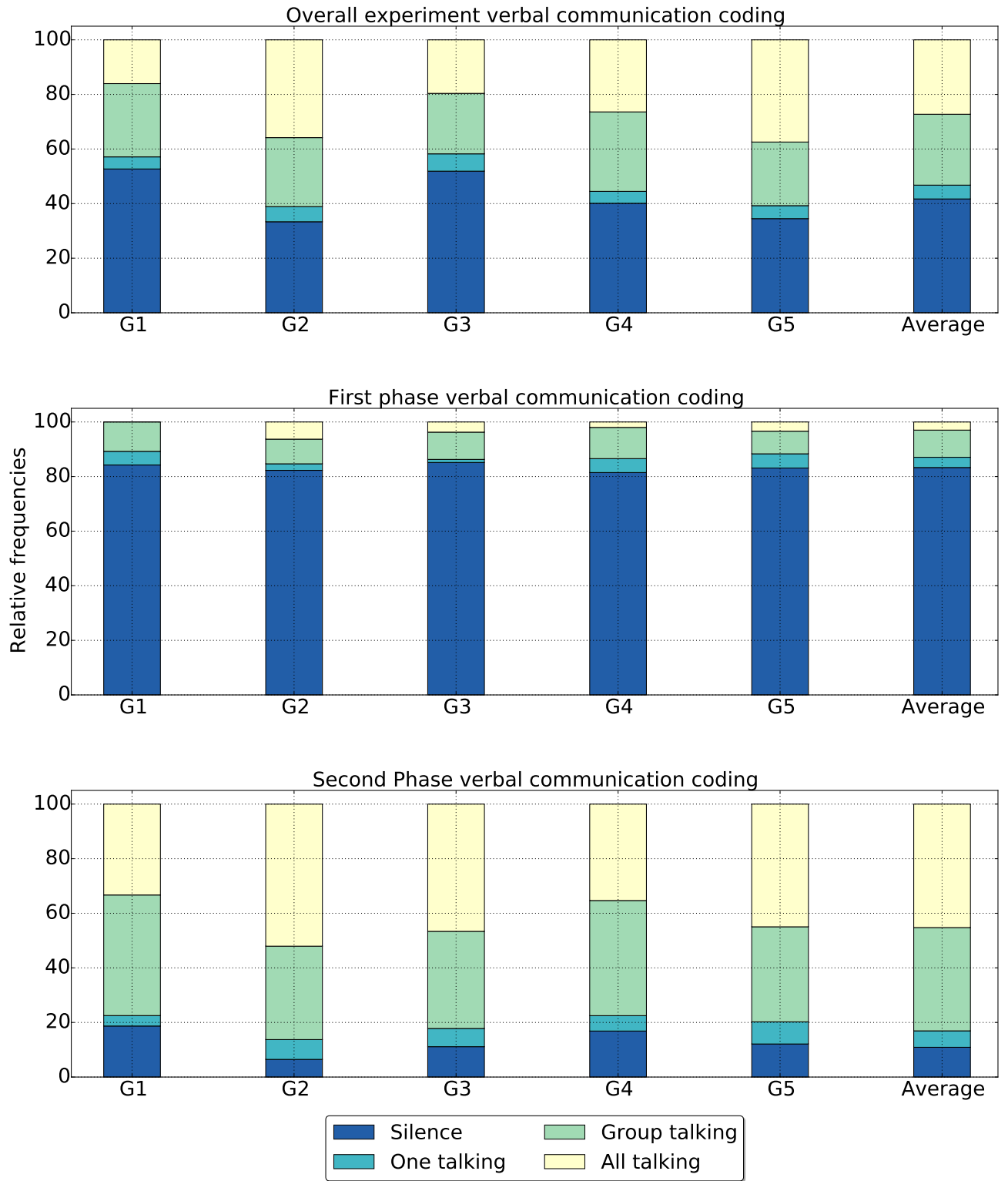
Figure 38: Stacked bar charts representing the relative frequencies of group shape coding. The firsts five histograms are representative of each group. The last histogram represents the average of this set of codes for the whole user study. It is coherent with the frequencies contained in Table VII.
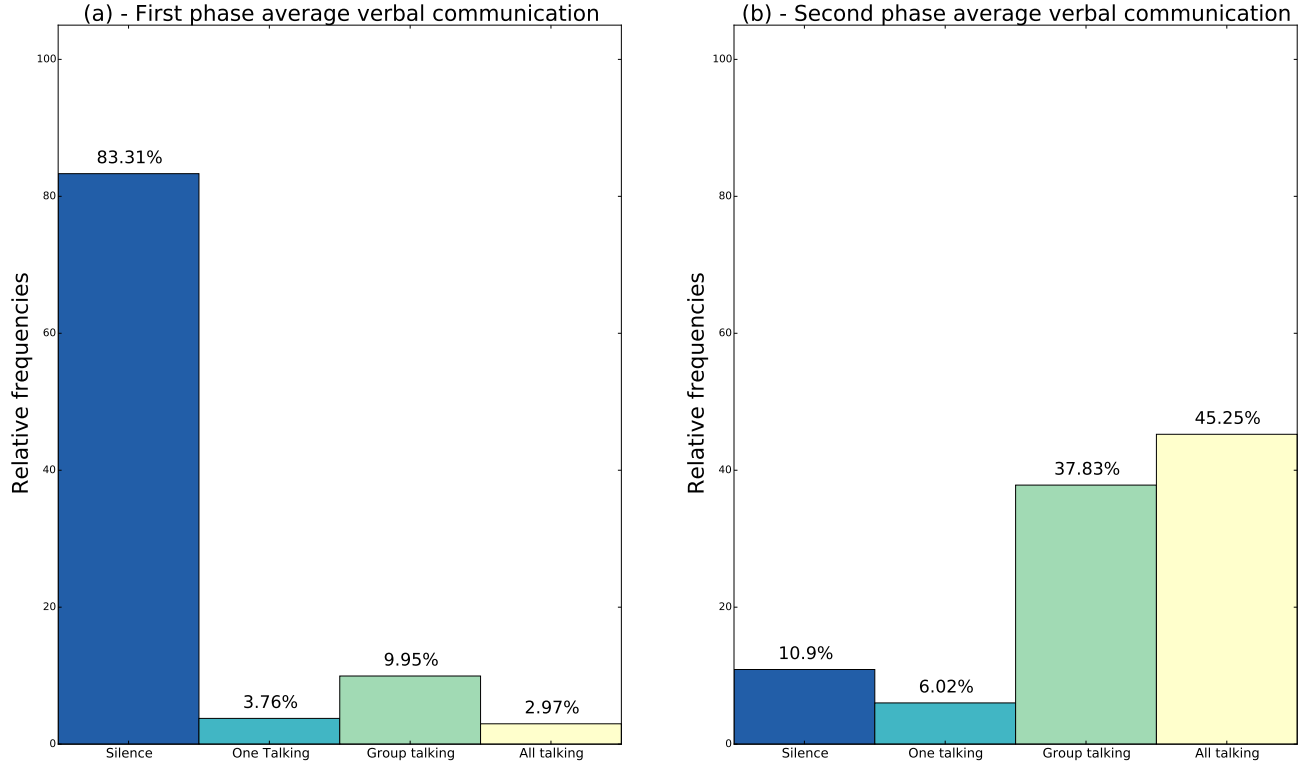
Figure 39: These two bar charts represent the relative frequencies of the verbal communication coding in the two phases. We can appreciate here how these frequencies change depending the phase.

- *one talking*: when a participant says something alone, his talk is longer in the second phase (M=8.38s) than in the first one (M=6.38s) $(F(32, 23) = 1.978, \hat{F} = 9.138, p < 0.001 — t(40) = 1.684; \hat{t} = 3.351, p < 0.001)$;

- *all talking*: it is unlikely that participants talk all together in the first phase.

We will take the verbal communication into account when visual attention and group shape are introduced. Further analysis on their joint distribution will be done to understand if these variables are dependent.

### 5.7.3.1   Conversation Semantics

We did not find any differences regarding the semantics of the verbal communication between the two approaches. Participants discussed in the same way independently of the approach. The number of complaints related to the frustration on finding an admissible schedule increased evenly for both approaches as the time consumed in the negotiation phase increased. A statistical test confirmed that the number of complains for both approaches belongs to the same population. This still confirms that

TABLE VIII: NUMBER OF COMPLAINTS FOR EACH GROUP FOR BOTH APPROACHES

| Group | #Complaints traditional app. | #Complaints display app. |
|---|---|---|
| **G1** | 6 | 0 |
| **G2** | 3 | 10 |
| **G3** | 11 | 8 |
| **G4** | 13 | 13 |
| **G5** | 16 | 14 |

the employed approach does not influence the task itself. We can find the number of complaints divided by group and approach in Table VIII.

For this reason, we analyze the semantic of the conversations only for the display approach since we want to describe the human behavior while interacting with a LMVD. An interesting detail is that participants were very focused on the task, observing the others' moves and planning their own strategies. This behavior did not leave any space for conversations not related to the task. Indeed, dividing the semantic of the conversation in *negotation*, *complaints*, and *general comments*, 87.1% of the conversations duration regarded negotiations while the remaining part was related to general comments, such as *'Is it my turn?'*, or complaints, such ash *'Can anyone answer me? I'm trying to conclude this session. Please.'*. Most of the general comments were done in the first phase. Here are rare complaints (7.3%) while negotiation conversations are completely absent. In the second phase, participants talked mainly to negotiate (91.1%) and sometimes complained about the difficulty on finding a solution (7.3%).

The most frequent ways of starting a negotiation were using the sentences *'Can I trade this slot ...'* and *'Can you move this paper ...'*. To show a typical situation in which participants remained stacked in a situation, we displayed below a conversation where they tried to negotiate a slot, but they did not find an agreement quickly.

```
P1 - You are not losing anything.  You are gaining points, but you want to gain more points.
P2 - No.  No, I don't wanna.  I'm not gaining points.
P1 - You are gonna gain points.
P2 - No, because its value is two.  And you want to move my post-it here that is also two.
P1 - So, you are not losing anything.
P2 - Exactly, I already lost point, my next move will make me earn points.
P1 - No, you lost points because you wanted to be kind to him.
P2 - No.  Because I could help to find a solution, but I already lost points.  Other guys
     can now lost their points or whatever.
P3 - Come one, P1.
P1 - Tell him.  He wants to gain points.
```

```
P2 - It's your fault.  It's your fault.  Because you didn't respected the constraints.
P3 - Yes, but he already lost points and you have a problem.
P1 - No, no.  No, no.  Because of you.
P2 - Yes, the problem is yours.
P1 - If I have switch up, I'm gonna lose points.  So, I don't wanna lose a lot of points.
   So, if you want to understand...
P2 - I already lost points.  I already lost points.
P1 - Wait, wait, wait.  If you want to understand, I'm not gonna to lose a lot of points.
P4 - You're right, I have another solution.
(Conversation continued for other 2 minutes.)
P1 - There is two solutions.  One solution is I lose a lot of points, he gains one...
P2 - I'm not gaining, I'm not gaining.
P1 - ...can I finish?  Can I finish?
P2 - Ok.
P1 - One solution.  I lose a lot, he gains one.
P2 - How?
P4 - He is not gaining one.
P1 - Wait, wait, wait.
P2 - How?
P1 - Oh, you are right.  You don't get anything.
P2 - Eh, exactly.
P1 - So, nothing happens.  I lose a lot points.  This is first solution.  We finish,
   but I lose point.  Sounds good, right?  Second solution...
P2 - Sounds good.
P4 - You can still move...
P1 - Please, let me finish.  For the love of god.  One time in my life.
(Silence)
P1 - Ok.  Nobody loses anything, I lose points, we finish.  Solution number one
   Solution number two:  I lose points, he gains points, we finish.
P2 - Solution number two.
P1 - No!
P2 - Solution number two!
P1 - No, because if I lose points, you gain more points.
P2 - I wanna win.
P1 - You are not going to.
(...after 6 minutes of conversation, they agreed with the solution number one proposed
   by P1)
```

### 5.7.4   Visual Attention

Before starting the data analysis for visual attention, it is worth to mention that the task employed in our user study has the particularity to use also a physical document, the constraints sheet, that serves as external memory for the user. In the occasion in which this task is executed for an actual conference, the knowledge provided by the constraints sheet is owned by the chairs. This means that users visual attention might be directed on the display longer than in our experiment. However, users might use an external support as well. An interesting variation of the experiment could have been to provide the constraints through the LVD, but this would have meant changing the scheduling task essence. The presence of a physical item serving as external memory let us create the *mixed* code for

TABLE IX: STATISTICS FOR VISUAL ATTENTION CODES

| Category | Absolute freq. | Relative freq. | Mean [s] | Median [s] | Max [s] |
|:---:|:---:|:---:|:---:|:---:|:---:|
| Mixed | 119 | 27.50% | 19.03 | 12.89 | 77.05 |
| Document | 34 | 4.19% | 10.14 | 9.85 | 17.02 |
| Display | 167 | 29.42% | 14.49 | 9.93 | 70.43 |
| Disengaged | 8 | 0.73% | 7.05 | 6.2 | 9.26 |
| Each other | 147 | 38.38% | 21.45 | 16.63 | 100.55 |

the visual attention measure. When participants are in the *mixed* state, they continuously switch their visual attention between the document and the LVD. If we would merge this state with the *display* state, this new state would be the more frequent for the visual attention. Instead, considering our coding there is no predominant state in average on the others, as Figure 40 shows.

Table IX shows the statistics for the visual attention codes. *Display* (29.42%), *each other* (38.38%) and *mixed* (27.50%) states have close and high relative frequencies. Instead, *document* and *disengaged* states have both low relative frequencies (4.19% and 0.73% respectively). Such a low percentage for the relative frequency of *disengaged* state indicates that in general the group is involved in the task. Furthermore, there is a high variability of the duration of the states. For example, the *each other* code has a mean of 21.45s, a median of 16.62s and a maximum of 100.55s.

The same phenomenon observed for verbal communication (the big diversity between the two phases) is present also for the participants visual attention and we separate now the analysis between the two phases of the experiment, as usual. The bar charts in Figure 41 show the data for the visual attention respectively for the first and the second phase. The two phases are respectively very similar in all the study runs. In the first phase, participants are mainly in the *mixed* state (52.3%) or are looking at the display (33.25%). Basically, all the time spent looking at the documents for a considerable amount of time interval is consumed in the first phase (10.53% of the first phase) since no groups looked at their documents for more than five seconds in the second phase. In the fist phase, no one is *disengaged*, while sometimes participants look at *each other* (3.92%). This frequencies for the visual attention states are expected since participants needed to acquire knowledge about the constraints in their material and to plan and modify their strategies depending on the configuration evolution, so looking at the display and their documents. As expected, a completely different behavior is acted in
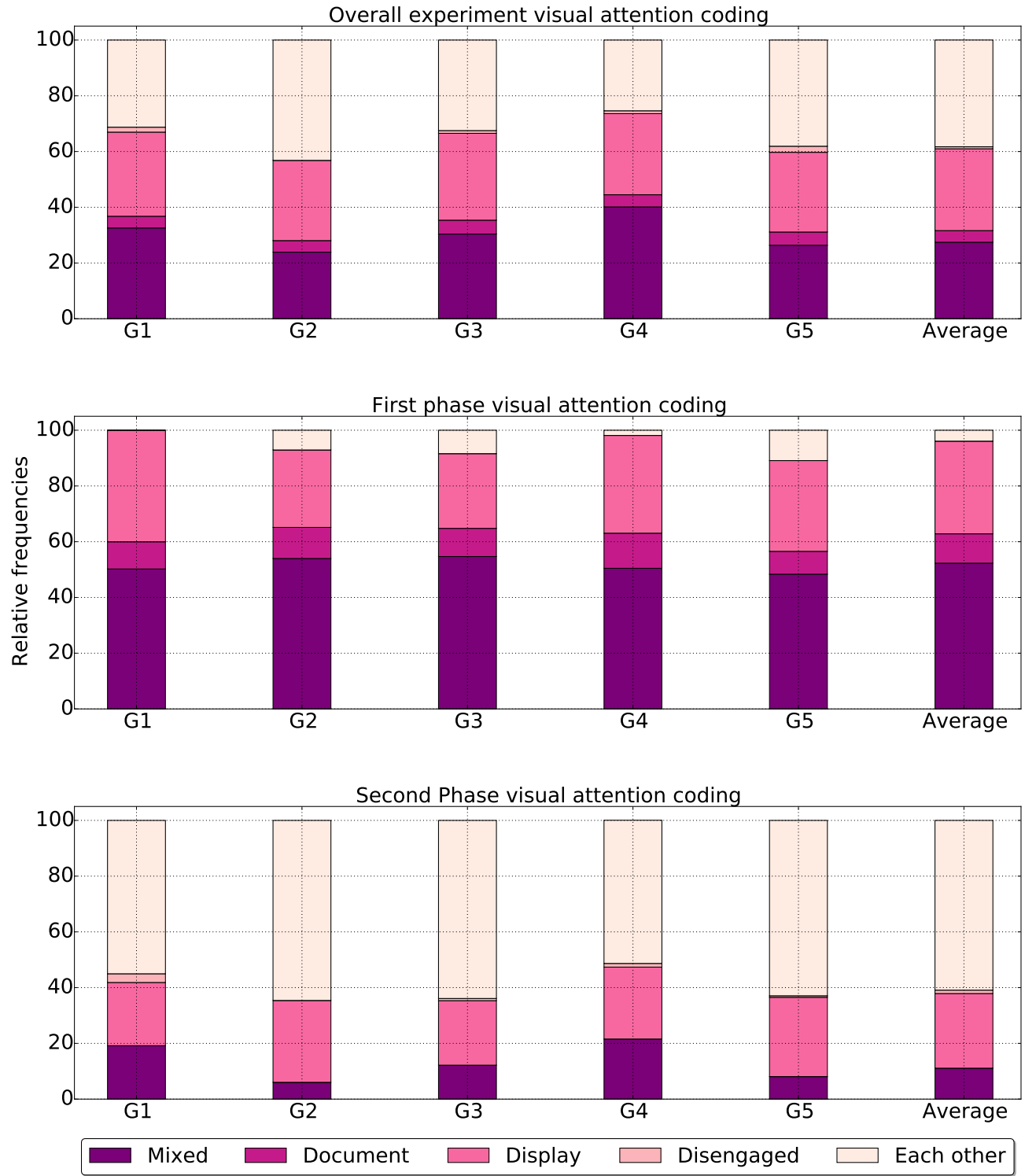
Figure 40: Stacked bar charts representing the relative frequencies of visual attention coding. The firsts five histograms are representative of each group. The last histogram represents the average of this set of codes for the whole user study. It is coherent with the frequencies contained in Table IX.

the second phase. Participants look at *each other* for most of the time (60.91%). A consistent quantity of time is still used to look at the *display* (26.79%), but the quick sight switch between document and display are less frequent (11.1%), even though still present. Sometimes most of the participants are *disengaged* (1.2%) and generally it happens after a discussion occurs.
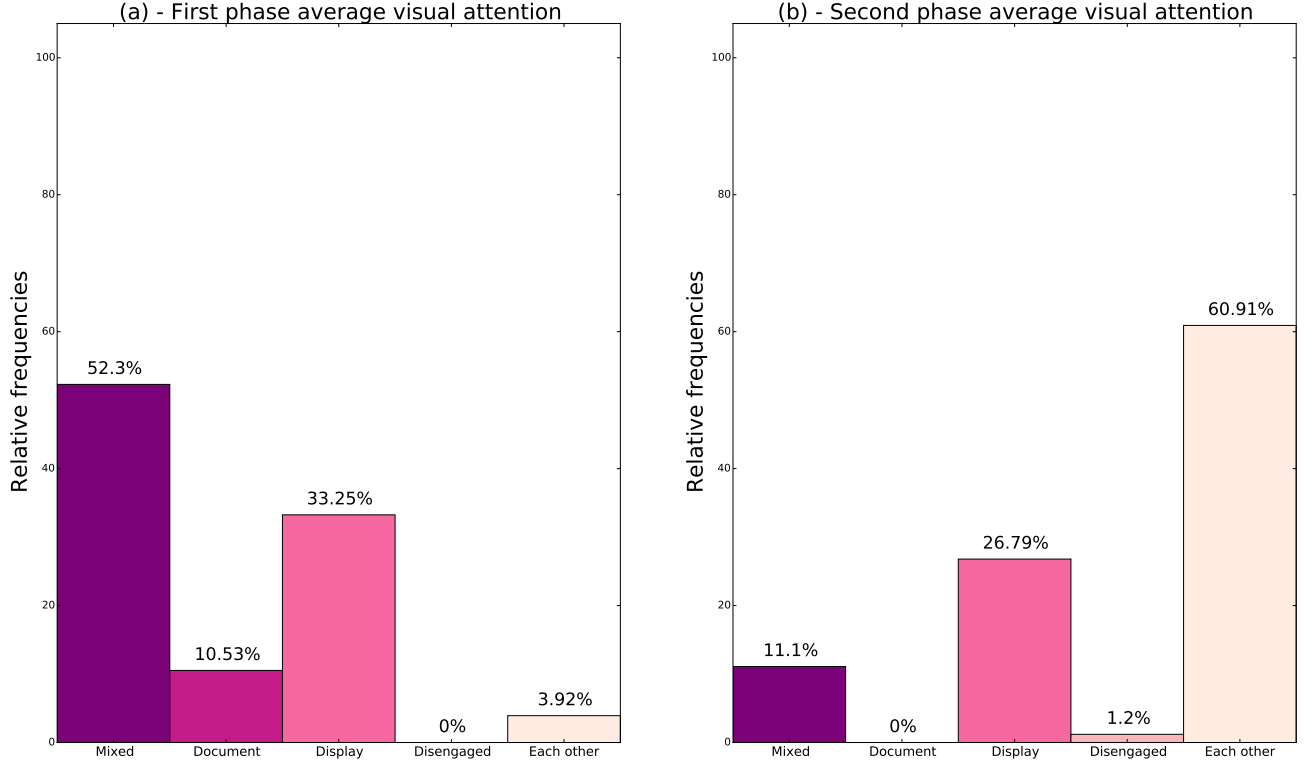


Figure 41: These two bar charts represent the relative frequencies of the visual attention coding in the two phases. We can appreciate here how these frequencies change depending the phase.

As we did for the verbal communication, we analyzed if the coded visual attention states have different characteristics for the two phases. Since data for the *disengaged* and the *documents* states are absent in the first and second phase respectively, we only state that these two states are likely to happen in those distinct phases. We found that depending on the phase, the following codes for visual attention have the following characteristics:

- *mixed*: participants change their attention frequently between their documents and the display for a longer period during the first phase (M=21.02s), than the second one (M=14.82s) $(F(81, 36) = 1.641, \hat{F} = 2.033, p < 0.01 — t(96) = 1.661; \hat{t} = 2.478, p < 0.008);$

- *each other*: during the negotiation phase, participants look at each other longer (M=23.9s) than in the first phase (M=8s)

  $(F(123, 24) = 1.812, \hat{F} = 24.683, p < 0.001 — t(129) = 1.657; \hat{t} = 7.799, p < 0.001)$;

- *display*: participants look at the display for a longer time in the second phase (M=17.9s) than in the first phase (M=12.43s)

  $(F(86, 81) = 1.441, \hat{F} = 2.264, p < 0.002 — t(149) = -1.655; \hat{t} = -2.975, p < 0.002)$.

### 5.7.5     Verbal Communication and Visual Attention

A more sophisticated analysis let us make other interesting observations on the previous two measures. Verbal communication and visual attention can be studied together as did by [24]. We found that the two codes have a significant correlation using a Pearson's independence test ($\chi^2(10, 0.05) = 28.3$, $\hat{\chi}^2 = 425.13$, $p < 0.001$). Since we are studying the users behavior, we also calculated the Cramer's $V$ to show the strength of the association. According to Cohen [55], the verbal communication and visual attention has a medium effect size (Cramer's $V = 0.32$). To understand the significance of each coupled measure, we calculated a contingency table showing the adjusted residuals presented in Table X. Adjusted residuals are an easier way to show the significance of the association between two variables and are standardized values of the difference between the observed frequency and the expected frequency of the codes. In general, a positive (negative) value implies that the observed frequency is higher (lower) than the expected frequency. Since it is a standardized measure, we can use an usual gaussian p-value to understand the significance of the measure. We will use a level of significance of 0.05. Consequently, if an adjusted residual is lower than -1.96 or higher than 1.96, there is only a probability lower than 0.05% that the observation is given by chance.

The clearest association between verbal communication and visual attention is referred in the *each other* code. All the cells associated to this code have a consistent significance. The most significant association involving the *each other* state is with the *all talking* state ($a_{5,4} = 14.53$). This indicates that when most of the participants are talking all together, their visual attention is directed to the other participants. It might seem trivial, but it is not that way because participants are in the presence of a massive device in which all the task operations are being executed. At the same time, it is unusual that participants look at *each other*, if no one is talking ($a_{5,1} = -10.49$). Instead, when people are in *silence*,

TABLE X: ADJUSTED RESIDUALS FOR THE VERBAL COMMUNICATION AND VISUAL ATTENTION CODES

|  | Silence | One talking | Group Talking | All talking |
|---|---|---|---|---|
| **Mixed** | 5.88 | 2.10 | -4.22 | -5.50 |
| **Documents** | 9.78 | -2.91 | -5.42 | -3.57 |
| **Display** | -4.47 | 5.35 | 3.96 | -4.58 |
| **Disengaged** | 5.81 | -2.48 | -2.22 | -2.35 |
| **Each other** | -10.49 | -5.23 | 4.76 | 14.53 |

their visual attention is often directed to their *documents* ($a_{2,1} = 9.78$) or they quickly switch their attention between the display and the documents ($a_{1,1} = 5.88$), that is the *mixed* state. Looking at the *display* is a frequent action in the occasion in which a unique participant is talking ($a_{3,2} = 5.35$). It has the interesting consequence that when a single participant is talking, the others do not watch him, but the display. Indeed, analyzing the video recordings, it happens often that when a single participants are talking, he is referring to the display and consequently the others watch at the display. Most of the participants was rarely *disengaged* during the task, but this visual attention code is interestingly associated with the *silence* code ($a_{4,1} = 5.81$). Others minor significance effects are present and can be found in the contingency table.

### 5.7.6 Group Shape

We decided to study the shape assumed by the groups, since having an increase in the number of users in a multi-user environment makes the space they occupy more relevant, and knowing how groups occupy the environment might be helpful for setting up the space properly before a similar task is performed. We observe the same two-phase phenomenon observed in all the previous measures. The groups' behavior relative to their shape has important differences between the two phases and the overall frequencies depend on how the negotiation phase lasted. Table XI contains the information on the frequencies and other statistics for the group shape codes. We observe that the most used shape is *one isolated* (35.54%). This is due to the particularity of the first phase to be turn-based. All the other states have a relative frequency around the 20%. Figure 42 contains stacked bar charts related to the relative frequencies of the shapes assumed by the groups. There is a graph present for each group and one for their overall average.
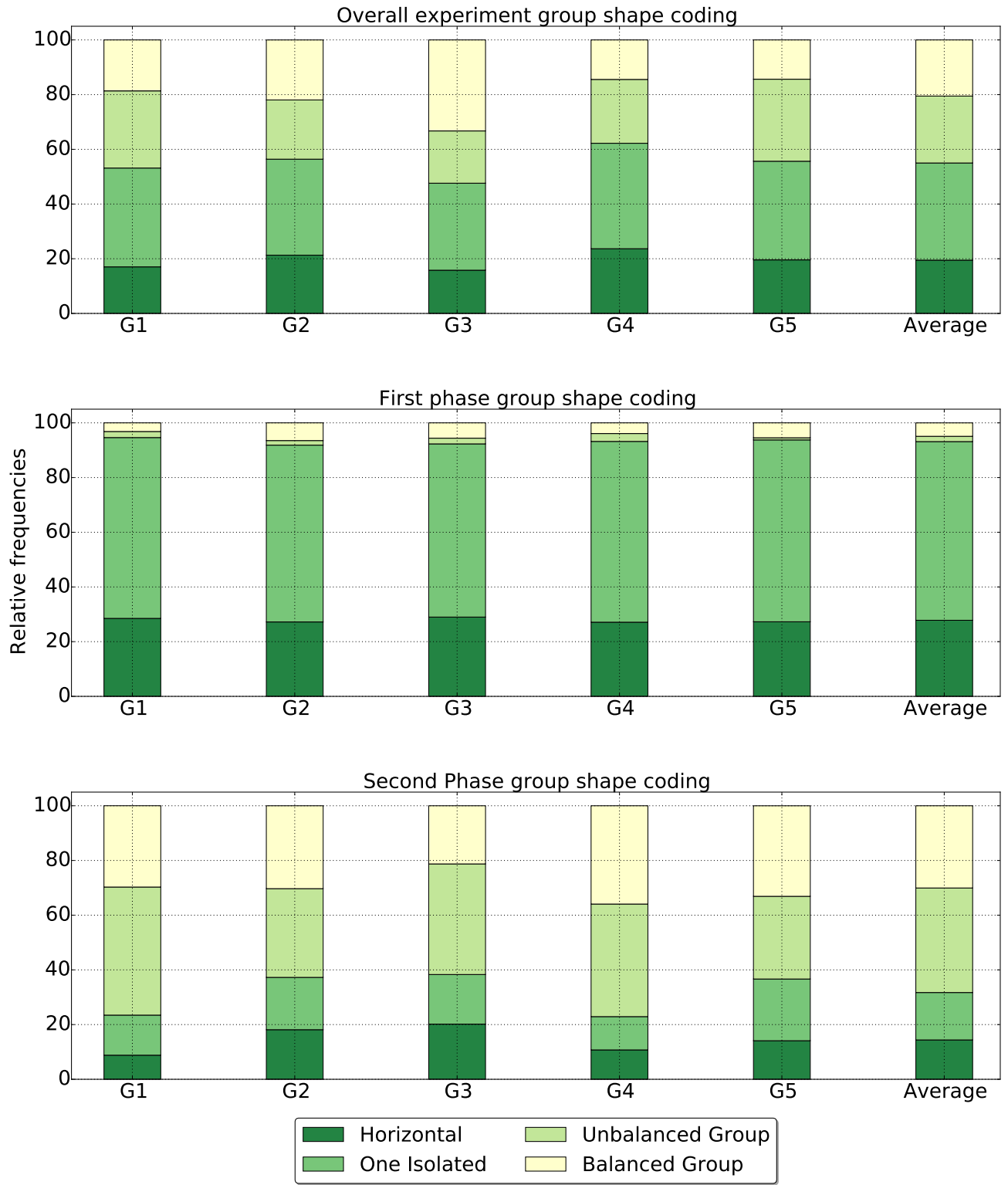
Figure 42: Stacked bar charts representing the relative frequencies of group shape coding. The firsts five histograms are representative of each group. The last histogram represents the average of this set of codes for the whole user study. It is coherent with the frequencies contained in Table XI.
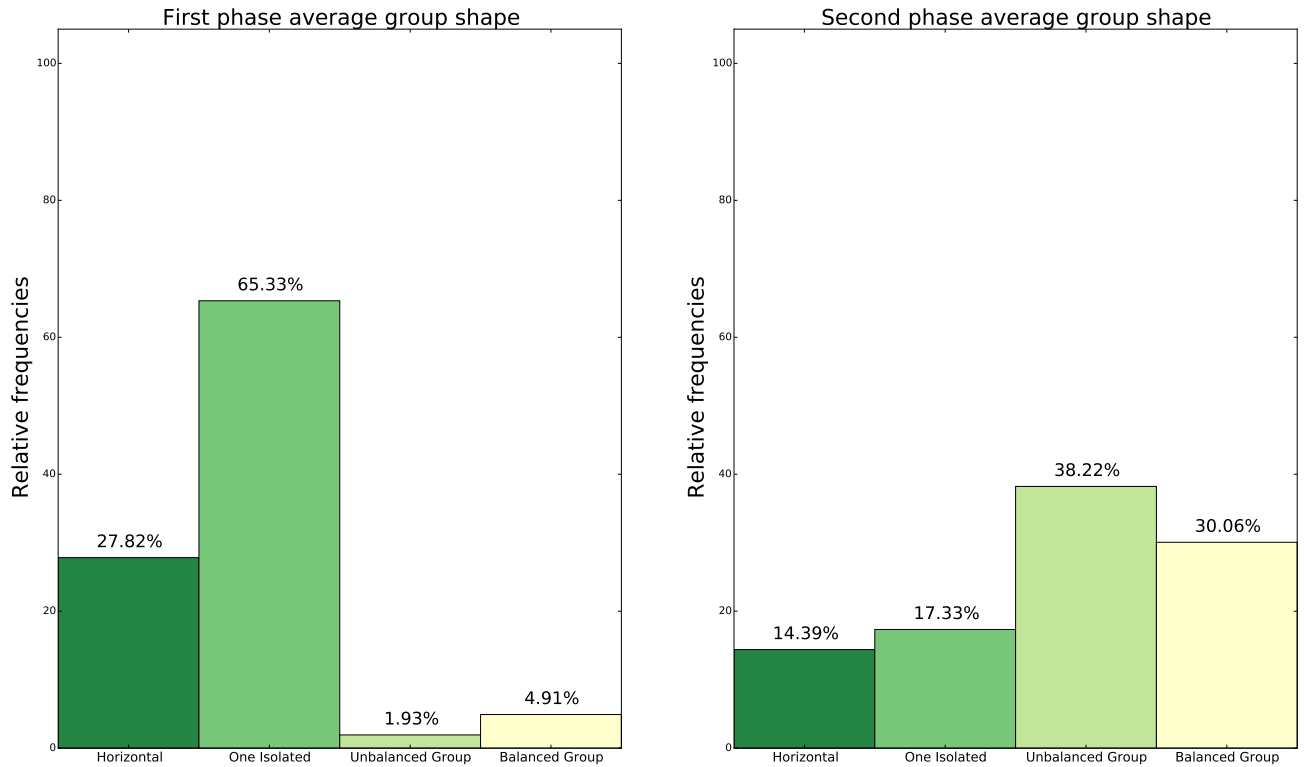
Figure 43: These two bar charts represent the relative frequencies of the group shape coding in the two phases. We can appreciate here how these frequencies change depending the phase.

Figure 43 shows the group shape divided in the two phases of the experiment. In average group shape states have all similar behavior, hence we preferred to show only a standard bar chart of the average for each phase. In this graph, it is easier to appreciate the differences between the phases and understand the variations of the relative frequencies. The first phase, being turn-based, is characterized by the *one isolated* state. Most of the time a participant is detached from the group (65.33%). In this phase, the second important state is the *horizontal* one (27.82%). Although no instruction on what position to assume was given to participants, they all preferred to stay in a horizontal and equal distance to the display during this phase; a participant detached from the group to approach the display only when he needed to perform his action. A very low relative frequency of the other two states (6.84%) suggests that participants rarely broke the non-given rule to maintain a horizontal shape when they are not called to take their turn. In the negotiation phase, the previous tendency is overturned even though the *horizontal* and *one isolated* states continue to have an interesting frequency (14.39% and 17.33% respectively). However, participants preferred to stay in groups in the second phase. There is a slight preference on assuming the *unbalance group* state, that is when two groups have an unequal distance

from the display, rather than the *balance group* state. Indeed, this code has a relative frequency of 38.22%, while the *balanced group* one is present 30.06% of the time of the second phase.

TABLE XI: STATISTICS FOR GROUP SHAPE CODES

| Category | Absolute freq. | Relative freq. | Mean [s] | Median [s] | Max [s] |
|---|---|---|---|---|---|
| Horizontal | 95 | 19.48% | 16.88 | 11.02 | 139.3 |
| One isolated | 201 | 35.54% | 14.54 | 11.07 | 59.76 |
| Unbalanced groups | 73 | 24.46% | 27.71 | 23.04 | 88.53 |
| Balanced groups | 59 | 20.52% | 28.79 | 25.89 | 118.65 |

It is interesting analyzing the differences between the same code observed in the two different phases. The codes belonging to the shape code set have the following characteristics:

- *horizontal*: the groups maintain the *horizontal* shape longer in the negotiation phase (M=27s), than in the first phase (M=11.85s). This is caused by the fact that this state is continuously interrupted by the turn changing of the users in the first phase

  $(F(24, 69) = 1.676, \hat{F} = 35.716, p < 0.001 \text{ — } t(24) = 1.711; \hat{t} = 2.123, p < 0.023)$.

- *one isolated*: a participant stays isolated longer in the second phase (M=18.26s) than in the first one (M=12.72s)

  $(F(147, 54) = 1.484, \hat{F} = 1.09, p > 0.05 \implies \text{t-test with same variance — } t(199) = 1.652; \hat{t} = -3.804, p < 0.001)$;

- *balanced group*, *unbalanced group*: it is unlikely that groups assume this shape during the first phase.

### 5.7.7 Verbal Communication and Group Shape

We found interesting associations between the verbal communication and group shape codings. Initially we found that there is a significant correlation between them $(\chi^2(9, 0.05) = 23.589, \hat{\chi}^2 = 140.84, p < 0.001)$. The Cramer's $V$ index of 0.42 indicates a medium effect size. Table XII shows the adjusted residuals for this pair. We can notice an interesting geometry of the table. Indeed, it is divided in four zones. The top-left and the bottom-right zones present positive adjusted residuals, while top-right and

TABLE XII: ADJUSTED RESIDUALS FOR THE VERBAL COMMUNICATION AND GROUP SHAPE CODES

|  | Horizontal | One isolated | Unbalanced group | Balanced group |
|---|---|---|---|---|
| Silence | 3.70 | 2.23 | -3.22 | -4.78 |
| One talking | 1.98 | 3.52 | -3.41 | -3.98 |
| Group talking | -3.75 | -2.81 | 4.26 | 4.60 |
| All talking | -3.59 | -4.25 | 4.08 | 6.46 |

bottom-left present negative adjusted residuals. The positive adjusted residuals on the bottom right are combinations of codings indicating that participants are interacting (the two group shapes and the talking of more than one user). The cells at the top-left represent low level of interaction since they are composed by the *horizontal* and *one isolated* group shape codes and the *silence* and *one talking* verbal communication codes.

When participants are in groups, both *balanced* or *unbalanced*, they generally talk in groups or all together (adjusted residual roughly higher than 4). *Silence* and talking by a person is infrequent when participants are gathered in groups since the adjusted residuals on the table top-right cells are all lower than $-3$. Passing to the low level of interaction associations, people tend to stay silent when they are parallel to the display ($a_{1,1} = 3.70$) and this also happens when a single user approaches the display, but with lower confidence ($a_{1,2} = 2.23$). What happens most when a single user is isolated and the others inactive is that one participant talks ($a_{2,2} = 3.52$), and analyzing the video we found that the one who is talking is the one that approaches the display. It is rare that there are conversations that involve more than one user in the *horizontal* and *one isolated* states since the adjusted residuals for these pairs are approximately all lower than 3. This can be identified by the bottom left of the table.

### 5.7.8 Visual Attention and Group Shape

The last codings to analyze together are the visual attention and the group shape. The two measures are correlated: visual attention differs for group shape ($\chi^2(12, 0.05) = 28.3$, $\hat{\chi}^2 = 208.3$, $p < 0.001$). The effect size is medium being Cramer's $V = 0.39$. Table XIII shows the adjusted residuals for visual attention and group shape. The first thing we notice is that people looking at *each other* are grouped together, more likely in the *unbalanced group* code ($a_{5,3} = 7.53$), but also in the *balanced group* code ($a_{5,4} = 2.75$). These two cells represent the high level interaction part of the table. Furthermore,

TABLE XIII: ADJUSTED RESIDUALS FOR THE VISUAL ATTENTION AND GROUP SHAPE CODES

|  | Horizontal | One Isolated | Unbalanced groups | Balanced groups |
|---|---|---|---|---|
| **Mixed** | 6.01 | 4.49 | -5.93 | -5.89 |
| **Documents** | 2.68 | 1.98 | -2.11 | -3.11 |
| **Display** | -3.25 | -1.90 | -1.87 | 7.48 |
| **Disengaged** | -2.21 | -1.75 | 5.27 | -0.71 |
| **Each other** | -4.94 | -4.10 | 7.53 | 2.75 |

participants are more likely in *unbalanced group*s, if they are *disengaged* ($a_{4,3} = 5.27$). If participants are looking at the *display*, they are often in a *balanced group* ($a_{3,5} = 7.48$). This indicates that when groups are at the same distance from the display, they actually look at it. The top-right part of the table involves the groups, the *mixed* and *document* states. The fact that these adjusted residuals are lower than $-2$ indicates that when participants are gathered in groups, they do not often look at documents, neither do they enter in the *mixed* state. In the top-left part of the table, we have four positive and significant adjusted residuals indicating that when the participants have a *horizontal* or *one isolated* shape, they often look at their *document*s and also enter in the *mixed* state. In the end, it is infrequent that in a *horizontal* shape, participants are *disengaged*, looking at *each other* or at the *display* for more than 5 consecutive seconds. It is very interesting to notice how the *display* code for visual attention has negative value for all the adjusted residuals, except for the *balanced group* state. This is justified by the fact that the *mixed* state is generally predominant in the first phase, rather than the *display* state, as we described previously. Hence, when participants look at the display for more than 5 seconds, they are in *balanced groups*.

### 5.7.9    Display Usage

At the beginning of the scheduling session with the display approach, each group was able to set up the environment within the LMVD. Each participant had to position and resize his paper container in a desired position, while the whole group had to decide how to position and resize the schedule table. Before starting the task, each participant has to agree with the final positioning of the items in order to avoid someone taking a better position or having and oversized container. In such a way, groups were free to choose how to organize the space within the LMVD. This decision has influenced how
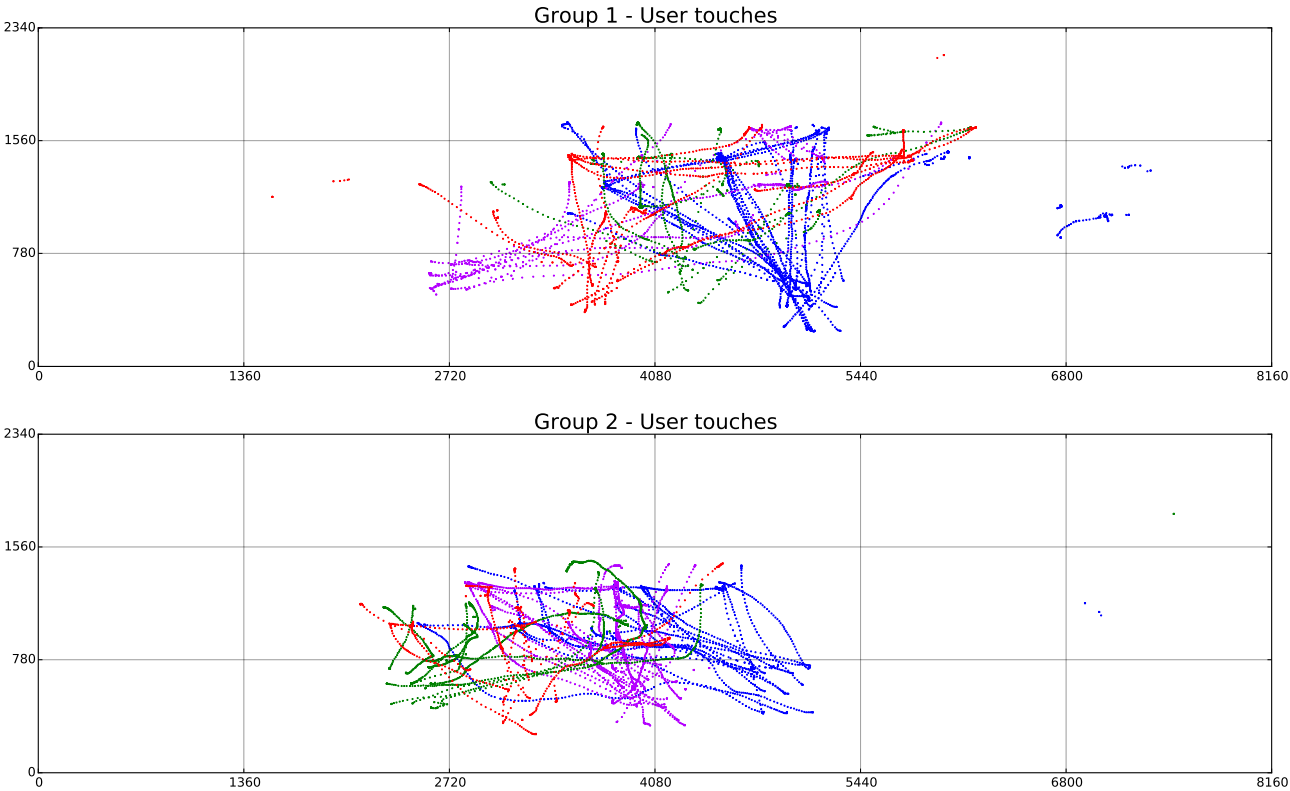
Figure 44: The two graphs shows each touch of each user during the scheduling session using the LVD. The color is representative of the user and they are the same used in Table VI. The eighteen rectangles represent the 6x3 displays composing our LVD.

participants used the display. The initial condition in which the environment was presented is shown in Figure 20. All the analyzed groups preferred to organize the environment in a condition similar to the one in Figure 45.



Figure 45: Participants preferred to organize the environment positioning the items in the central part of the LVD, leaving the lateral and bottom parts empty.

It is evident that in this condition presented by Figure 45, the lateral and the top parts of the display are rarely used during the task execution. Indeed, participants ended up touching approximately less than the 30% of the display surface. This is not a surprising result since earlier studies [24] have similar percentage of display usage. Figure 44 shows the touches that participants of G1 and G2 did. They are very similar to the others group touches since each group decided to place the table at the center of the screen. Moreover, we analyzed how touches were distributed within the eighteen displays composing our LVD. From the Figure 46, there are displays that are not even used. This information is compatible with Figure 45 since no virtual items are placed within the most lateral displays. These percentages are compatible with the percentage of display coverage with windows that is of only 22%.

### 5.7.10    Questionnaire Results

Questionnaires gave us important feedback on the Conference Scheduler application. From the acquired results and the recordings, we can state that the application did not contain any lack of functions or disadvantage that would have made the comparison unfair. Participants were satisfied by the application. We can identify it from Table III where all the questions related to the ease of use and the interface of the Conference Scheduler obtained a result higher than 8 on a scale of 10. Regarding

Touches distribution over the whole large display

| | | | | | |
|---|---|---|---|---|---|
| 0% | 0% | 1% | 2% | 1% | 0% |
| 0% | 3% | 38% | 25% | 3% | 1% |
| 0% | 3% | 11% | 11% | 0% | 0% |

Figure 46: The figure represents the percentage of usage of each display by all the groups. Each rectangle represents a display of our LVD.

the visualized information by the interface, there were frequent suggestions to make collapsible rooms more evident and to have the constraints on the display. The second request might be added easily to the application, even though in a real context the constraints belong to the chair's knowledge and it is not known a priori. A frequent suggestion on the offered functions was to add the possibility to swap the post-it notes. We did not provide this possibility to avoid errors due to wrong release. However, if a confirmation of the swapping action is requested, this error can be avoided.

The set of questions related to the comparison between the approaches (Table IV) still gave a positive result. Participants voted with an average of 2.8 (SD=0.5) on a scale between -5 and 5. No one gave a negative vote and all of them were distributed between 2 and 4. Participants preferred the display approach because it was clearer, more flexible, and customizable. They appreciated the possibility to resize and move the containers around the display however much and wherever they desired. One participant simply thought that performing this task with the LMVD was funnier and that it is an important variable for these 'boring problems'. All of them thought that technology can help to tackle manual problems and the main reasons are: technology does not give the space for errors, it gives more flexibility to perform the operation since an infinite number of functions can be installed, it is easier to undo and redo the participants' actions, it is easier to maintain order in the environment using technology, and it is possible to eliminate or reduce set up times and the usage of paper.

We must do a clarification on these outcomes. The strong preference of the display approach and the high results obtained by its characteristics might be biased by the prevalence on the population of individuals familiar to the computer field. This implies that we cannot state too general conclusions

under these aspects. For example, some people are skeptic on the employment of the technology on solving problems that were solved manually in the past, and they generally refuse a priori the new approaches. In fact, even though participants preferred the technological approach, this did not confer any advantage to them since results are similar in all the cases. Instead, what we want to point out is that people not expert in the conference-scheduling field found the application interface enough complete in terms of functions and informativeness. One of our goal was to produce an application that did not negatively influence the task outcome.

There was only one group ($G_2$) that expressed unanimously the presence of a leader for the traditional approach. In that situation, $P1$ was voted by all the other participants as leader, and we can notice from Table VI that he obtained the highest score. For the other groups using the traditional approach, no one expressed the presence of a leader. However, it happened for four groups that someone expressed this presence using the LMVD. In $G_{1,display}$ no one expressed the presence of a leader, probably because of the very short duration of the second phase (1m13s). In $G_{2,display}$ and $G_{4,display}$ participants voted unanimously for a leader that then resulted to be the winner. In the other two study runs with the display, no participant obtained more than one vote. There are two interesting tendencies to notice. The first one is that participants found the presence of someone to be a leader more in the display approach. The second one is that in all the occasions a leader was elected, he obtained a result higher than 38 (Mean=35.5, Median=36, Mode=35). Even though there are not enough data to run a statistical test, it is still interesting to notice these two tendencies.

## 5.8   Comparison with other studies

It is not easy to make a fair comparison with the other studies because no studies on LMVDs were done in multi-user environments considering a competitive variable. However, we want to analyze the similar and the divergent results of some studies on LVDs with ours. The most similar studies were conducted by Birnholtz et al. [49] ($S_1$) and by Jakobsen and Hornbæk [24] ($S_2$). The former used a 3x6 LVD without touch capabilities letting users interact with single and multiple mice. The latter used a 3x4 LMVD but in a collaborative environment. We talked about these two studies in the related work chapter. In $S_2$ are posed some interesting open questions on LVDs and we will try to give our contribution to them.

In $S_2$, users were free to move around the room, like in our experiment. Nevertheless, there is a

substantial difference between the position occupied by the users. In their study, users stayed 91% of the time to a touching distance to the display. Instead, in our study the participants occupied this distance for only the 15% of the time. This is obviously caused by the diversity of the two tasks. Even though no rules about the user position were imposed, the first phase was turn-based, and the participants preferred to stay far from the display when it was not their turn. We also think that users preferred to maintain this distance to have a better overview of the task evolution. This induction might be an answer to an open question posed by $S_2$ related to understanding the benefit of backing away from the LVD. Indeed, we observed how participants preferred to stay far from the display to have a clear overview of the situation. It was not clear if users benefited by this action since we did not focus on it, but there is this tendency to backing away from the display by the participants.

$S_2$ observed that participants shared evenly the display, even though no explicit negotiation for the space was performed. The same happened in our study. This is an interesting observation since in our study was present a competitive variable. Despite its presence, no participant voluntarily impeded the usage by another user. In the case another participant needed to use an occupied part of the display, he waited for some seconds and then used it without any impact on his performance caused by this little delay. It never happened that a free slot in the negotiation phase was needed by more than one person. It would be interesting to see what would have happened between the two users, in this case.

In the collaborative task of $S_2$ participants used simultaneously the touch display. In our study, there were only rare situations in which users acted simultaneously. Participants were influenced by the presence of a main central shared item (the scheduling table) in which they were able to act. Indeed, if another participant was using the display, the others preferred to wait the end of his interaction, because his presence probably would have prevented or made tougher the action to perform.

If we divide the analysis of our task in the two phases, we can find some similar results with $S_2$ for the visual attention. Indeed, there users looked generally at the display and rarely at each other. It happened also in our first phase that users preferred to look at the display rather than at each other, even in the occasion a single user was performing the action of his turn on the display. Instead, in the second phase this tendency is completely overturned since users mostly looked at each other.

Display usage is another similarity with $S_2$. In our study the users interacted less with the display and once the have decided the containers position, no changes were performed on the windows dispo-

sition. Furthermore, our LVD was bigger and composed of two more column of displays. Then, the display was slightly positioned higher on the wall, making the use of the last row easier, but the use of the top row harder. That is the opposite of what happened in $S_2$. In both studies there was a low tendency of covering the display with windows with an average percentage lower than 50%. Since the most lateral displays were not used in our study and the top row were rarely used, our percentage of display covering was very low (22%). If we do not count at least the most lateral displays, this percentage grows up to 33%, that was the minimum percentage of display covering in $S_2$ (the maximum is 58%). Unfortunately, we do not know this percentage for $S_1$, but analyzing the figure of user activity there is a clear similarity with Figure 44. Indeed, in both studies there is the presence of a central shared item and most of the interactions are concentrated in that location.

We can do an interesting comparison between the competition in $S_1$ and ours. Three participants were tested on the same task using a single mouse first and three mice then in $S_1$. Contrarily to their hypothesis, the second condition was the one where participants acted more for themselves. Moreover, in this condition participants talked less. In our study, we think that participants felt the competition more in the first phase. Indeed, in the second phase they collaborated more trying to help each other on finding an admissible schedule. We observed that with the increase of the time, participants forgot about the competition and collaborated more. In the first phase, participants felt more the competition and talked less, thinking more to their own interests and how to maximize their scores. Moreover, people talked much in the occasion of negotiations in both studies.

# CHAPTER 6

## CONCLUSION

This thesis wanted to give a contribute to the literature of the human-computer field for LMVDs. It proposed a simple technique to develop an interface for a LMVD. The same technique was used to develop the Conference Scheduler application, a support tool for the generation of a schedule for a scientific conference. The tool was used in a user study to analyze two aspects:

- the former was the analysis of the human behavior in a multi-user collaborative/competitive environment;

- the latter was a comparison of the efficiency between a LMVD and the traditional approach on solving the same task.

We precisely analyzed groups of four people performing a scheduling session for a scientific conference. The groups' goal was to find an admissible schedule that satisfied the everyones constraints and that maximized the schedule quality. At the same time, each participant had the personal goal of maximizing his personal score; this represented the competitive variable within the environment.

We described and analyzed the human behavior performing this task under the aspects of verbal communication, visual attention, group shape, display proximity and usage. The data we gathered did not communicate any efficiency improvement in the usage of the technological approach compared to the traditional one, neither in the quality of the results, nor in the consumed time, nor in the number of complaints made by the participants. Still, we did not identified any decrease in the quality of all these measures. However, we found that participants preferred the usage of the Conference Scheduler application for several reasons, by analyzing the questionnaires outcome. Nevertheless, these outcomes might be influenced by the presence of individuals mostly belonging to a computer science field in the analyzed subject pools. Another limitation might be the absence of expert individuals on performing a scheduling of a conference. This means that users were influenced by the inexperience on the problem, and on the inherent learning process. However, this effect may have been limited by the fact that everyone was inexpert of the field. We also want to point out that the analyzed measures, especially the approach efficiency, may be influenced by the limited number of topics, papers, and slots within

our mock conference. Actually, a real conference has a wider domain for each variables. This brings on a longer task and the effects on the analyzed measured may be different.

The particularity of having a competitive variable within the environment is not a novelty in the research of LVDs (eg. [49]), but it is for the one of LMVDs. Indeed, no study on LMVDs in multi-user environments considered a competitive variable, but only collaboration among participants (eg. [24]). For this reason, it was not simple to make a comparison among the results of others studies. Nevertheless, this comparison was a necessary step to add significance to our findings. Indeed, as Knudsen et al. [3] stated in their article, a typical error is to make general conclusions by studying users performing a very specialized task. This brought to have inconsistent literature for large displays. A sad consequence is that some studies on the same environment state divergent conclusions. The only way to create a solid literature is by finding the reasons of these differences and adding them to their own conclusions.

This is what we did. With our study we do not want to impose our conclusions for all the variety of competitive tasks. Instead, we wanted to give a contribute to the literature of the human-computer interaction field providing statistically relevant conclusions on a specialized task and making comparisons with similar studies. We also hope that in future studies, our conclusion can be merged, confirmed or also denied by their conclusions. In general, since large displays are starting to spread in more realities, more studies need to be conducted in order to produce the missing large display paradigm that Moreland discuss about in his work [10].

Future work should be done to make more consistent our findings. First, more participants need to be involved in order to have more data to perform further analysis and to support our conclusions. To deeply understand the human behavior in such environment, a study exclusively oriented toward this subject should be conducted, neglecting the efficiency analysis of the employed approach. To have a more display-interactive study, it is possible to integrate the material given to the users in the Conference Scheduler interface. It would be interesting to analyze the participants behavior in this new condition and the differences with our study.

# CITED LITERATURE

1. Rogers, Y. and Lindley, S.: Collaborating around vertical and horizontal large interactive displays: which way is best? Interacting with Computers, 16(6):1133–1152, 2004.

2. Ball, R. and North, C.: Effects of tiled high-resolution display on basic visualization and navigation tasks. In CHI'05 extended abstracts on Human factors in computing systems, pages 1196–1199. ACM, 2005.

3. Knudsen, S., Jakobsen, M. R., and Hornbæk, K.: An exploratory study of how abundant display space may support data analysis. In Proceedings of the 7th Nordic Conference on Human-Computer Interaction: Making Sense Through Design, pages 558–567. ACM, 2012.

4. Bi, X. and Balakrishnan, R.: Comparing usage of a large high-resolution display to single or dual desktop displays for daily work. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pages 1005–1014. ACM, 2009.

5. Agostini, A. and Di Biase, E.: Large multitouch screens to enhance collaboration in the classroom of the 21st century: an italian experiment. IxD&A, 15:40–56, 2012.

6. Marrinan, T., Aurisano, J., Nishimoto, A., Bharadwaj, K., Mateevitsi, V., Renambot, L., Long, L., Johnson, A., and Leigh, J.: Sage2: A new approach for data intensive collaboration using scalable resolution shared displays. In Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2014 International Conference on, pages 177–186. IEEE, 2014.

7. Jacucci, G., Morrison, A., Richard, G. T., Kleimola, J., Peltonen, P., Parisi, L., and Laitinen, T.: Worlds of information: designing for engagement at a public multi-touch display. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pages 2267–2276. ACM, 2010.

8. Elrod, S., Bruce, R., Gold, R., Goldberg, D., Halasz, F., Janssen, W., Lee, D., McCall, K., Pedersen, E., Pier, K., et al.: Liveboard: a large interactive display supporting group meetings, presentations, and remote collaboration. In Proceedings of the SIGCHI conference on Human factors in computing systems, pages 599–607. ACM, 1992.

9. Streitz, N. A., Geißler, J., Holmer, T., Konomi, S., Müller-Tomfelde, C., Reischl, W., Rexroth, P., Seitz, P., and Steinmetz, R.: i-land: an interactive landscape for creativity and innovation. In Proceedings of the SIGCHI conference on Human Factors in Computing Systems, pages 120–127. ACM, 1999.

10. Moreland, K.: Redirecting research in large-format displays for visualization. In Large Data Analysis and Visualization (LDAV), 2012 IEEE Symposium on, pages 91–95. IEEE, 2012.

11. Andrews, C., Endert, A., Yost, B., and North, C.: Information visualization on large, high-resolution displays: Issues, challenges, and opportunities. Information Visualization, page 1473871611415997, 2011.

12. Heermann, P. D.: First-generation asci production visualization environment. Computer Graphics and Applications, IEEE, 19(5):66–71, 1999.

**CITED LITERATURE (continued)**

13. Smith, P. H. et al.: Data and visualization corridors: Report on the 1998 dvc workshop series. Technical report, California Institute of Technology, Center for Advanced Computing Research, Pasadena, CA (US), 1998.

14. De Almeida, R. A., Pillias, C., Pietriga, E., and Cubaud, P.: Looking behind bezels: French windows for wall displays. In Proceedings of the International Working Conference on Advanced Visual Interfaces, pages 124–131. ACM, 2012.

15. Tan, D. S. and Czerwinski, M.: Effects of visual separation and physical discontinuities when distributing information across multiple displays. In Proc. Interact, volume 3, pages 252–255, 2003.

16. Grudin, J.: Partitioning digital worlds: focal and peripheral awareness in multiple monitor use. In Proceedings of the SIGCHI conference on Human factors in computing systems, pages 458–465. ACM, 2001.

17. Han, J. Y.: Low-cost multi-touch sensing through frustrated total internal reflection. In Proceedings of the 18th annual ACM symposium on User interface software and technology, pages 115–118. ACM, 2005.

18. Bezerianos, A. and Isenberg, P.: Perception of visual variables on tiled wall-sized displays for information visualization applications. Visualization and Computer Graphics, IEEE Transactions on, 18(12):2516–2525, 2012.

19. Ragan, E. D., Endert, A., Bowman, D. A., and Quek, F.: How spatial layout, interactivity, and persistent visibility affect learning with large displays. In Proceedings of the International Working Conference on Advanced Visual Interfaces, pages 91–98. ACM, 2012.

20. Prensky, M.: Digital natives, digital immigrants part 1. On the horizon, 9(5):1–6, 2001.

21. Kruger, R., Carpendale, S., Scott, S. D., and Greenberg, S.: Roles of orientation in tabletop collaboration: Comprehension, coordination and communication. Computer Supported Cooperative Work (CSCW), 13(5-6):501–537, 2004.

22. Scott, S. D., Carpendale, M. S. T., and Inkpen, K. M.: Territoriality in collaborative tabletop workspaces. In Proceedings of the 2004 ACM conference on Computer supported cooperative work, pages 294–303. ACM, 2004.

23. Czerwinski, M., Smith, G., Regan, T., Meyers, B., Robertson, G., and Starkweather, G.: Toward characterizing the productivity benefits of very large displays. In Proceedings of INTERACT, volume 3, pages 9–16, 2003.

24. Jakobsen, M. R. and Hornbæk, K.: Up close and personal: Collaborative work on a high-resolution multitouch wall display. ACM Transactions on Computer-Human Interaction (TOCHI), 21(2):11, 2014.

25. Michael, R. G. and David, S. J.: Computers and intractability: a guide to the theory of np-completeness. WH Free. Co., San Fr, 1979.

26. Sajadi, B., Lazarov, M., Gopi, M., and Majumder, A.: Color seamlessness in multi-projector displays using constrained gamut morphing. Visualization and Computer Graphics, IEEE Transactions on, 15(6):1317–1326, 2009.

**CITED LITERATURE (continued)**

27. Sajadi, B. and Majumder, A.: Markerless view-independent registration of multiple distorted projectors on extruded surfaces using an uncalibrated camera. Visualization and Computer Graphics, IEEE Transactions on, 15(6):1307–1316, 2009.

28. Baudisch, P., DeCarlo, D., Duchowski, A. T., and Geisler, W. S.: Focusing on the essential: considering attention in display design. Communications of the ACM, 46(3):60–66, 2003.

29. Rantanen, E. M. and Goldberg, J. H.: The effect of mental workload on the visual field size and shape. Ergonomics, 42(6):816–834, 1999.

30. Simons, D. J. and Chabris, C. F.: Gorillas in our midst: Sustained inattentional blindness for dynamic events. Perception, 28(9):1059–1074, 1999.

31. Robertson, G., Czerwinski, M., Baudisch, P., Meyers, B., Robbins, D., Smith, G., and Tan, D.: The large-display user experience. Computer Graphics and Applications, IEEE, 25(4):44–51, 2005.

32. Czerwinski, M., Robertson, G., Meyers, B., Smith, G., Robbins, D., and Tan, D.: Large display research overview. In CHI'06 extended abstracts on Human factors in computing systems, pages 69–74. ACM, 2006.

33. Benko, H., Wilson, A. D., and Baudisch, P.: Precise selection techniques for multi-touch screens. In Proceedings of the SIGCHI conference on Human Factors in computing systems, pages 1263–1272. ACM, 2006.

34. Morrison, G. D.: A camera-based input device for large interactive displays. Computer Graphics and Applications, IEEE, 25(4):52–57, 2005.

35. Guimbretière, F., Stone, M., and Winograd, T.: Fluid interaction with high-resolution wall-size displays. In Proceedings of the 14th annual ACM symposium on User interface software and technology, pages 21–30. ACM, 2001.

36. Cao, X. and Balakrishnan, R.: Visionwand: interaction techniques for large displays using a passive wand tracked in 3d. In Proceedings of the 16th annual ACM symposium on User interface software and technology, pages 173–182. ACM, 2003.

37. Myers, B. A., Bhatnagar, R., Nichols, J., Peck, C. H., Kong, D., Miller, R., and Long, A. C.: Interacting at a distance: measuring the performance of laser pointers and other devices. In Proceedings of the SIGCHI conference on Human factors in computing systems, pages 33–40. ACM, 2002.

38. Nishimoto, A.: Multi-user interface for scalable resolution touch walls. 2014.

39. Ball, R., Varghese, M., Sabri, A., Cox, E. D., Fierer, C., Peterson, M., Carstensen, B., and North, C.: Evaluating the benefits of tiled displays for navigating maps. In IASTED International Conference on Human-Computer Interaction, pages 66–71, 2005.

40. Gouin, D. and Evdokiou, P.: A showcase of visualization approaches for military decision makers. Technical report, DTIC Document, 2004.

41. Oonk, H., Smallman, H., Moore, R., and Morrison, J.: Usage, utility, and usability of the knowledge wall during the global 2000 war game. Technical report, DTIC Document, 2001.

**CITED LITERATURE (continued)**

42. Tan, D. S., Robertson, G. G., and Czerwinski, M.: Exploring 3d navigation: combining speed-coupled flying with orbiting. In Proceedings of the SIGCHI conference on Human factors in computing systems, pages 418–425. ACM, 2001.

43. Tan, D. S., Czerwinski, M. P., and Robertson, G. G.: Large displays enhance optical flow cues and narrow the gender gap in 3-d virtual navigation. Human Factors: The Journal of the Human Factors and Ergonomics Society, 48(2):318–333, 2006.

44. Patrick, E., Cosgrove, D., Slavkovic, A., Rode, J. A., Verratti, T., and Chiselko, G.: Using a large projection screen as an alternative to head-mounted displays for virtual environments. In Proceedings of the SIGCHI conference on Human Factors in Computing Systems, pages 478–485. ACM, 2000.

45. Kin, K., Agrawala, M., and DeRose, T.: Determining the benefits of direct-touch, bimanual, and multifinger input on a multitouch workstation. In Proceedings of Graphics interface 2009, pages 119–124. Canadian Information Processing Society, 2009.

46. Bradel, L., Andrews, C., Endert, A., Vogt, K., Hutchings, D., and North, C.: Space for two to think: Large, high-resolution displays for co-located collaborative sensemaking. 2011.

47. Grinstein, G., Connell, T. O., Laskowski, S., Plaisant, C., Scholtz, J., and Whiting, M.: Vast 2006 contest-a tale of alderwood. In Visual Analytics Science And Technology, 2006 IEEE Symposium On, pages 215–216. IEEE, 2006.

48. Andrews, C., Endert, A., and North, C.: Space to think: large high-resolution displays for sensemaking. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pages 55–64. ACM, 2010.

49. Birnholtz, J. P., Grossman, T., Mak, C., and Balakrishnan, R.: An exploratory study of input configuration and group process in a negotiation task using a large display. In Proceedings of the SIGCHI conference on Human factors in computing systems, pages 91–100. ACM, 2007.

50. Bradel, L., Endert, A., Koch, K., Andrews, C., and North, C.: Large high resolution displays for co-located collaborative sensemaking: Display usage and territoriality. International Journal of Human-Computer Studies, 71(11):1078–1088, 2013.

51. Vogt, K., Bradel, L., Andrews, C., North, C., Endert, A., and Hutchings, D.: Co-located collaborative sensemaking on a large high-resolution display with multiple input devices. In Human-Computer Interaction–INTERACT 2011, pages 589–604. Springer, 2011.

52. Park, K. S., Leigh, J., Johnson, A. E., and Cho, Y.: Design of awareness interface for distributed teams in display-rich advanced collaboration environments. In Groupware: Design, Implementation, and Use, pages 105–120. Springer, 2004.

53. Isenberg, P., Fisher, D., Paul, S. A., Morris, M. R., Inkpen, K., and Czerwinski, M.: Co-located collaborative visual analytics around a tabletop display. Visualization and Computer Graphics, IEEE Transactions on, 18(5):689–702, 2012.

54. Tang, A., Tory, M., Po, B., Neumann, P., and Carpendale, S.: Collaborative coupling over tabletop displays. In Proceedings of the SIGCHI conference on Human Factors in computing systems, pages 1181–1190. ACM, 2006.

## CITED LITERATURE (continued)

55. Cohen, J.: Statistical power analysis for the behavioral sciences. 2nd edn. hillsdale, new jersey: L, 1988.

# VITA

| NAME | Davide Tantillo |
|------|-----------------|
| **EDUCATION** | |
| | M.S., Computer Science, University of Illinois at Chicago, May 2016, USA |
| | B.S., Computer Science and Engineering at Politecnico di Milano, 2014, Italy |
| **SCHOLARSHIPS** | |
| Spring 2016 | Research Assistantship (RA) position at Electronic Visualization Lab (University of Illinois at Chicago) |
| **LANGUAGE SKILLS** | |
| Italian | Native speaker |
| English | Written: Good – Spoken: Good |