VI-TRUST: Visualization and Interaction on project "Transitions to the Urban Water Services of Tomorrow"

Extended Abstract

João Oliveira, no. 67013
Instituto Superior Técnico
Avenida Rovisco Pais, 1
Lisboa, Portugal
joao.delgado.oliveira@tecnico.ulisboa.pt

ABSTRACT

Collaborative software has been explored for a long time, but still there are obstacles to interaction that technology has been unable to overcome. In some situations, people discard face-to-face meeting software and go back to using non-software tools.

We hypothesize that one of the reasons for this to happen is that systems using only traditional means of interaction cannot capture natural means of communication between people, among which are spatial relationships. The study of this type of communication is known as proxemics and we propose using them for improving face-to-face meeting software. In our system we allow users to present and highlight their work simply by moving in the room. This movement is something that people do anyway, even when not using software – to present work, users move to the front of the room, where they can easily capture the focus of everyone. When someone moves in that position, all of the people in the room understand what that means, but computers must be informed with explicit input from a user.

With a system that is aware of the spatial relationships, this doesn't happen – the system adapts as users change position, requiring no other input. To evaluate the impact of proxemics, we built two systems with the exact same functionality and add proxemics to one of them. We tested both systems with users and our results show that proxemics leads to less explicit interactions, which in turn yields better user satisfaction.

Keywords

proxemics, CSCW, natural interfaces, HCI

1. INTRODUCTION

In this section we begin by detailing the motivation behind our work in subsection 1.1. We then explain how that motivation leads to the main challenge for this dissertation and what is our main research goal in subsection 1.2. Through the achievement of that goal we can provide two important contributions, which are discussed in subsection 1.3. Finally, we present the remaining document structure in subsection 1.4.

1.1 Motivation

Software for supporting human collaboration has been in use for a long time, but in face-to-face meetings, people are still feeling limited by computers. The truth is that humans communicate through means that are not typically perceived by machines, in particular spatial relations. Even though we don't realize it consciously, we interact with each other through very simple spatial cues that everyone understands.

Because the software we use is not capable of perceiving this implicit communication, it forces users to inform the system through explicit interactions. This results in people feeling like the computer is in their way, instead of supporting their tasks. As such, there is an added protocol present in the software that wasn't there before. Because of this, users tend to ignore the system, using older methods and not effectively adopting the solutions developed for them.

1.2 Objectives

The challenge for this dissertation is to address the issue identified in the previous subsection for a particular scenario - a face-to-face meeting targeted at raising new ideas and making decisions for the TRUST European project - Transitions to the Urban Water Services of Tomorrow[1].

Early in this document we identify the recurring problems of applications with similar purposes - commonly referenced as CSCW (computer supported cooperative work). We speculate that these problems can be solved through the adequate application of proxemics in the design and implementation of CSCW software. Proxemics is the study of the use of space in human communication and recently it has been increasingly adopted in human-computer interaction.

Our main research goal is to understand whether proxemics can effectively reduce the number of explicit interactions (mouse clicks, taps, gestures) with the computer. We hypothesize that through proxemics it is possible to communicate more information with less explicit interactions. With the ability to understand the use of space implemented, users are required to spend less effort interacting and there is a reduction in the protocol for collaboration.

1.3 Contributions

The main contribution of this work is understanding the impact that proxemics can have in CSCW. We have developed a solution for a particular scenario but our work can be reproduced for similar situations. Through the comparison of two versions of our system whose only difference was that one took advantage of proxemics, we have observed the

[1]Project Reference: 265122; Website: http://www.trust-i.net/

differences in how the software is used and how that affects user satisfaction. We have then discussed those differences and how they address the recurring issues of CSCW applications.

Because the scope is limited to a face-to-face meeting we have been able to accurately identify where using proxemics makes sense. We have experimented with several alternatives for the same function and have evaluated user feedback on each of them. This discussion on why certain alternatives work or not provides an important basis for other people designing CSCW applications aware of proxemics information.

1.4 Document Structure

In the remaining document we present how we understood our challenge, how we approached it and how we evaluated our contributions. First we discuss the related work in section 2, where we explore the recurring problems of CSCW and how using proxemics might provide a solution to those. In section 3 we explain how we approached those problems in our scenario and how other works on proxemics have influenced our design. The impact of proxemics was evaluated in section 4, which explains how we compared two versions of our software - one with and other without proxemics Lastly, we conclude in section 5, presenting a final discussion on our work and on what could be done in the future.

2. RELATED WORK

Building a natural interaction for a collaborative software requires knowledge in both CSCW and Human Interaction - be it with other humans or with machines. In the following CSCW subsection the scope of the project will be analyzed in the context of CSCW. By analyzing previous work in the same scope it is possible to understand what problems remain unsolved. The Proxemics subsection is a review of work on proxemics in isolation - understanding how humans use space to communicate with each other. We further explore Proxemics applied to HCI reviewing work by other researchers that have used proxemics in their studies. In this subsection, a set of works is highlighted as an inspiration for what can be done with proxemics. Lastly, we present a brief discussion on all of these topics.

2.1 CSCW

In order to define the scope of the work to be produced, it is important to identify where it fits in the CSCW matrix (Figure 1). Due to a preliminary analysis of the users' needs with our partners it was decided to focus only on the "same place, same time" quadrant.

The key characteristic of this quadrant is that users already have an established means of communication much more effective than anything that can be done with current technology - their presence. As such, the application should be seen as a way to mediate and support the communication, without ever getting in its way. This observation becomes even more relevant when analyzed together with Grudin's work on why CSCW applications fail (Grudin [6]). Grudin identifies three problems that arise from a lack of understanding of how people perform their work: the disparity between who does the work and who gets the benefit; the breakdown of intuitive decision making; and the underestimated difficulty of evaluating CSCW applications.

Dourish and Bellotti [4] further identified problems with traditional collaborative software in regards to awareness -

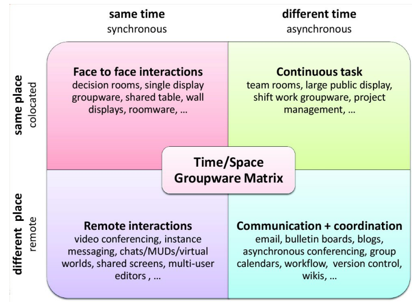

Figure 1: The CSCW matrix, defined by Johansen [8].

defined by the authors as "an understanding of the activities of others, which provides a context for your own activity". The authors then proceed to propose a new approach based on shared feedback. The main difference is that the information on others' activities is displayed in a shared workspace, not in each of the private ones.

2.2 Proxemics

Proxemics was introduced by Edward T. Hall as the "study of man's perception and use of space" (Hall [7]). In this text, Hall focused on the different ways one perceives and uses space based on one's culture. Perhaps one of his most fundamental conclusions is that there is no pattern that can be applied to every individual - different people don't react in the same manner to the same stimuli.

Hall [7] later developed the field of proxemics when he identified categories for the distance between two people. According to Hall, the social relationship between two individuals is directly related to physical distance, which means that it is possible to estimate the interactions that might emerge from the distance category at which people position themselves.

Another important contribution are F-Formations, identified by Kendon [11]. According to the author, people tend to organize themselves in spatial formations when interacting, creating a shared space where they can communicate. This has two main purposes: to allow the participation of every individual involved and to separate the group from other individuals not taking part in the interaction.

Based in the position and orientation of the group's participants in these spaces, Kendon identified the following formations: Vis-à-vis - participants are face to face (restricted to two participants); L-shape - the angle between participants is approximately 90°; Side-by-side - the angle between participants is approximately 0°; and Circle - participants are organized in a circle.

In our scenario, F-Formations are particularly important because we are interested in detecting when people want to share information. This happens when they are facing the audience, which is organized around a meeting table, defining an approximate Circle formation - everyone is involved in the discussion and there is a person in particular who is leading it.

2.3 Proxemics applied to HCI

One of the most influential works in this area is the one by Nicolai Marquardt and Saul Greenberg, applying the concepts of proxemics to human-computer interaction and ubiquitous computing. Instead of having devices oblivious to their surroundings, they were able to create a system that provides the devices with info on which users and other devices are around them. Nicolai's thesis provided three main contributions: a proxemics framework, the Proximity Toolkit and three case studies.

The proxemics framework is targeted at designers building interaction around proxemics. The author defined five key dimensions to consider when designing - distance, orientation, movement, identity and location. He also identified a common strategy when building this kind of interactions - the gradual engaging pattern -, which defines how a system should react to user proxmity, moving between awareness, reveal and interaction.

The Proximity Toolkit provides a service for detecting proxemic relationships between devices and/or users. With this toolkit it becomes very simple to develop proxemic interactions - there is only the need to subscribe to events provided by the service and react to them accordingly.

The three case studies are explorations of the design of systems using proxemics. Because of the novelty of this approach, these studies provide valuable information.

Another example that takes advantage of proxemics is Range, developed by Ju et al. [9]. Range provides an interactive white-board that allows different interactions depending on the distance of the users. The distances and corresponding interactions were derived from work by Hall [7] and are in accordance with the gradual engaging pattern identified by Greenberg et al. [5].The main difference between Range and the work in this thesis is that instead of using a screen as a white-board, we use it as a support for presentation and communication of the current stage of the session.

Also on the subject of white-board based applications, work by Rekimoto [12] on using hand-held computers together with a shared screen to improve the interaction is extremely relevant. Rekimoto first identifies the design problems of white-board only approaches like inputing text, getting personal data and multitasking. He then shows how these problems can be addressed with the help of a hand-held computer, producing several solutions. Considering this work was produced in 1998, before the rise of the smartphone and the tablet, taking advantage of current technology should allow for even more solutions using multitouch screens together with mobile devices.

For understanding how to support different user formations in the room, the work by Cristani et al. [2] is an important reference. The authors were able to detect F-formations using unobtrusive computer vision techniques and concluded that the results of their system were consistent with proxemics findings. This work is particularly interesting for learning how to detect F-formations, which are required for adapting the interaction to moments when groups of users are together.

Designing the interactions can never be completely separated from the design of the place where these interactions take place. In building a meeting room system, special care must be taken in how the room is arranged. The book by Sommer [13], "Personal Space: The Behavioral Basis of Design", provides a means to take advantage of spatial arrangement, affecting the user's behavior towards interaction. In our case we took advantage of a long table and a space near the screen to encourage users to present their work.

2.4 Summary

From the analysis of CSCW work, it is clear that this software fits in the "same place, same time" quadrant of the CSCW matrix (Johansen [8]). Understanding how people work in this situation allows us to know how to address the reasons why CSCW applications fail (Grudin [6]): the disparity between who does the work and who gets the benefit; the breakdown of intuitive decision making; and the underestimated difficulty of evaluating CSCW applications. Being a "same place, same time" system means that it must also deal with problems related to awareness, as identified by Dourish and Bellotti [4].

Understanding proxemics makes it possible to address some of these concerns, namely the breakdown of intuitive decision making and providing awareness. In general terms, it should allow for a more 'natural' interaction.

In our scenario we take advantage of the moment when people approach the shared screen for presenting their work. Therefore we are interested in changing visualization depending on where people are relative to the screen. Therefore we aim to identify at what distance to the shared screen a participant becomes a speaker depending on cultural backgrounds and mostly on the physical disposition of the room. According to research on F-Formations, approaching the screen is not enough - we should also consider whether the person is facing the audience or not.

Applying proxemics to HCI is relatively new and has recently received a great improvement with Marquardt's contribution.This and other works on similar applications indicate that interactive systems benefit from proxemic interactions.

What is not so clear is the benefit of this type of interactions in real applications - in our case a face-to-face meeting. Which interactions make sense in this scenario? How much of the interaction should be done through more traditional methods or through proxemics? And how can we measure the impact of proxemics on the way users perceive the system?

In the following section we explain our approach to provide an answer to these questions.

3. USING PROXEMICS TO IMPROVE COLLABORATION

From the analysis of related work it is clear that CSCW applications have a set of recurring problems and proxemics might provide solutions. In the previous section we identified four main problems with CSCW applications (based on work by Grudin [6] and Dourish and Bellotti [4]): 1. the disparity between who does the work and who gets the benefit; 2. the breakdown of intuitive decision making; 3. the underestimated difficulty of evaluating CSCW applications; and 4. dealing with awareness.

We consider that two out of these four problems can be addressed by developing a system around proxemics: the breakdown of intuitive decision making and providing awareness.

The breakdown of intuitive decision making happens because CSCW software adds protocol to tasks that can be carried out in simpler terms using traditional resources. By having a system that detects the spatial relations between people and devices, it is possible to take advantage of how people interact with each other - by changing the state of the system at the right moment we are able to simplify the protocol.

Improving awareness is also something that can and has been done in other proxemics solutions. It is precisely one of the main advantages of using the gradual engaging pattern, as identified by Marquardt and as put into practice in the Range white-board. Because the system reacts to the user's use of space, there is an opportunity for a better control of what the user interacts with. This directly improves awareness of the system in regards to people, which in turn allows people to be more aware of what is happening in the meeting - the system is always showing information correctly adapted to the situation, therefore people can easily be informed just by checking the shared and/or their own screen.

In this section we discuss our approach to developing a face-to-face meeting software that targets the CSCW problems through the use of proxemics. In subsection 3.1 we discuss the context of this project - what scenario we must support and what work has been done previously. We then move on to briefly explain how our solution supports this scenario without proxemics in subsection 3.2. We also discuss the software architecture of our solution in subsection 3.3, which leads to how we equipped the system with proxemics, presented in subsections 3.4 and 3.5.

3.1 Context

Before the beginning of the development process, the proposal was to create an immersive environment designed for collaboration among multiple decision-makers / stakeholders. It had two clearly defined goals: to design a decision theater to support collaboration among decision-makers and stakeholders; and to explore new paradigms of natural user interfaces in order to avoid the traditional human-computer interaction limitations in these types of scenarios.

The meeting process was also provided in the beginning of the development by the partners of the project, who have been undertaking this sessions using white boards, pen and paper for a few years. The steps are heavily based on the work by Alegre and Coelho [1] on Infrastructure Asset Management (IAM) of Urban Water Systems:

1. Writing down the mission and vision.
2. Participants propose objectives and work on their text until a consensus is reached.
   (a) Each objective must be assigned to a TRUST dimension: Economic, Social, Environmental, Assets or Governance.
   (b) Each objective should be compared and validated with the objectives suggested by TRUST in 'A Master Framework for UWCS Sustainability'.
3. For each objective, participants propose criteria. At this point, they might conclude that the objectives need to be redefined. If so, the participants should return to step 2.
4. For each criterion, participants propose metrics. When doing this, users may use metrics contained in a metrics library. Again, they might realize that they should go back to step 2 or 3.
5. Every metric should be assigned a relative weight of 0.5, 0.75, 1, 1.5 or 2. The weight of a criterion is the sum of the weight of its metrics. In the same way, the weight of an objective is the sum of the weight of its criteria.
6. Once the participants are satisfied, they export their work to a file that can be used in further stages of the planning process.

3.2 Base System

Our goal is to understand whether we can use proxemics to improve CSCW software, particularly in our scenario. But that does not mean that all of the interactions must be based on proxemics. In fact, that is a problem of some of the work previously done in this area - it might be interesting to explore from an interaction point of view, but it is clear from a CSCW perspective that it would not be practical in this context.

Basing our final solution mostly in proven interaction methods assures that we produce a valid system - one that could be used by real companies. This in turn raises the value of our results, as it shows that using proxemics in the right situations introduces improvements in real applications, not just in hypothetical prototypes. This validity is also supported by the fact that we based our development process in user-centered guidelines - we first identified our users and incrementally developed prototypes with a higher level of fidelity until we reached the final version presented here.

Our solution consists of a distributed system based on a shared screen and personal devices (tablets and/or personal computers). This screen always shows the current focus of the meeting and it is controlled by a mediator through his own device (either a tablet or a personal computer). Other users can freely navigate the system and can submit their contributions at different moments during the meeting. Every person can see everyone else's contribution, but only the author and the mediator can change it. Every contribution is distributed through all the devices and the shared screen. This solution of having personal and shared spaces is based on work by Dourish and Bellotti [4] and Rekimoto [12].

The shared screen process must be the first to start. All users are then able to join the session by introducing their names. As soon as the mediator joins, he is able to edit the mission and vision which, as mentioned in the previous subsection, should be defined before the meeting.

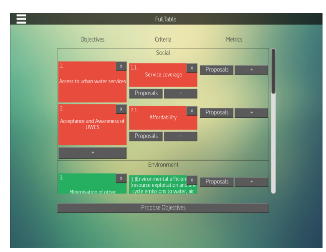

Figure 2: Screenshot of the table of objectives, criteria and metrics.

The main goal of the session is to fill the table of objectives, criteria and metrics (Figure 2). This table is filled by entering proposal mode, where users can submit their proposals and alternatives. Once all users agree on the goals, criteria and metrics, the next step is to assign a relative weight to each metric. To visualize the impact of the assignment, we have introduced a dynamic circular chart. Each

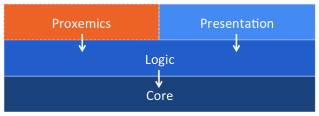

Figure 3: The software architecture.

user is able to make a personal weight distribution and it is up to the mediator to decide which one is better.

3.3 Software Architecture

All of these features are supported by the architecture illustrated in Figure 3. It is a layered architecture comprised of three main layers: Core, Logic and Presentation. A fourth optional layer is added at the same level as the presentation layer for dealing with proxemics.

The Core contains the main elements manipulated by the system. This includes the items (objectives, criteria and metrics) and the proposals and alternatives, which are a simplified version of items. The Logic layer is the one that controls the state and transitions in the application. It manipulates the elements of the Core according to the current state and it exposes controls for changing it. The Logic module also handles network synchronization - all devices are aware of a global state and each of them also have a local state. The Presentation layer presents content according to the current state of the Logic layer and it is able to use the controls exposed by this layer to request state changes. Finally, the optional Proxemics layer interacts only with the Logic layer by interacting with only a subset of controls - the ones that control visual adaptation according to proxemics information. It is important to note that these controls are also available to the Presentation layer, allowing access to the same functionality even without proxemics. This is what we use to be able to compare the two versions of the system.

3.4 Candidate Proxemics Interactions

We considered that two functionalities would be good candidates for the implementation of proxemic interactions: mediator definition and argument presentation. For defining the mediator we simply used proxemics information directly, but for the argument presentation we introduced the concept of a ”relevance” value.

If the system is in proposal mode, the relevance value adjusts the size of a user’s proposal - the higher the relevance, the bigger his contributions will appear. This behavior is illustrated in Figure 4. This change is visible in the shared screen and in every user’s device.

As for weight discussion, we have implemented a similar system - when users’ relevance is lower than a certain value their weight distribution is shown (Figure 5).

Considering these two features, several alternatives were imagined with the same group of experts as before (Tables 1 and 2).

Implementing all these alternatives would require a tremendous development effort, therefore we decided to test them using the Wizard of Oz method suggested by Kelley [10] - having users believe that the system was automatically detecting their position and movement, when in reality it was a developer who was controlling everything. Because no features were actually developed before testing, this method allowed us to explore a large amount of alternatives with no development effort.

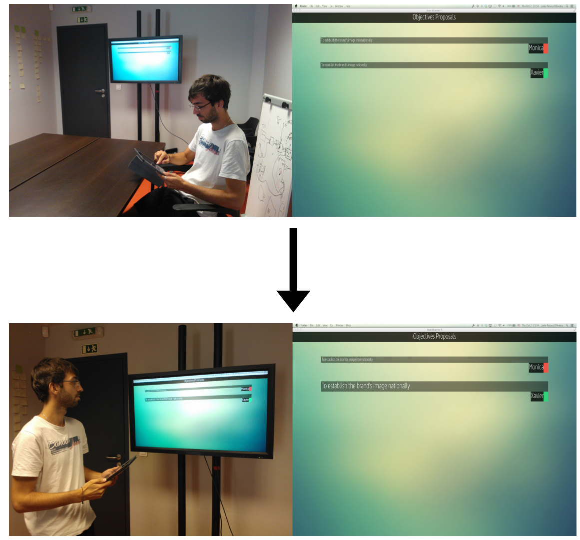

Figure 4: Visual adaptation in proposals screen.

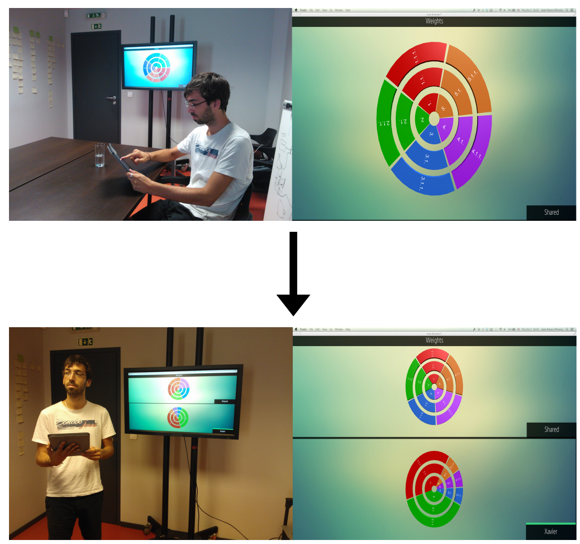

Figure 5: Visual adaptation in weights screen.

3.4.1 Preparation

We invited a total of thirty users to test our alternatives. They were divided in groups of three, allowing us to carry ten sessions. These users were mostly software engineers (66%) and most of them had a small amount of experience at the company - 86% of the users had been at the company for less than a year.

To achieve the illusion of a working system, a room was prepared with a Kinect device (which was turned on but not actually doing anything). Two users used a tablet each and the third user was given a laptop.

Each session took approximately one hour and thirty minutes and it had two parts, one for testing the mediator definition and another for testing the argument presentation. In the beginning of the session we did a brief (five minutes) presentation of the software and process explaining only the necessary features for these tests.

The Mediator Definition part began with handing out a

Screen	The mediator is the person closest to the screen
Standing	The mediator is the last person who stood up
Screen & Standing	The mediator is the person standing closest to the screen
Arm	The mediator is the person who last raised his arm
Pointing	The current mediator points to the next mediator
Standard	The current mediator selects the next mediator using the software

Table 1: Different alternatives for usage of proxemics in mediator definition.

Screen	Relevance is proportional to distance to shared screen
Screen & Center	Same as 'Screen' but user must be facing the center of the room
Height	Relevance is proportional to the height of the body
Standard	Each user is able to change his relevance value using a slider in their device

Table 2: Different alternatives for usage of proxemics in argument presentation.

script to the users: *Once you become the mediator, select a screen from the menu and say out loud the word "done". If the current mediator definition method is triggered by the current mediator (pointing and standard), select the person on your right to be the next mediator. If the current mediator definition method is triggered by the users, wait for the person on your right to become the mediator. Repeat this process until the first mediator becomes mediator again.*

Each group followed these instructions a total of six times, one for each method, taking approximately thirty minutes. The first mediator was chosen randomly and the sequence of methods was also changed randomly for every session.

Once all methods had been tested, we asked the users to fill a questionnaire which was targeted at understanding which method was the best in their opinion.

For the Argument Presentation part we decided to limit the tests only to the presentation of arguments in the objective definition phase - it is the easier one to understand because it is the first to take place and it provides a good sample for all the remaining moments of discussion, because they are all similar.

We gave users a scenario they were familiar with, which was also used in the final tests: *Suppose that the group is a team responsible for creating a new YDreams spin-out, YWear, whose focus is to create wearable technology. In this session you are going to list and prioritize the goals to achieve in the next six months, assuming the spin-out is starting now.*

We also gave them a set of 20 objectives that we prepared with the partners. The idea was that the users would choose one objective and defend it. As a group they should then decide which objective was the most important and discard the others. By giving these objectives to the users, we were able to make our tests shorter in time because there was no need for users to come up with their own ideas, they could focus only on what was important for our tests: the discussion.

Each group did a total of four discussions, one for each argument definition method. Once an objective had been used by anyone, no one else could use it again. We allowed users to freely decide whether they would present their argument one at a time, two at a time, or even three at a time. We only set a maximum time limit of ten minutes. The order at which each method was tested was again randomized in different sessions.

For understanding the opinion of the users in regards to the tested argument definition methods, we gave them five minutes to fill a small questionnaire.

In order to make sure that there was no influence of the order in the results, we changed the order of the parts for each session: half began with the mediator definition tests and the remaining ones began with the argument presentation tests. Because the session required some time from the users, we also did a small five minute break in between parts.

3.4.2 Results

During the mediator definition method tests it was clear that users weren't feeling comfortable with a proxemics alternative to the standard solution. This inadequacy was even clearer as soon as the results of the questionnaire were analyzed.

The most popular proxemics alternative was the "Screen + Standing" one. Yet, only 6 users (20%) considered it useful, only 2 users (7%) stated they would use it often and again only 2 users stated it was very easy to use. The majority of the users considered all of the proxemics alternatives to be impractical. For every alternative, the number of users who stated they would never use it was always above 20 (67%), with a similar result for ease of use (very difficult to use was selected by more than 20 users in every proxemics alternative). It was unanimous that the best option was simply the standard one.

Regarding the argument presentation functionality, users were more receptive to the proxemics solutions. During the tests we observed that there were two alternatives that users were clearly appreciating: "Screen" and "Screen & Center". This observation was later visible in the results.

When asked if they would use each of the alternatives, there were actually more users replying they would use the Screen alternative (93%) or the Screen + Center (60%) alternative than those stating they would use the Standard alternative (43%). A similar number of users considered the same two alternatives to be easier to use than the standard one. It is relevant to note that users did consider the standard alternative to react better to their expectations, but even so they preferred using the Screen and Screen & Center alternatives (as supported by data discussed in the two previous items). This is an indicator that even though users felt they had a more accurate control using the standard alternative, the flow of the discussion benefited from using proxemics.

Overall, the Screen alternative performed better than the Screen + Center alternative. What happened often was that users were presenting their argument, facing the center of the session, but as soon as they would turn slightly, even if it was to show a detail on the screen, their relevance dropped, which was against what they would want from the system - while they are presenting, the users wanted the system to highlight their work, regardless of the fact that they are facing the center. This observation seems to contradict F-Formations, but we may consider that the fact that there is a screen with relevant information makes that screen part of the formation and, as a consequence, there is no need for the user who is presenting to be facing the audience only.

3.5 Equipping the System with Proxemics

From the analysis of the previous subsection we decided to implement the argument presentation using proxemics and leave the mediator definition implemented with traditional controls.

Improving argument presentation through proxemics is directly targeted at the CSCW problems we previously identified: awareness and the breakdown of intuitive decision making.

By constantly highlighting what is currently in discussion without any explicit input from the users, everyone in the room can easily recognize what is going on, just by looking at the shared screen or their own device. And this system is completely dynamic - maybe at some point only one user is presenting his work, but as soon as another decides to intervene and get up to expose his ideas, the visualization automatically adapts to this change. Even if someone is completely distracted, it is easy to get back in the discussion because everything is highlighted automatically and in real time.

As for decision making, it benefits from our solution because subjects spend less effort interacting with the system - they are simply discussing like they would do without any software and decisions arise naturally. The process is never interrupted by peoples' explicit interaction with computers like plugging and unplugging cables for projecting information or even pressing buttons or adjusting sliders. Users can focus on their task and the system is there only to give them visual cues on what is going on - it never gets in their way.

With proxemics the relevance value is assigned automatically according to user distance to the shared screen. Without proxemics users must explicitly tell the system when this is supposed to happen. This is done by accessing a particular screen and adjusting a slider which controls the "relevance" value.

To handle the user identification we added a screen in the login phase of the system. In this screen there is a top-down view of the room, including a representation of the table, the screen, the sensor and the skeleton of each user as detected by the Kinect. The users are asked to tap the skeleton corresponding to them and the session can only begin once every user has done so.

3.6 Summary

In this section we first discussed the problems of CSCW that are good candidates for being solved through the use of proxemics. We then explained our scenario in more detail, a process which was already well defined before we began our work. To support this scenario we developed a base system which can optionally be equipped with a module for detecting proxemics in the room. Because of this architecture we were able to easily test several alternatives using the Wizard of Oz method and eventually implement the most promising ones.

These differences are evaluated in the following section, allowing us to understand whether proxemics introduces a benefit and if it targets the CSCW problems identified earlier.

4. EVALUATION

For validating our hypothesis - that proxemics introduces a benefit in collaborative face-to-face meetings - we compared the two developed systems. Because we evaluated collaborative software, we consulted the Methodology for Evaluation of Collaborative Systems (Damianos et al. [3]), which considers four levels of evaluation: requirement, capability, service and technology. Our goal is to conclude which system is better at each level and overall.

Considering this framework, the difference between both systems we are evaluating is at the technological level - we have two different implementations of the same relevance attribution service - one requiring the user to change the relevance manually and another which detects the relevance automatically. However, this does not mean that we can only evaluate at the technology level, because other levels might be affected by the way people use the system. The only exception is the service level, since the service is exactly the same.

The framework describes types of tasks that collaborative systems support, of which we select the ones which are the focus of our system: Planning, Brainstorming and Decision Making. Considering the objectives of the session, these types are the most adequate - users are given a goal and have to deliver a plan for reaching that goal; they have to brainstorm in terms of objectives, criteria, metrics and weights; and they must reach several decisions throughout the session, choosing between alternative objectives, criteria, metrics and weight assignments.

At the requirement level we are interested in understanding whether the different methods satisfy the requirements. This can be done by measuring task outcome, cost, user satisfaction and efficiency. Task outcome includes the number of generated artifacts, task completion and user ratings of product quality. Cost includes learning effort, number of turns, length of turns and total time. User satisfaction includes process satisfaction, outcome satisfaction, individual and group participation satisfaction. Efficiency includes the number of artifacts / time and user ratings about efficiency.

At the capability level we want to know which method allows a better communication between users. This is defined by the number of turns per participant, user ratings on quality of communication, getting floor control, getting the attention of other participants and ability to interrupt.

At the technology level we want to know which system performs better and how people react to it. This can be answered using standard user interface metrics - fundamental interactions, time and user satisfaction, which is also partly covered at the requirements and capability level.

With all the metrics selected, we moved on to define the experiment.

4.1 Experiment

The experiment involved eight groups of exactly three users - each one of the groups completed one session (either with or without proxemics). This number of users and groups was chosen in order to have the highest number of sessions possible - we needed at least three users to have a meaningful discussion and we wanted to have a reasonable number of sessions to compare.

We invited people working at YDreams to be our test subjects and were careful in making sure that the sessions were as equivalent as possible. Most of the users (58%) were interns at YDreams with ages ranging from 20 to 25. These interns all share a background in either software engineering (65%) or electrical engineering (35%). The remaining subjects were designers (17%), project managers (12.5%) and marketing specialists (12.5%). These professionals were generally more experienced, with ages ranging from 27 to 50. Out of all users, 87.5% were male and 12.5% were female.

To make sure that the experiments were equivalent, we selected two people from the intern group and one from the most experienced group for each session. This way it was not only guaranteed that the sessions were similar, but also that there would be a natural leader. We were also careful to group people together with other people they normally work with, increasing the chances of a dynamic and productive session.

All the test sessions took place in the same room, which was previously prepared for the effect, in the same way that it was done for the Wizard of Oz tests - a single table and a shared screen. We only changed one thing - in order to be sure that we didn't affect users' expectations, we removed the Kinect sensor from the room for non-proxemics sessions. If the sensor were there, it could be argued that users expected to use it somehow during the session and were disappointed not to. There were three devices available for the users - two tablets and one laptop. The only restriction we applied was that the mediator would always use the laptop.

Each session included: 10 minutes for a pre-session briefing - explaining the process and software using a slide presentation; 5 minutes for a pre-session questionnaire; 60 minutes to carry out the session; and 10 minutes for a post-session questionnaire.

The pre-session and post-session questionnaires were used for three purposes - measuring cost, communication and user satisfaction. We prepared a quick test covering topics on the process of the session and on the software usage and we gave users the same questionnaire before the session and after the session. This way we were able to measure which system is easier to learn. The post-session questionnaire also included questions regarding user satisfaction and communication.

The scenario that was given to the users was intentionally small in scope and also intentionally centered around a concept that every user (a YDreams' employee) could relate to. It was the same scenario as the one used in subsection 3.4.

A mediator was assigned in the beginning of the session by mutual agreement between the users. Because too many variables would change if the mediator changed (device in use, meeting dynamics, times, etc.) we removed the ability to change the mediator from the test applications. This way, there was one and only one mediator throughout each session.

The mission and vision were written before the sessions with a specialist and given to the users, since they are not supposed to be a part of the session, but previously established. The mission was "to create products that can be worn and that empower users with technology" and the vision was "to provide the most significant and aesthetically pleasant wearable technology products in the market".

During the session, we collected a series of values, in order to assess the previously discussed metrics. We instrumented the code and made it so that the server application would generate a file at the end of the session with the values we needed.

For measuring task outcome, we verified whether one of the methods produced more artifacts than the other (number of proposals, alternatives, objectives, criteria and metrics) and whether any method would influence the users individually to make more suggestions (number of proposals and number of alternatives per user).

In terms of cost comparison, we wanted to know if there were any differences in duration, not only the total duration of the session but also the duration of each of its segments. These values are also relevant for usability comparison.

The efficiency allows us to analyze the production of artifacts in regards to time. We measured the proposals, alternatives, objectives, criteria, metrics and weight proposals per minute. We also measured the number of weight proposals per minute in split-screen, which may indicate whether the use of split-screen influenced the number of weight proposals.

For comparing communication we want to know whether either method produced more turns and whether users participated more. This is measured by comparing data collected automatically (the distribution of user contributions over time) with the replies from the user satisfaction questionnaires, which have questions specifically targeted at this topic.

Finally, we compared the number of fundamental interactions (all clicks, taps and drag start, excluding text typing) in both methods. This metric is the most important in this work - this is the value where we expect to see a difference which is the starting point for the discussion of the remaining differences in all the other metrics.

Because we wanted to see if there were more differences in communication and time values we have also measured four more values through observation: the total time spent discussing alternatives in objectives, criteria and metrics (when two or more users engaged in discussion); the total time spent discussing weight assignment alternatives (when two or more users engaged in discussion); the total time when visualization is not adapted properly to alternatives discussion (if users are discussing alternatives, then the visualization should highlight those alternatives); and the total time when visualization is not adapted properly to weight discussion (when users are discussing weight distribution, the alternatives should be visible).

4.2 Results

We illustrate our final results with a series of figures that compare the average results for the proxemics and non-proxemics sessions. In those figures, we also include two more values: the minimum and the maximum. We also present our final results with a confidence value calculated using the t-test.

At the requirements level, the differences in completion and generated artifacts are inconclusive. However, there is a great difference in the perception of the results by the users (view Figure 6). The average result satisfaction in the proxemics solution is 3.75 out 4, while it is only 3.17 out of 4 for the non-proxemics solution (confidence: 98%). As for learning, there were no significant differences in the

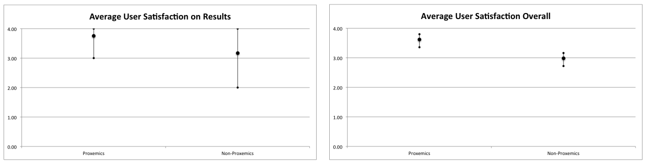

Figure 6: Comparison of user satisfaction with the results of the session and overall.

Figure 7: Comparison of self-assessment on process and software understanding.

results apart from the self-assessment. Users in proxemics sessions were more confident in their understanding of the process - 3.92 versus 3.33 (confidence: 99%) - and of the software - 3.83 versus 3.33 (confidence: 98%). These results are shown in Figure 7. Finally, efficiency was also similar in most situations: objectives/minute, criteria/minute and metrics/minute. However, there were differences in the number of users who showed their weight alternatives in the shared screen per minute. In the proxemics sessions an average of 0.0004 users showed their weights, while only 0.0001 users did the same in the non-proxemics sessions. This difference was calculated with a confidence of 97%.

For the capability level, we again point to the differences in the number of users who showed their weight assignment in the shared screen. By itself, this means only that there was more turn taking in this stage of the session. However, this fact acquires new relevance when analyzed together with the user satisfaction towards turn taking. Considering only the opinion of users about how they were able to express themselves and how others perceived them, we see that users in proxemics sessions were satisfied on average in a scale of 3.83 out of 4, while users in non-proxemics sessions were only satisfied in a scale of 3.28 out of 4 with a confidence of 99%.

At the technology level, there is a statistical difference in the number of fundamental interactions. The average number of fundamental interactions p/minute was 6.77 for the proxemics solution and 7.37 for the non-proxemics solution with a confidence of 98% (view Figure 8). The average satisfaction (Figure 6) for a proxemics session is 3.62 out of 4, while for a non-proxemics session it is only 2.98 out of 4 (confidence: 99%). The differences in time are also mostly negligible, except that users in proxemic sessions spend an

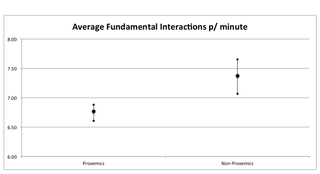

Figure 8: Comparison of the number of fundamental interactions.

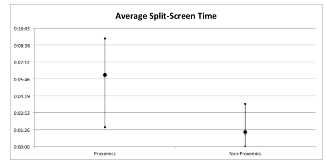

Figure 9: Comparison of the time spent in split screen.

average of 4:52 minutes more in split screen than the users in non-proxemics sessions with a confidence of 95% (view Figure 9). Additionally, the amount of time when the visualization is not correctly adapted to the discussion of weight alternatives is much higher in the non-proxemics sessions - 91% of the time versus 16% of the time in the proxemics sessions, with a confidence of 99%.

4.3 Discussion

Looking at the results we draw three important conclusions: 1. The proxemics solution yielded better user satisfaction at all levels; 2. Even though there weren't any significant differences in learning, users of the proxemics solution felt like they learned more; and 3. Users of the proxemics solution exercised more of the system, namely the visual adaptation of the shared screen to what is being discussed.

One of the reasons why users were more satisfied with the proxemics solution is the fact that it requires less fundamental interactions. This is a simple heuristic - if users can complete the same task with less work, they will be more satisfied with the interface. What's special about this case and about proxemics in general is that users can actually do more with less work. By correctly identifying when users would want the system to react, we have been able to produce a solution that reflects the users' needs without a conscious input and we have proved that, if applied correctly, this technique improves the users' perception of the system.

It might seem contradictory that in spite of having less fundamental interactions in the proxemics sessions, we could not find any significant difference in the duration of the sessions. Because we've also seen an improvement in satisfaction, we believe that the reason for finding no difference in duration is simply because the dominant component of the session is discussion. This is supported by the low number of interactions in the sessions - between 6 and 7 interactions per minute.

The improvement in satisfaction is also what explains why proxemics users felt like they learned more, even though they didn't. Users had a more positive attitude at the end of the session and felt better with themselves and with the results.

Knowing that users of the proxemics system took advantage of the adaptive visualization features makes us realize that this is something that users wanted, but that was not supported correctly in a non-proxemics solution. In other words, this means that we have met a users' need through proxemics, that could not be achieved without proxemics.

5. CONCLUSIONS

We hypothesized that CSCW applications could benefit from the use of proxemics as it allows the reduction of interaction protocol and a bigger focus on the task. We targeted two problems - breakdown of intuitive decision making and awareness in regards to what is being done. To address these issues we developed a system for a particular scenario that takes advantage of the moments when people intend to present their work by automatically adapting the visualization to what is in discussion. For understanding the impact of this feature, we compared two versions of our system with the same functionalities but with a fundamental difference - one took advantage of proxemics and the other didn't. The results show that proxemics effectively reduces the interaction protocol and as a results users are more satisfied with the system and their work.

The major accomplishment of this thesis is understanding exactly where proxemics is responsible for an improvement and to conclude that it addresses the problems we identified earlier. We observed that with proxemics there are less fundamental interactions, which shows that there is a reduction of protocol, improving user satisfaction. We proved that users are receptive to proxemics even in a serious environment. We didn't compare only how proxemics affected single interactions, but most importantly how it affected the whole system and how users perceived it. As for individual interactions, this thesis also presents experiments on several methods for using proxemics data in mediator definition and information visualization.

We evaluated a single session but conjecture that proxemics would also have an impact in the long term: because users feel more comfortable with the system and are more confident about their work, the adoption rates would be higher in theory. In the future, it could be interesting to assess if this is in fact true by having different companies use the software for more than one session and reporting on user satisfaction and adoption over time.

With better technology we could also think about improving the system itself. With more accurate hardware and algorithms in place we could consider taking further advantage of relationships between people and machines. Reliability could also be improved by addressing problems with occlusions. This could be done by taking advantage of more advanced hardware or by using multiple sensors.

Using proxemics in interaction still has a long way to go - until now it has mostly been used in interaction experiments, but we have shown the benefits of using it in a real world CSCW scenario. We were able to take advantage of a small part of all the spatial information communicated during a meeting and it already improved the users' opinion dramatically. With future technologies and experiments we can only expect to see much more improvements with the goal of achieving a level of interaction with machines closer to how we interact with each other.

References

[1] Helena Alegre and Sérgio T. Coelho. Infrastructure asset management of urban water systems. *Water Supply System Analysis - Selected Topics*, 2012.

[2] Marco Cristani, Giulia Paggetti, Alessandro Vinciarelli, Loris Bazzani, Gloria Menegaz, and Vittorio Murino. Towards computational proxemics: Inferring social relations from interpersonal distances. In *Proceedings of IEEE International Conference on Social Computing*, pages 290–297, 2011.

[3] Laurie Damianos, Lynette Hirschman, Robyn Kozierok, Jeffrey Kurtz, Andrew Greenberg, Kimberley Walls, Sharon Laskowski, and Jean Scholtz. Evaluation for collaborative systems. *ACM Comput. Surv.*, 31(2es), June 1999. ISSN 0360-0300.

[4] Paul Dourish and Victoria Bellotti. Awareness and coordination in shared workspaces. In *Proceedings of the 1992 ACM Conference on Computer-supported Cooperative Work*, CSCW '92, pages 107–114, New York, NY, USA, 1992. ACM. ISBN 0-89791-542-9.

[5] Saul Greenberg, Nicolai Marquardt, Till Ballendat, Rob Diaz-Marino, and Miaosen Wang. Proxemic interactions: the new ubicomp? *interactions*, 18(1):42–50, January 2011. ISSN 1072-5520.

[6] Jonathan Grudin. Why cscw applications fail: Problems in the design and evaluationof organizational interfaces. In *Proceedings of the 1988 ACM Conference on Computer-supported Cooperative Work*, CSCW '88, pages 85–93, New York, NY, USA, 1988. ACM. ISBN 0-89791-282-9.

[7] Edward Hall. *The Hidden Dimension*. Anchor, 1966. ISBN 0385084765.

[8] Robert Johansen. *GroupWare: Computer Support for Business Teams*. The Free Press, New York, NY, USA, 1988. ISBN 0029164915.

[9] Wendy Ju, Brian Lee, and Scott Klemmer. Range: exploring proxemics in collaborative whiteboard interaction. In Mary Beth Rosson and David J. Gilmore, editors, *CHI Extended Abstracts*, pages 2483–2488. ACM, 2007. ISBN 978-1-59593-642-4.

[10] John Kelley. An iterative design methodology for user-friendly natural language office information applications. *ACM Trans. Inf. Syst.*, 2(1):26–41, January 1984. ISSN 1046-8188.

[11] Adam Kendon. *Conducting Interaction: Patterns of Behavior in Focused Encounters*. Studies in interactional sociolinguistics. Cambridge University Press, 1990. ISBN 9780521380676.

[12] Jun Rekimoto. A multiple device approach for supporting whiteboard-based interactions. In *CHI '98: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 344–351, New York, NY, USA, 1998. ACM Press/Addison-Wesley Publishing Co. ISBN 0-201-30987-4.

[13] Robert Sommer. Personal Space. The Behavioral Basis of Design. *eric.ed.gov*, 1969.