“Who’s in Charge Here?” Communicating Across Unequal Computer Platforms

MARIA VELEZ
Rutgers University
MARILYN MANTEI TREMAINE
New Jersey Institute of Technology and Rutgers University
and
ALEKSANDRA SARCEVIC, BOGDAN DOROHONCEANU,
ALLAN KREBS, and IVAN MARSIC
Rutgers University

People use personal data assistants in the field to collect data and to communicate with others both in the field and office. The individual in the office invariably has a laptop or a high-end personal workstation and thus, significantly more computing power, more screen real estate, and higher volume input devices, such as a mouse and keyboard. These differences give the high-end user the ability to represent and manipulate collaborative tasks more effectively. It is therefore useful to know what impact these differences have on work performance and work communications. Four different platform combinations involving a PC and a PDA were used to examine the effect of communicating via heterogeneous computer platforms. The PC platform used a mouse, a keyboard, and a 3-dimensional screen display. The PDA platform used a stylus, soft buttons, and a 2-dimensional screen display. A variation of the Tetris wall-building game called Slow Tetris was used as the subjects’ collaborative task. A second factor in the experiment was role asymmetry. One subject was arbitrarily put in charge of the task solution in all of the combinations. An analysis of the solution times found that subjects with mixed platforms worked slower than their homogeneous counterparts, that is, a person in charge with a PC worked faster if his partner had a PC. An in-depth analysis of the communication patterns found significant differences in the exchanges between heterogeneous and homogenous combinations. The PC-to-PDA combination (with the person on the PC in charge of the solution) took significantly more time than the PC-to-PC combination. This extra time appears to come from the disadvantage of having a partner on the PDA who is unable to help in solving the problems. The PDA-to-PC combination took approximately the same amount of time

---

The research is supported by NSF Contract No. ANI-01-23910, New Jersey Commission on Science and Technology, and by the Rutgers Center for Advanced Information Processing (CAIP) and its corporate affiliates.

Authors’ addresses: M. Velez, A. Sarcevic, B. Dorohonceanu, A. Krebs, and I. Marsic, CAIP Center, Rutgers University, Piscataway, NJ 08854-8088 USA; email: {mariacv,aleksarc,dbogdan, krebs,marsic}@caip.rutgers.edu; URL: http://www.caip.rutgers.edu/disciple; M. M. Tremaine, Information Systems Department, New Jersey Institute of Technology, Newark, NJ 07102 USA; email: tremaine@njit.edu.

ACM Transactions on Computer-Human Interaction, Vol. 11, No. 4, December 2004, Pages 407–444.

as the PDA-to-PDA combination despite having one team member with a better representation. This member was, unfortunately, not in charge of the solution. The PDA-to-PC heterogeneous combination exhibited more direction giving, less one-sided collaboration, and more takeover attempts than any of the other combinations. Overall, roles were maintained in the partnerships except for the person with the PDA directing the person with the PC.

Categories and Subject Descriptors: C.2.4 [**Computer-Communication Networks**]: Distributed Systems—*Distributed applications*; D.2.2 [**Software Engineering**]: Tools and Techniques—*User interfaces*; H.1.2 [**Models and Principles**]: User/Machine Systems—*Human factors*; H.5.3 [**Information Interfaces and Presentation**]: Group and Organization Interfaces—*Collaborative computing*; *computer-supported cooperative work*; *evaluation/methodology*; *synchronous interaction*

General Terms: Design, Experimentation, Human Factors

Additional Key Words and Phrases: Collaboration differences, heterogeneous computing, media effects, mobile computing

1. INTRODUCTION

This article investigates the performance and communication patterns that take place when collaborators use unequal computer platforms for conducting their collaboration. We are envisioning a scenario where a person in the field has some form of a Personal Data Assistant (PDA) that contains a shared environment with another individual. This second individual in the dyadic communication is either located in the field and is equipped with the same form of display platform, a PDA, or in the office with a high-end workstation (we are calling this a PC). If the two individuals are communicating on the same type of platform, we say that the platforms are *homogenous*. If they are using different platforms, we say that the platforms are *heterogeneous*.

With the spread of wireless communication and the desire to travel light, a PC to PDA collaboration is a likely scenario for future work practices. We can envision people in the office sending reduced versions of spreadsheets or drawings to workers who are on site and working on them collaboratively. Although computer power is continuing to increase dramatically, there is likely to always be a significant difference between the PDA and the PC because of portability constraints placed on the PDA. This raises both technical and human communication issues given the limitations in computing power, bandwidth, display size, and input capabilities of the PDAs. The problem of how to build groupware that can accommodate such platform disparities while effectively supporting collaboration is not well understood. In this article, we examine one small part of the differences that are likely to exist between a PDA and a PC, differences in problem representations brought about by display and computing power differences, and differences in work roles.

Display disparities can take different forms. They can result from the physical characteristics of display devices, such as size, color resolution, aspect ratio, and spatial distribution. They can also result from the way information is visualized, such as: dimensionality of visualization (2D vs. 3D), degree of compression, abstraction or summarization, or level-of-detail differences. Most display differences are a combination of the above factors.

Work roles can also be very different. They can be peer-to-peer where two people are of equal status, mentor-to-disciple, boss-to-employee, expert-to-novice, and so forth. Each of these roles embodies different protocol requirements for communication exchange as the collaborators follow standard cultural practices. For example, Wynn [1979] found that an employee of lesser status in an office who needed to request a favor of an individual of higher status, repeated the request three times when following normal office protocol.

In this work, we focus on a combination of two types of display disparities—display size and dimensionality of visualization (2D vs. 3D). Since these differences are inherent in the two platforms selected, we also have two variations in user input: the traditional mouse of the PC interface and the stylus of the PDA interface. Possible application scenarios where 3D-support is likely to occur include training, equipment maintenance, and medical emergency support. We have chosen to compare the 2D- to 3D-visualizations because the number of pixels and screen size of a PDA prevent it from adequately rendering a useful 3D-display for many tasks. Nevertheless, this difference between the two visualizations is likely to have a profound effect on problem solving. In as simple a task as finding and selecting an icon on a screen display, Ark et al. [1998] found that users with a 3D-representation of the icons performed faster than an equivalent task with a 2D-representation. This performance advantage was maintained over two days of trials. Because input device differences are also likely to have an impact, we have designed both mouse and stylus actions to be point-and-click menu selections. In an earlier study [Marsic et al. 2002], we found significant performance differences that arose because of the differences in input devices. Although individuals in the 3D-environment had a better view of the task, they were hampered in moving about the 3D-environment because, even after training, they had difficulty navigating with the 3D-mouse.

1.1 Projected Effects of Platform Differences

If collaboration is to occur between two individuals, they must have a *shared understanding* of the task and the details of the task. If they do not have this shared understanding, they will have problems communicating and will fix these problems with additional communication. They are said to be establishing a common ground or doing *common grounding* with this conversational repair [Clark and Brennan 1991]. If two collaborators are working on disparate computer platforms, we can expect misunderstandings between them in terms of how they view the collaborative task and their partner's capabilities in interacting with the computer. Thus, we should observe common grounding conversations at the beginning of these collaborations.

If display and input differences are large, establishing a common ground may be difficult and users may perform the collaborative task poorly. In addition, there may be frustration or impatience resulting from the collaboration disparities. For example, if one collaborator is in charge of the task communication but has a low-end collaboration device, then that person may become frustrated because his or her PC collaborator is changing items in the shared workspace and suggesting solutions before the PDA collaborator has completely absorbed the problem.

Alternatively, the person with the PC may be adding their expertise to the field situation, but requesting updates at a pace that is too rapid for the fieldworker to accommodate. It may be that the representation on the PDA has been so reduced in size or detail that the PDA owner has trouble visualizing and comprehending the suggestions of the PC collaborator. Or, the person with the PDA may completely rely on the person seated at the PC to present a solution to the problem without providing any collaborative input. If these scenarios occur, we expect to see some transfer of authority to the person with the more powerful platform. We should be able to see this transfer in the forms of conversations that take place and the levels of politeness that are exhibited in the conversations [Brown and Levinson 1979]. A study involving communication between a flight officer and a helicopter pilot found a seamless transfer of authority that moved back and forth between the two communicants as each person's role became salient to the task [Linde 1988]. Where one computer platform gives a more powerful visualization of the task, it is likely that this visualization will always give the person behind the PC the most salient task role and thus, the continued authority in the collaboration. Linde argues that the seamless exchange of authority in collaborative tasks is a commonplace part of the collaboration. Therefore, if we limit this role, we may also impact the quality of the collaboration.

Dryer et al. [1999] show that human perception of other humans is affected by the computing device they are using, with less common devices causing negative attributions to be applied to the device's user. Their studies involved the perception of one member of a collaborating pair who used a supportive computing device. At the end of their article, they provide a checklist of ways in which a computer device can have social computing embedded in its design. One of the items on the checklist is "power," which they describe as "the extent to which a device puts one person more 'in charge' than another person and the extent to which the device communicates a difference in status." In the heterogeneous combination, the difference in capabilities and representations of the PC and PDA is likely to confer power on the PC user. If so, we should be able to observe some of this status conferral in the collaborative communication.

In contrast, collaborators with homogenous platforms should not have the same communication difficulties and are more likely to work in a collaborative fashion with both members of the communicating pair discussing the problem being presented. It is possible that collaborators with similar low-end devices will be unhappy with their collaboration because of the device's inability to support richer collaboration, but it is also likely that they will become more collaborative joining together on a difficult task.

Our research compares collaborations across disparate and similar platforms using the same task, while assigning one person to lead the collaboration. We look at performance differences in the collaborations and at the conversational exchanges that transpire during the collaboration in order to get a better understanding of how to best support collaborations over disparate platforms.

The article is structured as follows. We first present a summary of other studies that have looked at collaboration between users with different views of the work task. We then describe the task that our subjects were assigned to complete. Following this, we explain the study that we conducted. We then discuss our results and conclusions based on our analyses of the performance times and communications that were collected in the collaboration. Finally, we present the future work this study has suggested and the collaboration tools that might support more equal collaboration across heterogeneous platforms.

2. BACKGROUND AND RELATED WORK

A precursor to the evolution of heterogeneous groupware can be traced in the evolution of the so-called WYSIWIS (What You See Is What I See) technique. Since the beginning of groupware-software development, WYSIWIS has been viewed as an essential capability in groupware systems. However, although the WYSIWIS idealization recognizes that efficient reference to common objects depends on a common view of the work at hand, strict WYSIWIS was found to be too limiting, and relaxed versions were proposed to accommodate personalized screen layouts [Stefik et al. 1987]. In subsequent work [Tatar et al. 1991], problems were reported with non-WYSIWIS systems because manipulation and editing processes were private and only results were shared. This discontinuity of the interaction created ephemeral environment differences that affected collaboration.

Non-WYSIWIS is quite common in collaborative virtual environments (CVEs) [Hindmarsh et al. 1998; Steed et al. 1999] where collaborators navigate independently to accomplish their own goals. Hindmarsh et al. [1998] suggest that users have difficulties in establishing mutual orientations in CVEs, but that having some common frame of reference, for example a 2D-map added to the CVE, might alleviate the difficulties.

Research has also shown that support of mobile workers can be an effective use of computer-based collaboration. Kraut et al. [1996], for example, show that fieldworkers make quicker and more accurate repairs when a remote expert is providing assistance. The researchers also found that the communication media did not affect the quality of the task performance. When video was not used, the collaborators substituted voice descriptions of the task to generate a common ground between them.

A study by Billinghurst et al. [1999] found that asymmetries in collaborative interfaces impair collaboration, but that the impact is decreased if the asymmetries matched the role people played in the collaboration. In that study, the differences between the collaborators were not in the capabilities of the platform but in the states that each of the collaborators personally created for themselves. Our study focuses on collaborations where one of the collaborators has distinctly lower capability than the other. We assign roles of head problem-solver (communicator) and assistant (doer) to both the lower capability and higher capability platforms. Thus, we should expect that the asymmetry of platforms will impair collaboration most when roles do not match the asymmetry.

It is not known how disparities in computing platforms affect collaboration. Research in single-user interfaces has shown that display size does have an effect on performance in that the smaller screen size generally impedes task performance [Jones et al. 1999; Kamba et al. 1996]. In our earlier study

[Marsic et al. 2002], an office employee had to instruct a moving company employee remotely where to place furniture in a new office. The participants (simulated by students) were collaborating using desktop PC and PDA platforms with 3D- and 2D-virtual representations of the office, respectively. The collaborative teams were assigned to each of four conditions: one 2D-to-2D-communication, two 2D-with-3D-communications, and one 3D-to-3D-communication. We found that the performance times were significantly different depending upon the direction of the communication and the homogeneity of the platforms of each of the collaborators. Both the task time and the number of conversational misunderstandings between collaborators were significantly larger in the 2D-to-3D-collaboration than in any other setup, because of the viewing limitations of the 2D-environment coupled with the input limitations of the 3D-environment. Users found it difficult to orient in the 3D-space, so that moving around the environment and placing the furniture with a 3D-pointing device (the Magellan Space Mouse [Logitech 2002]) turned out to be a difficult task. The difficulty arose in visualizing the 3D-space and in manipulating the pointing device. In contrast, the collaborator in the 2D-environment had a limited view of the furniture placement task. This caused difficulty in translating the positioning instructions into 3D placement activities. Task performance was best in the 3D-to-2D-collaboration and the number of conversational requests for clarification in the movement instructions was significantly lower than in the other collaboration configurations. This combination of platforms succeeded because of the viewing advantages of the task manager and the input advantages of the furniture placer. The results bear some similarity to the findings by Steed et al. [1999], where leaders more readily emerged in a group task from the more immersive VR environment.

3. THE UNEQUAL PLATFORM APPLICATION

To generate our collaboration task, we used DISCIPLE, a software framework for developing heterogeneous collaborative applications [Marsic 2001]. The applications use different data representations in order to best match the local computing platform's capabilities. The collaborative events exchanged between the applications are transformed to a representation domain to preserve semantics. Using the toolkit, we developed two applications: a 2D-graphics editor (Pocketscape) and a 3D-virtual world (cWorld). We then used each of these to develop the task that our subjects used in their collaboration. The task we developed was a 3-dimensional problem-solving game that both subjects could view, one in 3D and one in 2D. This game is described below with Figures 1 and 2 illustrating the PC and the PDA views of the game, respectively.

3.1 The Slow Tetris Game

To examine the performance differences across platforms, we needed to develop a task that had the characteristics of a real-world task, but also allowed us to run our studies on the typical undergraduate subject pool available at a university. Two possible real-world scenarios match the task we chose. One is a 3-dimensional repair task where an expert in the home office aids a repairman

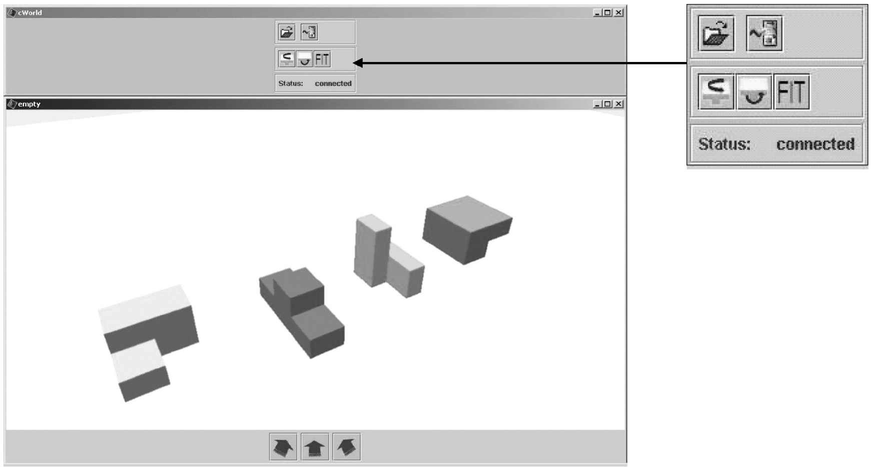
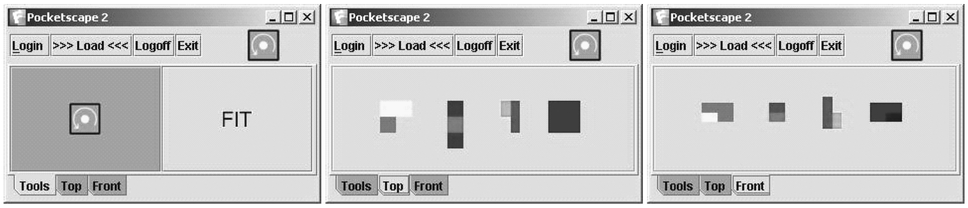


Fig. 1. A configuration of 3D-blocks built using *cWorld*. The left view of the configuration is shown. A front view and right view can be obtained by the user "clicking" on the arrows at the bottom of the screen. A detailed view of the toolbar is shown in the upper right hand corner.



Toolbar Top View Front View

Fig. 2. The toolbar, top and front views of the block parts as seen using *Pocketscape*. The block arrangements correspond to the configuration shown in Figure 1. The darker sections show depth in the 2D-views. Only one of the views is visible at a time to the PDA user. Users switch between the views by touching the PDA stylus to the tabs labeled "Tools", "Top", and "Front" at the bottom of the screen.

in the field, or where a repairman in the field describes a problem or a repair situation to a person in the home office. The second scenario is planning for the deployment of rescue efforts or military ground forces with a remote command post overseeing the effort, and personnel in the field making individual recommendations. The command post has a 3D-map of the terrain but not a view of local conditions, which are supplied by the individuals with PDAs. We set up the task so that one person would be giving instructions on the task while the other person performed the suggested manipulations. In this way, we artificially assigned status or control to one individual in the paired collaboration.

Since we were not likely to find equipment repair or hurricane rescue equipment deployment skills in our university subject population, we developed a block manipulation game that we call Slow Tetris. (We picked this name because the game is somewhat similar to a popular 2-dimensional game called Tetris$^{\mathrm{TM}}$.) The object of the task is to build a wall from a series of building blocks that are supplied to the collaborators. Figures 1 and 2 illustrate an example

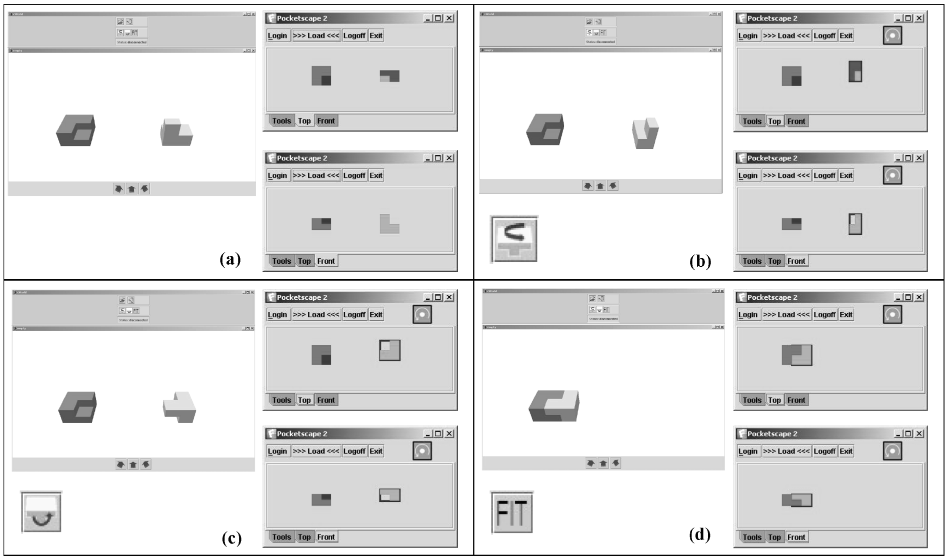


Fig. 3. A step-by-step solution of a two-piece task in the 3D and the 2D-environment. (a) Initial block positions. (b) Position of the second block (from left to right) after one Top Rotation (yaw). (c) Position of the second block after one Front Rotation (roll). (d) Completed wall after fitting the second block. Screen displays on the left are 3D and screen displays on the right are the top and front views of the blocks in 2D. The icon at the lower left of the 3D-screen is an enlargement of the tool for the rotation or fit moves.

set of block shapes that can be built into a wall in their 3D- and 2D-displays, respectively. The first block is fixed and the subsequent blocks are in the order they are to be placed in the wall. The subsequent blocks have to be rotated into the correct orientation so that they fit into the wall being built. The aim of the task is to build a wall in the minimum possible time.

Figure 3 illustrates the steps needed to create a wall with a two-piece task. The steps are shown in both the 3D- and 2D-environments. Two kinds of rotations are possible, (*i*) top- and (*ii*) front-rotation. The top-rotation rotates a block horizontally around the y-axis (yaw), as shown in Figure 3(b). The front-rotation rotates a block vertically around the z-axis (roll), as shown in Figure 3(c). All rotations are counter-clockwise. Figures 1 and 2 show the screen layouts that our subjects saw when they performed the task. The tool buttons for rotating and fitting the blocks in the 3D-task are shown on the top of the screen. We have enlarged the tool icons from the 3D-view and placed them in the lower left corner of Figures 3(b), 3(c), and 3(d). Rotation in the 3D-environment is performed by clicking on the front- or top-rotation tool and then clicking on the block. A tool is active until another tool is selected. Therefore, if a user wishes to rotate a block three times, he simply needs to click on the rotation tool and then click the block three times. Fitting is done by selecting the fit-tool. A small beep is sounded if the fit is not successful, and the block pieces remain unfitted.

Rotation in the 2D-environment is the same, except the user needs to display the view in which the rotation is to be carried out. So, to perform a top-rotation (yaw), the user must have the top view of the block displayed before touching

Table I. Role and platform assignment for the comparison of heterogeneous to homogenous platform collaborations. Communicators always maintain the same platform and always have the role of communicator. The same is true for doers.

Collaboration Type	Role Assignment
PC→PC	Tom→David
PC→PDA	Tom→Harry
PDA→PDA	John→Harry
PDA→PC	John→David

the stylus to the block to be rotated. To perform a front-rotation (roll), the user must have the front view of the block displayed. Once a block is perceived to be oriented correctly so that it will fit into the left wall piece, the user can select the fit-tool and touch the block that is to be fitted into the wall. To do this, they need to go to the Toolbar view. The piece will fit if it is appropriately oriented.

The 2D-representation is particularly unsuited to the puzzle-solving task of the Slow Tetris game. Therefore, the very difficulty of the 2D-representation exaggerated the differences between the PC and PDA platforms, and it can be argued that any findings from the study are not universally applicable because they do not represent the natural collaborative environment. In truth, heterogeneous collaboration environments are not widespread at the moment, so it is hard to say what an ecologically valid environment is. Our studies were done in quiet offices without the expected interruptions of an external mobile setting that could disadvantage the PDA user even more. Also, note that software utilities that adapt tasks to multiple platforms do not examine whether the representation of a particular problem is easy for the user, so that it is likely that collaborating users will be faced with future difficult representations.

3.2 Experimental Setup

To determine if collaboration is affected by inherent differences generated in displays on the PDA and PC, we generated an experiment setup that would compare teams of two collaborators communicating either with the same type of platform or with different platforms. Since collaborations are likely to be between individuals of different status, we also assigned one of the collaborators to direct the problem solving. In two cases, we assigned the person in charge of problem solving (we called this person the *communicator*) to a PC environment. In two other cases, we assigned the person in charge to a PDA environment. Thus, we had two individuals who were directing the solution of the Slow Tetris problem, either from a PDA or PC platform. They were collaborating with a person who was either on a PDA or a PC platform. (We called this second person receiving directions, the *doer*.) This arrangement gave us four different combinations of PDAs and PCs, two of them being homogenous and two being heterogeneous. We show these arrangements in Table I. The arrow indicates the direction of the communication and thus, who is in charge. We have created a virtual group of subjects named Tom, David, Harry, and John to illustrate our subject assignment for the four communication arrangements.

3.3 Expectations

In terms of interacting with the PC and PDA environments, the steps required to rotate a block in either the x-y plane or the y-z plane are identical. On the PC, the user selects the rotation tool desired and then applies the tool to the block being rotated. On the PDA, the user also selects the rotation tool and then applies the tool to the block being rotated. In the process of selecting the desired rotation, the PDA user switches views from top-to-front, but the PDA user can also switch views simply to better understand the problem. The PC user also has three possible views to select from: left, central, and right side. If we perform a keystroke level model [Card et al. 1980] analysis of both environments for just the user rotations, we have the following representations for expected performance times for the example shown in Figure 3:

$$\begin{aligned} Time_{\mathrm{3D}} &= T_m + T_{point(top)} + T_k + T_m + T_{point(block)} + T_k + T_m + T_{point(front)} + T_k \\ &\quad + T_m + T_{point(block)} + T_k + T_m + T_{point(fit)} + T_k + T_m + T_{point(block)} + T_k \end{aligned}$$

where T_m is the mental time to bring up that portion of the unit task, $T_{point(top)}$ is the time to point to the top rotation button, and T_k is the time to click the mouse button. If we assume that the times for each of the pointing tasks are approximately the same because the block parts are approximately equidistant from the tool buttons, we can reduce our model to:

$$Time_{\mathrm{3D}} = 6T_m + 6T_k + 6T_{point(\mathrm{3D})}.$$

If we generate a keystroke level model of the time for the 2D-representation, we obtain a similar model:

$$\begin{aligned} Time_{\mathrm{2D}} &= T_m + T_{point(top)} + T_m + T_{point(block)} + T_m + T_{point(front)} + T_m + T_{point(block)} \\ &\quad + T_m + T_{point(toolbar)} + T_m + T_{point(fit)} + T_m + T_{point(top)} + T_m + T_{point(block)}. \end{aligned}$$

All of the tool sizes in the 2D-representation are the same, so we expected to have similar times for selecting a tool since they are nearly equidistant from the block. The size of the fit-tool and the block are also equal. This gives us the following simpler model for the 2D-representation:

$$Time_{\mathrm{2D}} = 8T_m + 4T_{point(tool)} + 4T_{point(block)}.$$

There are no mouse clicks in the 2D-representation. The user only has to touch the block. In the 3D-representation, the target size is larger but the distance to travel to the block is longer. A calculation of pointing times using Fitts Law for both tasks gives a 2 second advantage to the 3D-representation. ($Time_{\mathrm{3D}} = 14.7$ sec vs. $Time_{\mathrm{2D}} = 16.4$ sec). We used the Fitts Law models provided by MacKenzie [1992] and MacKenzie and Soukoreff [2002] for these calculations. We also used a value of 1350 msec for the mental preparation time and 200 msec for the mouse button clicks as per the Card et al. [1980] measurements. Thus, in our design of the user manipulations, we have made the tasks approximately equal. However, we expected that most of our team's time would be spent on problem solving and communicating between the two environments so that even small differences would have little impact.

The task requires our users to perform *spatial reasoning*. Users must mentally develop a sequence of rotation steps that will correctly place the next block in the wall. This is easier to do in the 3D-environment because the blocks to be moved are already represented as 3D-blocks. In the 2D-environment, the user needs to mentally visualize the blocks in their 3D-form in addition to visualizing their rotation. This is a much more difficult cognitive task. We therefore expected that users of the homogenous combination of PC→PC collaboration would have the easiest task and the least problems with communication.

In contrast, we expected that the PDA→PDA collaboration team would have the most difficulty in completing their task because they would have no additional visual support on either platform to help with the solution. However, the team would not experience large amounts of communication problems because they would be working with similar platforms. On the other hand, the PC→PDA and PDA→PC communications would suffer from differences in understanding among the team members, but we expected that the PC→PDA collaboration would be effective and efficient because direction would be coming from the person with the most knowledge about the task. The PDA→PC collaboration is expected to be slow and cumbersome and to have the most communication problems. The person with the least informative display is guiding the block placement, possibly to the chagrin of the PC-platform person who can envision more optimal rotations. We expected this communication to be problematic but the task times to be better than the PDA→PDA communication because the combination still has the advantage of the PC representation.

More succinctly, we stated our expectations as follows:

$$\text{PC}\rightarrow\text{PC} \gg \text{PC}\rightarrow\text{PDA} > \text{PDA}\rightarrow\text{PC} > \text{PDA}\rightarrow\text{PDA}$$

in terms of *performance time*, where "$>$" means faster (shorter completion times) and "$\gg$" means much faster.

$$\text{PC}\rightarrow\text{PC} \neq \text{PDA}\rightarrow\text{PDA} \neq \text{PC}\rightarrow\text{PDA} \neq \text{PDA}\rightarrow\text{PC}$$

in terms of *communication patterns* where "$\neq$" means that the types and number of communication exchanges are significantly different. This portion of the experiment was exploratory in the sense that we did not know what types of communication exchanges would occur until we ran the experiment and coded what the communication exchanges were. Therefore, we wrote the following differences we expected to observe, but not the directions of the differences:

- —We expected problem solving to be unequally shared among the partners for different platform combinations.
- —We expected more common grounding conversations for the heterogeneous platform combinations.
- —We expected authority conflicts to be different for the different platform combinations.
- —We expected the amount of feedback team members give to their partner's move suggestions to be different across different platforms.

4. METHOD

4.1 Study Overview

A team of two was given the task of solving a set of Slow Tetris problems. Teams were randomly assigned to combinations of PDA and PC computers for their collaboration. The direction of the team communication is also assigned so that one collaborator has the role of communicating the directions (*the communicator*) to the other team member who performs the action (*the doer*).

We assigned these roles to convey status to one of the members of the pair, the communicator, who directs the activities of the other person. We also assigned these roles because pilot studies indicated that our team members were more likely to collaborate than work independently with these experiment restrictions. Although we assigned communicator and doer roles to our teams, in all cases, we told each member of the pair that they could and should contribute verbally to the solution. The assignment of roles to the task also allowed us to examine both directions of communication in the heterogeneous combination, that is, PC→PDA and PDA→PC. Teams used either homogeneous platforms (PC→PC or PDA→PDA), or heterogeneous platforms (PC→PDA and PDA→PC). The arrow (→) indicates the direction of the communication, that is, from the communicator (who is directing the block placement) to the doer (who is performing the block placement). None of the subjects in each of the two-person teams knew their partner before the study. This avoided the possible effect of preexisting communication patterns and status relationships.

4.2 Subjects

Forty-two graduate and undergraduate students were solicited for this study through advertisements posted in student centers around the university. Subjects were required to be right-handed males who enjoyed playing computer games. All subjects were paid at least $8.00 for one hour or less of participation. Subjects who took longer than an hour were paid $2.00 for each additional 15 minutes that they participated in the study. No subject required more than two hours to complete the experiment. In order to avoid subjects deliberately extending the experiment, the advertisement and the experiment consent form did not indicate that payments would be issued for the time that extended beyond the hour advertised.

After filling in experiment consent forms, subjects were trained on the Slow Tetris computer game, first on the PC and then on the PDA. Each subject completed 13 wall-building tasks on each of the computer platforms. The tasks were ordered so that simple 2-block tasks began the training, and later 3-block tasks used configurations learned in the earlier tasks. The time to move each block in the wall-building tasks was recorded. Six of the subjects did not complete the PDA version of the Slow Tetris tasks because it was too difficult for them. They were not used in the subsequent study. We did not use the data from four additional subjects because one of the subjects had problems formulating communications to his partners. This subject gave directions to both of his collaborators, asking them to perform a top rotation when he meant a front

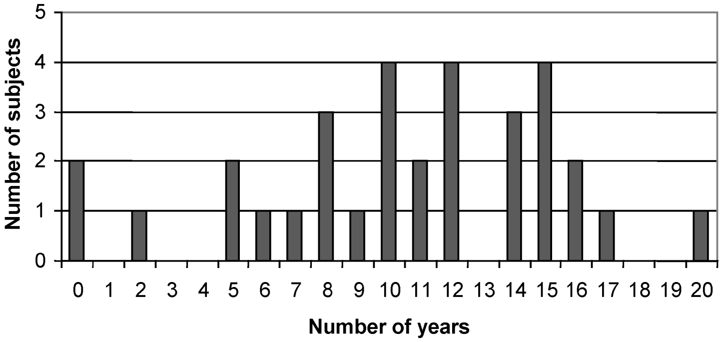


Fig. 4. Distribution of subjects' previous experience with video games.

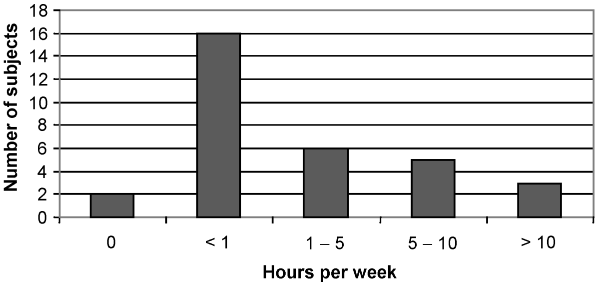


Fig. 5. Distribution of number of hours of video games played per week by our subject population.

rotation and vice versa. This was obvious because, like many of the subjects, he used hand gestures to demonstrate what he meant. Dropping this group gave us a total of 32 subjects. Questionnaires assessing each subject's computer background and video game playing experience were distributed and filled in at the completion of the study.

Answers to the demographic questions show that our subject population had a solid computer background, as well as significant experience with video games. Twenty subjects (62.5%) majored in computer science or an engineering discipline. All subjects had computer experience, with an average of 10 years. Also, a majority were video game aficionados, with an average of 10.6 years experience (see Figure 4).

Three subjects had less than five years of experience in playing video games, but their practice task performance was better or equal to the average of the subject population. Most subjects spent less than an hour per week playing video games, but three played more than 10 hours per week (see Figure 5).

We advertised for people with prior video game and computer experience in order to obtain subjects with the spatial skills needed for the Slow Tetris game. In doing so, we also obtained highly motivated subjects. The results shown in Figures 4 and 5 indicate that our subject population had the required skills.

4.3 Technical Setup

The hardware wireless setting for our experiment is shown in Figure 6. The server for collaboration and the *cWorld* (PC) client run on two IBM ThinkPad

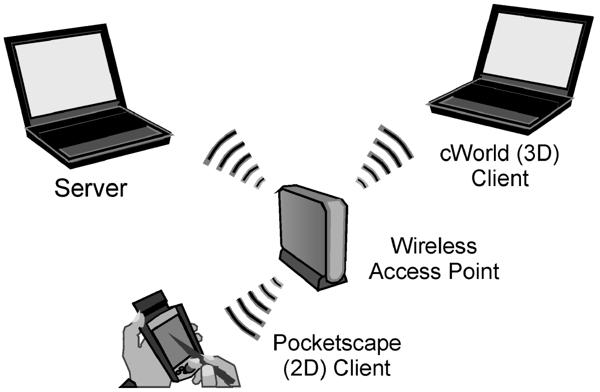


Fig. 6. The experiment setup.

Table II. Device differences in the two platforms.

Property	PC (Laptop)	PDA
Screen Size	11.25″ × 8.50″	3.02″ × 2.26″
Screen Resolution	1024 × 768	320 × 240
Input device	Computer mouse	Stylus and buttons
Network speed	11Mbps	11Mbps
Visualization	3D graphics	2D graphics

T23 notebooks with embedded Intel wireless cards. *Pocketscape* runs on a Compaq iPAQ PocketPC 3536 handheld with 32MB of memory and running the SavaJe XE operating system [SavaJe 2002]. PocketPCs had Orinoco Gold 802.11 wireless LAN cards (from Lucent Technologies). The entire system communicates through a Sony Vaio PCWA-A100 wireless LAN access point. The maximum communication bandwidth in our setting was 11Mbps.

Table II shows the variations in physical properties of the two collaborative platforms.

4.4 Procedure

The experiment involved three steps. First, we trained the subjects individually on both PC and PDA platforms using a set of progressively harder tasks. This training lasted for several weeks and consisted of subjects being asked individually to learn both the PC and PDA versions of the Slow Tetris game. After training was completed, we formed groups of four from subjects who had approximately equal completion times in the 3D-practice tasks. Our second step was to run the experiment using the groups we had formed. Table III gives the subjects' mean practice task completion times and the ranges between the means for each four-person group.

The groups were randomly assigned to one of eight trials that balanced the presentation order of the collaboration combinations. Each group member was assigned randomly to one of the following settings (Table IV).

Each user assumed either a communicator or a doer role and was only assigned to one platform. Although we trained our subjects on both platforms so that they would know how both platforms worked in the collaboration task,

Table III. Average practice completion time and range in completion times for subjects assigned to each four-person group.

	3D Avg Times (sec)	Range	2D Avg Times (sec)	Range
Group 1	47	39–53	107	62–286
Group 2	33	20–45	86	50–120
Group 3	47	35–66	88	38–116
Group 4	48	35–55	106	87–126
Group 5	23	10–30	40	31–54
Group 6	42	27–65	89	81–109
Group 7	23	18–30	37	26–46
Group 8	33	28–46	90	62–161

Table IV. Subject assignment to the different platforms and roles.

Communicators.	Doers.
Person A: PC	Person B: PC
Person C: PDA	Person D: PDA

they were assigned to one platform in the experiment. For example, Person A was randomly assigned to a PC and to a communicator role. Thus, Person A engaged in two experiment trials, one for the condition PC→PC and one for the condition PC→PDA. In both trials, Person A was a communicator and used a PC. Communicators were given the role of telling the doer what actions to take to solve the Slow Tetris problem. The doer was given the role of rotating and fitting the blocks. Our instructions told the doer that he could make rotation and fit suggestions to the communicator. Both members of the team were instructed to build the wall as quickly as possible. Each group of subjects was run separately for the four conditions in the experiment. We ran the individual conditions as follows.

Each team member was assigned a separate office for the Slow Tetris task. They communicated with each other by speakerphones. The doer made all moves. Our collaboration system instantly displayed changes on both screens. The doer and communicator were introduced to each other via the telephone although they had met briefly before. One experimenter was in each room making sure the experiment ran correctly. We only videotaped one subject with the video focusing on the screen display, not the subject. The third step of our experiment was the administration of the subject questionnaires, which was discussed earlier in Section 4.2 of this article.

4.5 Measures

Three types of measures were captured in this study. They included a video record of the doer's screen for all pairs of subjects. We also recorded (via a logging program) the block placement times for each subject during the practice tasks, the block placement times and the number of rotations for each pair during the study. In addition, we collected data via a post-study questionnaire on each subject's prior computer and virtual reality experience and the perception each subject had of the tasks and their partners.

Table V. Average time-per-block placement and standard deviations for each of the four collaboration environments.

Platform Configuration	Mean (sec)	Std. Dev. (sec)	N
PC→PC	52	50	28
PC→PDA	69	74	28
PDA→PC	111	120	28
PDA→PDA	105	92	28

Four experimenters coded the communication exchanges between each computer platform pair for a post-study analysis of the verbal transcriptions.

5. RESULTS

We present the results of our study as follows. In Section 5.1 we present the performance measures that we took on each pair's placement of the Slow Tetris blocks. In Section 5.2, we present descriptions of the communication patterns we coded. Section 5.3 provides the results from our communication coding. In the final section, we present the results from the questions that assessed each subject's perception of the collaboration task.

5.1 Performance

Although we intended to use only the logging times for our analysis, we needed to hand-code the initial block placement in 9 out of 112 instances because the subjects engaged in other than problem-solving and partner-to-partner communication behavior. For example, they did not talk with each other about the common-grounding problem, but turned to the experimenter to try to resolve their differences. Subjects were directed to talk with their collaborator, but we felt it appropriate not to count this time in the experiment time.

A Pearson correlation study revealed that the average time and the number of rotations used to fit a block are significantly correlated (Pearson $r = 0.811$, $p < 0.01$). Further inspection showed that a high number of rotations did not always correspond to the higher values registered for block fitting time. High rotation values were often associated with situations in which the communicator had trouble figuring out the solution, but they also occurred as a result of different problem-solving strategies. For example, subjects often rotated a wall piece four consecutive times in each view to help identify the shape of the block before they started to solve the problem.

The performance times came out in the order we predicted except for one case. Our prediction was as follows:

$$\text{PC}\rightarrow\text{PC} \gg \text{PC}\rightarrow\text{PDA} > \text{PDA}\rightarrow\text{PC} > \text{PDA}\rightarrow\text{PDA}$$

The average time-per-block placement in the PC→PC collaboration was the fastest, but only 17 seconds faster than the PC→PDA collaboration. The PDA→PDA collaboration was slightly faster than the PDA→PC collaboration (see Table V). This is a reverse of the order we predicted, but the time differences are smaller than the standard deviations and are not significant.

Table V shows that, as we predicted, the advantages of the 3D-environment helped its users to rapidly solve the Slow Tetris tasks. And, as predicted, the

Table VI. Tests of between-subjects effects.

Source	F	Significance	Observed Power[†]
Corrected Model	1.384	0.166	0.804
Intercept	2.752	0.100	0.375
Platforms	2.952	**0.037**[*]	0.684
Order in Trial	0.783	0.506	0.213
Platforms × Order in Trial	1.028	0.424	0.482

[*] Statistical significance.
[†] Computed using alpha = 0.05.

Table VII. Mean square values, F's and T's used to test contrasts.

	MS Contrast	F[*]	p
Question 1	3703	0.483	0.489
Question 2	63175	8.243	**0.005**
Question 3	847	0.111	0.740

[*] Based on MS of Error = 7663.69

difficulties in the heterogeneous communication caused this collaboration to be slower.

The average performance times were analyzed with a factorial analysis of variance, with the average time in the practice session of the communicator as covariate. We found the block placement times to be significantly different for collaboration pairs $F(3, 112) = 2.95$, $p < 0.05$ (see Table VI). No significant order effects were found.

We carried out a contrast analysis [Rosenthal and Rosnow 1985], to address the following research questions:

Q_1: Is performance time in homogeneous environments different from performance time in heterogeneous environments?

(PC→PC & PDA→PDA) vs. (PC→PDA & PDA→PC)

Q_2: Is performance time with a communicator on the PC platform different from performance time with a communicator on the PDA platform?

(PC→PDA & PC→PC) vs. (PDA→PC & PDA→PDA)

Q_3: Is performance time with a doer on the PDA platform different from performance time with a doer on a PC platform?

(PC→PDA & PDA→PDA) vs. (PC→PC & PDA→PC)

Table VII gives the results from this analysis. Only Question 2 found significant time differences between the communication factors assigned to each platform $F(1, 95) = 8.243$, $p \leq 0.005$.

A pairwise comparison of the collaboration pairs (see Table VIII) showed that the PC→PC collaboration was significantly faster than both the PDA→PDA ($p \leq 0.012$), and the PDA→PC ($p \leq 0.016$) collaborations. The significant time differences we found between the collaboration pairs correlate with the communication differences we found. These communication patterns are discussed in the next section.

Table VIII. Mean differences in performance time based on estimated marginal means. The times are seconds-per-block placement.

Communication Configurations	PC→PC	PC→PDA	PDA→PC	PDA→PDA
PC→PC	—	−21.552	**−62.915***	**−59.790***
PC→PDA	21.552	—	−41.363	−38.238
PDA→PC	**62.915***	41.363	—	3.125
PDA→PDA	**59.790***	38.238	−3.125	—

*The mean difference is statistically significant.

5.2 Communication

We transcribed a total of 32 collaborations on the Slow Tetris task. Each collaboration took the subject pairs approximately 20 to 30 minutes to complete. The transcriptions were done with two listeners so that we could accurately capture what was said by both collaborators, and also what actions the doer took to rotate and fit the blocks in the game.

The conversations taking place in all of the collaborations were noticeably different because of large subject differences (some subjects were more outspoken than others and some subjects were better communicators). The subjects' exchanges exhibited patterns similar to *adjacency pairs* [Luff et al. 1990]. Such pairs are commonly found in general conversational behavior, for example, a greeting followed by another greeting, or a request followed by an acceptance or a refusal. The basic paired action found in the conversations was an instruction followed by an action in the virtual environment with or without a reply associated with the action. As a consequence of the role asymmetry, the user with the communicator role is the one that usually initiates the exchange by giving an instruction, although in some occasions the doer directed the communication.

We did *not* use two of the most popular conflict-coding schemes in communication and psychology. They are the Verbal Tactics Coding Scheme (VTCS) [Sillars 1986] and the Marital Interaction Coding Scheme (MICS) [Weiss and Summers 1983]. Both of these schemes deal with interpersonal conflict which was not our focus in this study.

Behavior (coding) units are typically small intervals of conversation [Weiss et al. 1973], speaker turns [Krokoff et al. 1989], and thought turns [Sillars 1986]. We use a unit of behavior that is tied to our problem, which we call an *action unit*. We define an action unit as a continuous segment of conversation that has a direct relation to the action being performed on one of the blocks in the problem. An action unit always includes an action. (Occasionally an action unit will involve a set of consecutive actions if the communicator asks the doer to do multiple actions in a single request, or if a repair of an executed action is required.) Different action units are illustrated in Table IX.

One researcher in our team developed the communication categories to be coded by observing one fourth of the videotapes. A coding scheme for categorizing each of the action units was then written up and given to four other researchers who coded all of the subjects' action pairs while looking at the transcripts and the videos.

Table IX. Segments of conversations representing action units for a portion of a block placement. The action is shown in bold and surrounded by angle brackets.

Action Units	Speaker	Speech (PDA→PC Collaboration)
Action Unit 1	Doer	Okay. (long pause) Should I try fitting the top view?
	Comm	Oohh, let me see . . . how about the front view rotate once. . .
	Doer	Uuhh, front view rotate once <*doer rotates*> okay. . .
Action Unit 2	Comm	Top view rotate once. . .
	Doer Comm	In top view? Yeah.
	Doer	<*doer does rotations*> Okay.
Action Unit 3	Comm	Now front view rotate twice. . . and fit. . . no three times sorry. . .
	Doer	Yeah. <*doer does three front rotations; it works*> Okay, done.
Action Unit 4	Comm	Okay. (long pause) Front view and rotate three times. . . <*doer does rotations*>

Five main categories were identified: Collaboration, Authority Conflict, Affirmative Feedback, No Feedback, and Feedback. Collaboration refers to statements that seek a two-way solution to the problem. Authority Conflict remarks are weakly mitigated statements in search of gaining control over the problem solution. Affirmative Feedback expresses support for the actions suggested by the partner. No Feedback statements provide little or no feedback information about the actions requested or executed. Feedback is simply confirmation that the request has been heard and understood often through a simple repeat of the request.

The five major categories were divided into nine subcategories: Collaborative, Instruction Communication, Decision Approval, Takeover, Validation, Clarification, Repairing, Description, and One-Sided. These subcategories belong to our five major categories as follows: Collaboration (Collaborative, Clarification); Authority Conflict (Takeover, Instruction Communication); Affirmative Feedback (Decision Approval), No Feedback (One-Sided); and Feedback (Description, Validation, Repairing). Common Grounding was coded separately because it transcended all of the above categories. Table X presents our coding structure. We used Brown and Levinson [1979] extensively to recognize variations in negative and positive politeness. Positive politeness was used to identify Collaborative and Clarification communication. Negative politeness was used to identify Takeover and Instruction Communication. Sections 5.2.1 through 5.2.10 describe each category in detail and provide a representative example of the identifying communication.

5.2.1 *Decision Approval.* In this form of communication exchange, the communicator gives the directions for moving the block and receives confirmatory feedback from the doer, that is, the doer expresses his agreement with the proposed movement. Often the agreement is emotive. For example, the doer says “Ok-a-a-a-ay!” instead of “Okay.” At other times, the doer is more active in expressing approval by stating positive value words such as “Good!” “Nice!” or “That’s it!” Table XI presents an example of this type of collaboration. We believe it occurs because both team members understand the problem and both perceive the steps towards solution.

Table X. Coding categories and code.

Main Categories	Subcategories	Code
Affirmative Feedback	Decision Approval	DA
Authority Conflict	Instruction Communication	I
	Takeover	T
Collaboration	Clarification	CL
	Collaborative	C
Feedback	Description	D
	Repairing	R
	Validation	V
No Feedback	One-Sided	O
Content Categories	Common Grounding	CG

Table XI. Example of decision approval communication.

Speaker	Speech (PC→PDA Collaboration)
Comm	Aahh, the red object...
Doer	Yeah...
Comm	...top view, anticlockwise
Doer	Top view, anticlockwise. <*doer rotates the top view once*> Fine.
Comm	Okay, front view, anticlockwise <*doer rotates the front view once*>... once more.
	<*doer rotates the front view once*> Fit it.
Doer	<*doer tries to fit it and it works*> That's it! Nice.

Table XII. Example of instruction communication exchange.

Speaker	Speech (PDA→PC Collaboration)
Comm	Okay.
Doer	I think we have to turn it in top view now.
Comm	Okay, uuuhhh, alright... top view rotate once <*doer does top rotation*> hmmm ...
Doer	One more time I think.
Comm	Okay. <*doer does one more time in top*>
Comm	Okay... try to fit it... <*fit works*> Okay...

5.2.2 *Instruction Communication.* In this form of communication exchange, the doer tells the communicator what actions should be performed. The utterances are not mitigated, but direct. The communication exchanges are similar to takeover in terms of politeness, but in this case, the instructions are not executed until approval from the communicator is received.

The communicator preserves his authority by approving or disapproving the instructions. An example of this form of collaboration is shown in Table XII. In some cases, the communicator completely ignores the doer's instruction and his disapproval is expressed by giving the doer a different instruction.

5.2.3 *Takeover.* In this form of communication exchange, the doer counters the communicator's orders and, instead, gives his own orders and acts on them. This is different from a Collaborative scenario because the doer's communication is not mitigated, but direct. Following such a takeover, the communicator often switches roles with the doer and engages in mitigated conversation. Thus, this mode reflects a status exchange.

Table XIII. Example of takeover communication by doer.

Speaker	Speech (PDA→PC Collaboration)
Doer	<*after long pause*> How about one more top rotation?
Comm	Try it.
Comm	<*another long pause while doer looks at pieces from the left side view*> Try rotating once from the front view.
Doer	Okay, now should I try fitting it?
Comm	Okay.
Doer	No. <*doer does front rotation instead*> I rotate it once more on the front view.
Comm	That's okay, try... Rotate it three more times on the front view.
Doer	Okay... <*doer does two front rotations*> Okay.
Comm	Once more... excuse me, that's once more.
Doer	Once more?

Table XIV. Example of clarification communication.

Speaker	Speech (PDA→PC Collaboration)
Comm	One top rotation...
Doer	Front rotation?
Comm	Top rotation... <*doer does top*>

In some cases the takeover is not accepted, and the communicator begins to ask if the doer has actually carried out a requested rotation. In a few cases, the doer ignores the communicator and takes over the problem solution without engaging in further conversation. Table XIII gives an example of a takeover situation.

5.2.4 *Clarification.* In this form of communication exchange, the doer does not understand the communicator's directions. The doer repeats, in a questioning manner, what he thinks is the direction given and waits for the communicator's answer before executing any action. In most cases there is little or no feedback after the action is performed. Table XIV presents an example of this communication behavior.

Clarification statements can also occur when the instruction is incomplete. In many cases, the number of rotations the communicator is asking for is omitted. The doer, therefore, requests a numerical amount. The confusion usually happens when the previous instruction involved more than one rotation or was a request for a fit action. This type of communication behavior is shown in Table XV.

5.2.5 *Collaborative.* In this form of communication exchange, both partners try to help each other solve the task by making suggestions. Both members of the team converse using mitigated statements to convey politeness to their team member.

When the doer has an idea of how to solve the problem, he presents the idea to the communicator as a question. The communicator, in turn, transfers authority to the doer by giving his directions as a question. In some cases, the communicator actively asks the doer if he has any suggestions for the problem solution. An example of this form of collaboration is shown in Table XVI.

Table XV. Example of clarification exchange due to an incomplete instruction.

Speaker	Speech (PDA→PDA Collaboration)
Comm	Alright. Uuuhhh... alright now... on front hit the purple one... <*doer does nothing*> Did you get that?
Doer	How many times?
Comm	Hit the purple one once.
Doer	Alright, got it. <*doer rotates purple block once in front view*>

Table XVI. Example of collaborative communication.

Speaker	Speech (PDA→PC Collaboration)
Comm	I can't believe this is... Alright. Another front please.
Doer	<*doer does front rotation*> Okay... Now the top?
Comm	Yes. And the top and I think we will get it <*doer does top rotation*> Fit. <*doer does fit and fit works*> Okay.
Doer	Nice, now the last one.
Comm	Okay. Alright, Now... do ... a top.
Doer	A top. <*doer does top rotation*> Okay. Again?
Comm	Okay. Can you try to fit it? I'm not sure it'll work.
Doer	No, it won't fit it. <*doer, at first does nothing, but then tries to fit block. It does not work*>
Comm	It won't fit yet? Do another top, please.
Doer	Okay. <*doer does top rotation*>
Comm	Oh man... Now? Would it fit now?
Doer	Aaah, no. <*doer does nothing*>
Comm	Okay. Do another top.

5.2.6 *Description.* In this form of communication exchange, the person performing the actions provides feedback to the person giving the instructions after they are executed. In contrast to the affirmative statements, the description statements are not meant to be a confirmation. The doer is, in this case, announcing the successful execution of an action (e.g., whether a block fits into the wall) or, if a mistake is made, a report of the mistake.

The use of words such as "Okay" or "Done" is common and the intonation used is flat and monotonous. An example of this form of collaboration is shown in Table XVII.

5.2.7 *Repairing.* This form of communication exchange is coded when either the communicator or doer perceives that a mistake has been made and request a correction. This is the only case where we have more than one user-action coded in a single action unit. The person detecting the mistake lets his partner know that an error has been made and either asks that new actions be made or takes the corrective actions, if a doer. Table XVIII presents an example of this type of communication. In most cases, the doer is the one who executes the wrong actions, and notifies the communicator of the problem. Then he makes the corrections without waiting for additional instruction. In a few cases, the doer reports the error but lets the communicator describe the fixes to make.

5.2.8 *Validation.* During validation exchanges, the doer gives feedback to the communicator by repeating the instruction that he heard. Usually, the doer's

Table XVII. Example of description exchanges made by the doer.

Speaker	Speech (PDA→PDA Collaboration)
Comm	It works, okay, great . . . uuuhhh, okay, from front rotate once.
Doer	<*doer does rotation in the front view*> Okay.
Comm	Once more.
Doer	<*doer does another rotation in the front view*> Okay.
Comm	From top rotate once.
Doer	<*doer does rotation in the top view*> Okay.
Comm	Once more.
Doer	<*doer does another rotation in the top view*> Okay.
Comm	Again.
Doer	<*doer does again rotation in the top view*> Okay
Comm	And now try to fit.
Doer	<*doer tries to fit; it works*> Okay
Comm	Okay.

Table XVIII. Example of repairing communication.

Speaker	Speech (PDA→PDA Collaboration)
Comm	Alright, front view once.
Doer	Alright, hold on. . . I'm sorry. . . <*doer does top rotation, rotates 3 times on top and rotates once in front*> Front view once. . .
Comm	Yes, alright. . . Front view once more I guess. . . <*doer does two rotations by mistake, does two more to get to the previous position and then rotates once*>

Table XIX. Example of validation exchanges.

Speaker	Speech (PDA→PDA Collaboration)
Comm	Alright, now hit the blue piece once in the front.
Doer	Front. <*doer goes to the front view*> Alright. <*doer rotates once in the front view*>
Comm	Oh, sorry. . . hit it twi. . . hit it two more times. . .
Doer	One, two <*doer rotates two more times in the front view*> Alright.
Comm	Uhmmm, try to fit that.
Doer	Fit that. . . <*doer tries to fit; it didn't work*> No, it doesn't fit.

feedback is given before he executes the command, but he does not wait for further confirmation before making the rotation. We believe that this feedback is not only intended to let the communicator know that the directions were received and understood, but also to emphasize that the actions are going to be executed. Table XIX gives an example of validation statements made by the doer.

5.2.9 *One-Sided.* In this form of communication exchange, the communicator is the only person giving directions. The doer simply does what the communicator asks, with no feedback. The directions are all given in the imperative and there is no mitigation. In these dialogues, the communicator rarely tries to establish a common ground and there is no discussion about the task. The directions are clear enough so that the doer does not need to ask for clarification. The task is accomplished quickly and perfunctorily. Table XX gives an example of a One-Sided communication.

One could argue that a One-Sided communication is likely to occur with an individual who is normally not very communicative. However, since we observed

Table XX. Example of one-sided communication.

Speaker	Speech (PC→PDA Collaboration)
Comm	Okay, rotate once from front. <*doer does front rotation*>
Comm	Uh, let's see, uh two times from front. . . fit. <*doer does two front rotations and fits piece successfully*>
Comm	Okay, ummmm. . . rotate twice from front. <*doer does two front rotations*>
Comm	And, uh once from top. <*doer does one top rotation*>
Comm	Fit. <*doer fits piece successfully*>
Comm	Twice from front. <*doer does two front rotations*>
Comm	Ummm, once more on front. <*doer does one front rotation*>
Comm	Twice on top. <*doer does two top rotations*>
Comm	Uh, twice from front. <*doer does two front rotations*>
Comm	Once on top. Fit. <*doer does one top rotation and fits piece successfully*>

Table XXI. Count and percentage of one-sided communications for doer in each session. Arrows indicate those doers who tended not to participate in either collaboration.

Experiment number and doer's device	Homogeneous Collaboration	Heterogeneous Collaboration
Exp. 1 PC doer	2–25%	29–63%
Exp. 1 PDA doer	4–15%	3–27%
Exp. 2 PC doer	1–10%	3–11%
Exp. 2 PDA doer ⇒	**11–65%**	**26–74%**
Exp. 3 PC doer	1–6%	3–9%
Exp. 3 PDA doer	13–48%	12–100%
Exp. 4 PC doer	12–92%	11–50%
Exp. 4 PDA doer ⇒	**27–90%**	**20–87%**
Exp. 5 PC doer	20–61%	7–50%
Exp. 5 PDA doer ⇒	**56–88%**	**30–97%**
Exp. 6 PC doer	22–52%	4–16%
Exp. 6 PDA doer	0–0%	0–0%
Exp. 7 PC doer	35–88%	7–20%
Exp. 7 PDA doer	8–57%	10–59%
Exp. 8 PC doer	4–15%	2–20%
Exp. 8 PDA doer ⇒	**21–72%**	**26–90%**

each individual performing in two collaboration combinations, we could then ascertain if the person in the doer role did not speak in both pairings. The values marked in Table XXI show those collaborations in which more than 60% of the communication exchanges were One-Sided for both collaboration sessions. Only 4 out of 16 doers had a high number of One-Sided exchanges in both sessions. All of the instances of One-Sided collaboration belong to PDA doers. This suggests that it is the difficulty of the representation and not the quietness of the individual that caused One-Sided collaborations to occur.

5.2.10 *Common Grounding.* In the heterogeneous collaborative pairs, we observed conversational exchanges that attempted to establish common reference terms between the collaborators and a common understanding of the actions being requested by the communicator. All subjects had been trained on both the PC (3D-version) and the PDA (2D-version). Nevertheless, when the

Table XXII. Example of both subjects trying to establish a common ground across their different environments.

Speaker	Speech (PDA→PC Collaboration)
Comm	I need to get to a little understanding here. I have a 2D-display and you have a 3D-display...
Doer	Aha.
Comm	... and I need to know to match my rotations to yours.
Doer	Uuhhhh ...
Comm	Okay ... You have two buttons, right?
Doer	Yeah.
Comm	Uuhhh, a left and a right button for rotations?
Doer	Uuhhh. ... .yeah.
Comm	Do a left rotation on the right component... so I can see what happens.
Doer	<*doer does nothing*> Uuh, do I rotate the top or do I rotate just the entire component?
Comm	Aaahhh, just select the left rotation button and then click on the red block ...

time came to communicate the rotation actions, the communicator had considerable difficulty. Table XXII gives an excerpt of just such a communication. The misunderstanding actually goes on for longer than is shown in the table.

A key problem the subjects had in communicating across the heterogeneous environments was in coming up with some form of common terminology for the rotations. Various linguistic forms were used to instruct one's partner in how to rotate the block to be fitted. They included the following for rotating the top view (yaw):

Rotate top
Rotate top anticlockwise
Rotate top counterclockwise
Top once
Rotate around y-axis
Rotate horizontal
Left button
Hit top once

5.3 Communication Coding Results

Our communication coding expert established the ten coding categories and wrote coding descriptions for four other researchers who coded all of the collaboration pairs. We coded each action unit as one of the categories shown in Table X. In the cases where the coder was unsure of what code to assign to an action unit, the unit was coded as U (Undefined).

An analysis of the coding reliability indicated the existence of interpretative differences attributed to the Decision Approval (DA) and Description (D) coding. Coders 1 and 4 coded these statements as DA and coders 2 and 3 coded them as D. Since these two categories were highly similar, we collapsed them into a single category labeled Description (D). The main category Affirmative Feedback was merged with the main category Feedback for similar reasons.

Cohen's Kappa coefficient [Kraemer 1982] was used to determine the reliability within and between coders. The resulting kappa values were analyzed using the kappa interpretation scale suggested by Landis et al. [1977]. According to this scale, kappa values higher than 0.81 show "Almost Perfect" agreement; values between 0.61 and 0.8 represent "Substantial" reliability; kappa

Table XXIII. Intracoder reliability.

Coder	Cohen's Kappa
Coder 1	0.777
Coder 2	0.677
Coder 3	0.894
Coder 4	0.632

Table XXIV. Cohen's Kappa coefficients for coder pairs.

Coder Pairs	Cohen's Kappa
Coder 1 and Coder 2	0.680
Coder 1 and Coder 3	0.761
Coder 1 and Coder 4	0.621
Coder 2 and Coder 3	0.660
Coder 2 and Coder 4	0.577
Coder 3 and Coder 4	0.628

values between 0.41 and 0.6 indicate "Moderate" agreement; values ranging from 0.21 to 0.4 represent "Fair" reliability; kappa values between 0.00 and 0.2 are regarded as showing only a "Slight" level of agreement; and finally, "Poor" agreement is shown by kappa values below 0.00.

In order to calculate the intracoder reliability, the expert that created the coding scheme coded each action unit, and these results were compared to the coders' scores. The reliability measures within coders are shown in Table XXIII.

Coder 3 presented "Almost Perfect" agreement. "Substantial" agreement was achieved by Coders 1, 2 and 4. Coder 4 exhibited the lowest within reliability value.

We calculated a Cohen's Kappa to determine intercoder reliability. The results of this analysis for each experimenter pair are shown in Table XXIV.

The strongest agreement measures were found between coders 1 and 2, coders 1 and 3, and coders 2 and 3. In contrast, the weakest correlations were found between coder 4 and coders 1, 2 and 3. Investigation into the discrepancies between coder 4 and the other coders revealed that the participation of coder 4 as an experimenter in the studies influenced the coding assignments. Coder 4 used information about the subjects' personality and mood as the main rationale behind coding decisions. We decided not to use coder 4 further in the data analysis because additional knowledge beyond the coding rules was being used in the coding decisions.

Cohen's Kappa coefficient was used again to verify the reliability of the nine coding categories. Table XXV shows the kappa scores obtained in this test. Those kappa values in the "Almost Perfect" or "Substantial" level of agreement are considered to be an acceptable indicative of agreement above chance level.

Two categories, Description (D) and One-Sided (O), obtained "Substantial" or "Almost Perfect" levels of agreement with all three coder pairs. Repairing, Take-Over, and Validation attained a "Substantial" level of agreement (kappa values of 0.61, 0.62 and 0.71, respectively) with at least one of the pairs. The

Table XXV. Coding categories reliability for pairs of coders. Kappa values ≥.61 are considered satisfactory.

Coder Pairs	C Collaborative	CL Clarification	D Description	I Instruction Communication	O One-Sided	R Repairing	T Take-Over	V Validation	U Undefined*
Coder 1 & Coder 2	0.366	0.195	**0.789**	−0.006	**0.846**	0.332	0.530	**0.709**	0.088
Coder 1 & Coder 3	0.464	0.563	**0.826**	0.275	**0.885**	**0.612**	**0.618**	0.240	
Coder 2 & Coder 3	0.254	0.088	**0.804**	0.252	**0.863**	0.220	0.546	0.190	
Coder 3 & Expert	**0.873**	**0.874**	**0.892**	**0.808**	**0.956**	**0.703**	**0.911**	0.366	

*Expert and Coder 3 did not have units coded as undefined.

Table XXVI. Cohen's Kappa reliability scores for common grounding coding.

Coder Pairs	Cohen's Kappa
Coder 1 and Coder 2	0.680
Coder 1 and Coder 3	0.761
Coder 1 and Coder 4	0.621
Coder 2 and Coder 3	0.660
Coder 2 and Coder 4	0.577
Coder 3 and Coder 4	0.628

remaining coding pairs achieved Take-Over kappa values indicating "Moderate to Substantial" agreement (range: 0.55 to 0.6), Repairing kappa results were mostly "Fair" (0.22, 0.33, and 0.61), and Validation kappa values for two of the pairs of coders indicated "Slight to Fair" agreement. Clarification and Collaborative categories—both part of the main category Collaboration—were coded with, at most, "Moderate" levels of agreement. The results from two of the pairs of coders in the Clarification category indicated "Slight" agreement, while the third pair value was in the "Moderate" range. The kappa values for Collaborative ranged from "Fair to Moderate." The results from all the pairs coding the Instruction Communication category varied from "Poor to Fair," indicating a low level of agreement.

We also coded Common Grounding, independent of the other categories. The reliability of the Common Grounding codings is shown in Table XXVI. The reliabilities are extremely low, most likely due to the low number of instances found for Common Grounding. The low number and the coding reliabilities make this category too unreliable to do further evaluation with.

The scores from the most reliable coder (See the last row of Table XXV) were used to calculate the distribution of all the communication types that occurred in each platform pairing. The counts and row, column and total percentages for each coding category are shown in Table XXVII. As previously stated, Collaborative and Clarification did not reach acceptable reliability levels. Only three coded categories achieved acceptable reliability: Description (part of the Feedback main category), One-Sided (part of No Feedback), and Takeover (part of Authority conflict).

Table XXVII. Distribution of collaboration patterns across the platform pairs considering coding subcategories.

Platform Configuration		Coding subcategories			Total
		D Description	O One-Sided	T Takeover	
PC→PC Mean: 52	Count	59	99	5	163
	% within Platform	36.2%	60.7%	3.1%	100%
	% within categories	22.8%	22.6%	23.8%	22.7%
	% of total	8.2%	13.8%	0.7%	22.7%
PC→PDA Mean: 105	Count	62	123	2	187
	% within Platform	33.2%	65.8%	1.1%	100%
	% within categories	23.9%	28.1%	9.5%	26.0%
	% of total	8.6%	17.1%	0.3%	26.0%
PDA→PC Mean: 111	Count	81	72	8	161
	% within Platform	50.3%	44.7%	5.0%	100%
	% within categories	31.3%	16.4%	38.1%	22.4%
	% of total	11.3%	10.0%	1.1%	22.4%
PDA→PDA Mean: 69	Count	57	144	6	207
	% within Platform	27.5%	69.6%	2.9%	100%
	% within categories	22.0%	32.9%	28.6%	28.8%
	% of total	7.9%	20.1%	0.8%	28.8%
Total	Count	259	438	21	718
	% within Platform	36.1%	61.0%	2.9%	100%
	% within categories	100%	100%	100%	100%
	% of total	36.1%	61.0%	2.9%	100%

Table XXVIII. Distribution of collaboration patterns across the homogenous and heterogeneous platform pairs.

	Description	One-Sided	Takeover
Homogenous	116 (45%)	243 (55%)	11 (52%)
Heterogeneous	143 (55%)	195 (45%)	10 (48%)

We ran a Pearson χ^2 test of association to determine if the distribution of the coded events is statistically different from a chance distribution. The results were significant ($\chi^2 = 28.4$, df $= 6$, $p < 0.001$). We then ran a Pearson χ^2 test of association to evaluate different platform combinations. We conducted three such tests:

(1) Heterogeneous Platforms (PDA→PC & PC→PDA) vs. Homogenous Platforms (PC→PC & PDA→PDA). Table XXVIII shows the contingency table for this comparison.
(2) PDA communicator (PDA→PC & PDA→PDA) vs. PC communicator (PC→PDA & PC→PC). Table XXIX shows the contingency table for this comparison.
(3) PDA doer (PC→PDA & PDA→PDA) vs. PC doer (PDA→PC & PC→PC). Table XXX shows the contingency table for this comparison.

Table XXIX. Distribution of collaboration patterns across the PDA and PC communicator platform pairs.

	Description	One-Sided	Takeover
PDA comm	138 (53%)	216 (49%)	14 (67%)
PC comm	121 (47%)	222 (51%)	7 (33%)

Table XXX. Distribution of collaboration patterns across the PDA and PC doer platform pairs.

	Description	One-Sided	Takeover
PDA doer	119 (46%)	267 (61%)	8 (38%)
PC doer	140 (54%)	171 (39%)	13 (62%)

Table XXXI. Results of χ^2 measures of association for communication patterns.

Type of Comparison	χ^2 value	df	p value
Hetero vs. Homo	7.455	2	0.024
PDA comm vs. PC comm	3.082	2	0.214
PDA doer vs. PC doer	17.274	2	<0.001

The reader may note that the above three tests are similar to the linear contrasts presented earlier on the performance time results. The results are shown below in Table XXXI. They differ from the performance time analysis in that comparisons 1 and 3 are significant and comparison 2 is not significant. This is directly opposite the results uncovered with the linear contrast analysis. This suggests that the performance times are not as tightly tied to communication differences as we had initially anticipated.

As can be seen from the contingency tables for the three platform comparisons, the homogenous and heterogeneous communication patterns differ both in the amount of description (five percent more in the heterogeneous collaborations) and in the number of one-sided events (five percent more in the homogenous collaborations). There appear to be no takeover differences between homogenous and heterogeneous arrangements, but we note that takeover differences existed both in the PDA communicator vs. PC communicator and in the PDA doer vs. PC doer comparisons. If we look at the Table XXVI (the overall contingency table), we see that takeovers occur whenever platforms are equal, or when the doer has a better platform than his communicator.

In the PDA doer vs. PC doer contingency table, we find that the significant differences are due primarily to a large amount of one-sided communication when the person taking commands has a PDA (22% more). We see that this occurs in both the homogenous and heterogeneous platform conditions indicating that the person in the doer role followed the assigned role when the problem representation was more difficult. The takeovers are also larger for the PC doer. This difference stems mostly from the PDA→PC combination. More takeovers are seen in all cases where the problem representation for the doer is easier, that is, on a PC platform.

Table XXXII. Distribution of common grounding patterns across the computer platform pairs.

Platform	Total
PC→PC	2
PC→PDA	1
PDA→PC	4
PDA→PDA	2

If we examine Table XXVI, the overall contingency table, we see that One-Sided communications were the most common type of utterances found (61%). They occurred primarily in the platforms where the doer had the less powerful platform: PC→PDA and PDA→PDA. Description statements were the second most common type of communication with 36% of the scores. The heterogeneous platforms conversations presented the highest numbers of Description statements (83.5%), and 50.3% of Description utterances were found in the PDA→PC platform combination. Takeovers were found to occur mostly in the PDA→PC, PDA→PDA and PC→PC platform combinations. Takeovers were so few (only 2.9% of the total) that they could not be considered to reliably make a difference in the collaborations.

Tables XXXII presents the counts for the Common Grounding category in combination with the subcategories in which it was coded. A total of nine action units were coded as Common Grounding, five of them were found on the heterogeneous platforms. The PDA→PC had one, and PC→PDA had four Common Grounding codings; however, they accounted for only a small percentage of the action units coded, respectively. In most of the cases, the Common Grounding occurred in the first two to five action units coded.

5.4 Questionnaire Analysis

After subjects completed all wall-building tasks, a questionnaire was distributed individually to each subject. The questionnaire elicited subjects' attitudes towards the user interfaces of the platforms they were assigned to and the two collaboration scenarios they participated in. We also obtained demographic information on the subjects, which we discussed in Section 4.2 of the article where we characterized our subject population. All 32 subjects completed the questionnaire, but not all questions were answered.

The subjects were asked to evaluate both applications on a 7-point Likert scale, where 1 equaled "strongly disagree" and 7 equaled "strongly agree." Averages of their answers were calculated for each evaluation question. Figure 7 graphs these averages. The graphs show that the subjects liked the 3D-better than the 2D-application. This is also supported by the written comments they provided on the questionnaire. Twenty-one subjects gave positive evaluations of the 3D-application using such terms as "clear," "convenient", or "nice." Six subjects indicated that the 3D-application was "fun" or "enjoyable." When comparing 3D to 2D, the subjects wrote that the 3D-application required less time and effort to use comfortably. When asked to comment on the 2D-application,

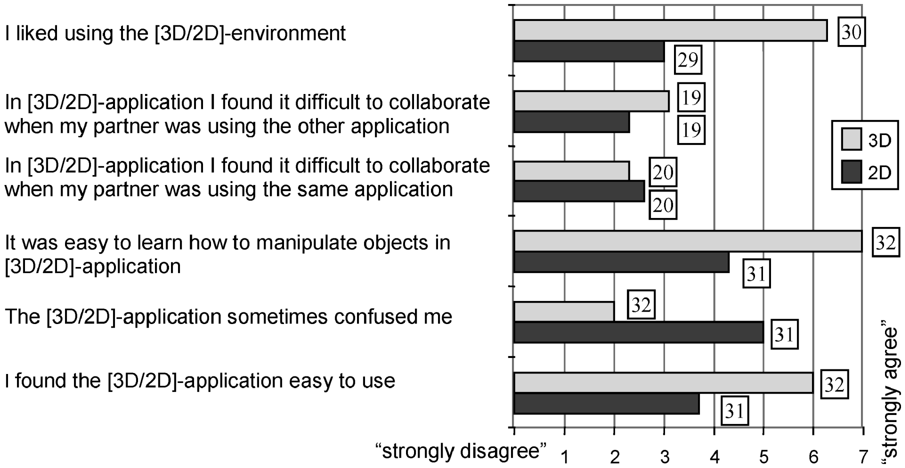


Fig. 7. A comparison of subjects' attitudes towards the 3D- and 2D-applications. The questions that were asked are to the left of the bar graphs. The stripe bar pattern represents the subjects' 3D-response, and the gray bar pattern represents the subjects' 2D-response. The Likert scale that shows the average scores of the questions is situated below the bar graphs. The numbers at the end of the individual bars show how many subjects responded to the particular question.

22 subjects indicated that they were confused while using it. They reported that the 2D-application made it difficult to manipulate blocks and that it was hard to get used to.

Eight subjects expressed positive opinions about the 2D-application, indicating that it was easy to figure out rotations or that they considered it to be a "simple and intuitive interface." Three of the subjects thought that the 2D-version was fun, while one said it reminded him of solving jigsaw puzzles. This positive attribution may have come from the relatively long experience in playing video games for this group of subjects, 10.5 years on average.

Figure 8 shows how subjects evaluated the heterogeneous and homogenous collaborations they participated in. Again, the subjects were asked to give their assessment for each printed statement on a scale from 1 to 7, where 1 equaled "strongly disagree" and 7 equaled "strongly agree." The bar graphs in Figure 8 show that subjects found the heterogeneous collaborations to be more difficult than the homogeneous ones. The subjects generally found it more difficult to communicate with a partner who had a 2D-environment (questions 1 and 4). It was particularly difficult for the 3D-doer to receive guidance from a 2D-communicator (question 1). A less strong but similar feeling was expressed by the 2D-communicator (question 3). If we compare questions 1 and 2, it appears that the 3D-communicator found it slightly more difficult to collaborate with the 3D-doer (question 2) than with the 2D-doer (question 1). Since our communication coding showed the 2D-doer to be very passive in this combination, it may be that the 3D-communicator preferred the passivity.

Over half of the subjects skipped answering these questions. Although we pilot-tested the questionnaire, we found that subjects who were actually engaged in the study were confused by the questions shown in Figure 8. In

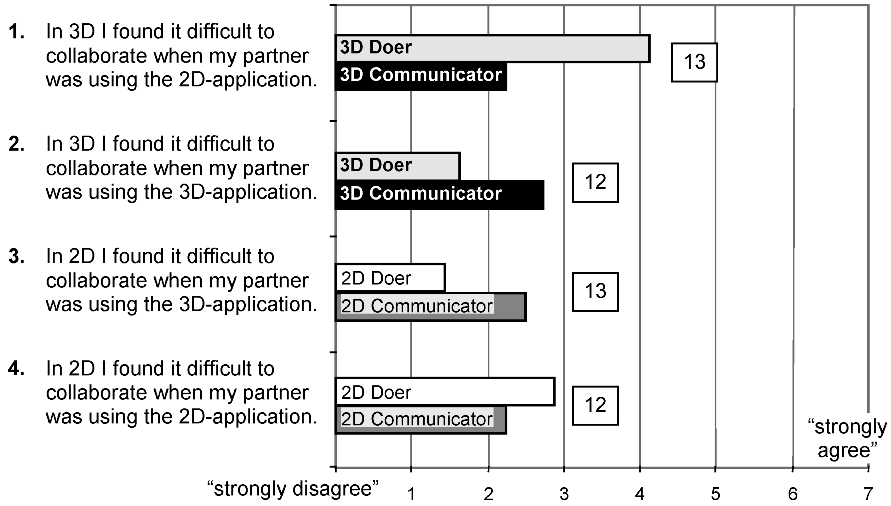


Fig. 8. A comparison of subjects' attitudes towards the heterogeneous collaboration. The questions that were asked are to the left of the bar graphs. The white bar represents answers from subjects who were PC doers. The black bar represents answers from subjects who were PC communicators. The gray bar represents answers from subjects who were PDA doers. The dotted bar represents answers from subjects who were PDA communicators. The scale that shows the average scores of the questions is situated below the bar graphs. The numbers at the end of the individual bars show how many subjects responded to the particular question.

particular, because they collaborated in both heterogeneous and homogenous combinations, they were not sure whether the "2D" and "3D" referred to them or to their partners. Subjects were assigned either a 3D-platform or a 2D-platform only. When asked about a platform they had not used, that is, in a 3D-to-2D comparison, they felt that the question was not applicable and skipped it.

T-tests run on the questionnaire results primarily showed differences between subjects' perceptions of how difficult the two platforms were to use. The following statements from Figure 7 were found to be significant at a confidence level of 95%:

—Subjects liked the 3D-application more than the 2D-application, $t(30) = 7.908$, $p < .001$.

—Subjects perceived the manipulation of objects to be easier in the 3D-application than in the 2D-application, $t(34) = 7.335$, $p < .001$.

—Subjects perceived the 2D-display more confusing, $t(34) = 6.075$, $p < .001$.

—Subjects considered the 3D-application easier to use than the 2D-application, $t(34) = 7.351$, $p < .001$.

6. DISCUSSION

What we have done in our study is give collaborators two representations of a problem-solving task. One is a poor representation (PDA) and one is a good representation (PC). We have given people authority roles in collaboration pairs assigned to solving the shared problem. The performance measurements show that when both collaborators have a good representation, they quickly and easily solve the problem, but when the communicator (person in charge) has a platform with a poor representation, then the performance times become larger. Pairwise comparisons found significant differences between PC→PC and both PDA→PDA, and PDA→PC combinations. We did not find significant differences between the PC→PC and PC→PDA combination or between the PDA→PDA and the PDA→PC combination. This differed from our initial hypotheses. In particular, we predicted that the PDA→PC combination would be significantly faster than the PDA→PDA combination. They were not, and the mean time for PDA→PC was slightly larger than the PDA→PDA combination. We also predicted that the PC→PC combination would be significantly different from the PC→PDA combination but, although the mean times were different in the predicted direction, no significant difference was uncovered. Our linear contrasts also were not able to demonstrate any significant time differences between the homogeneous and heterogeneous platform combinations. From these results, we can conclude that we were unable to find significant performance differences due to heterogeneity of the platforms. However, we do demonstrate that a good representation on both platforms (PC→PC) gives singificantly optimal performance over any platform combination in which at least one of the collaborators has a bad representation.

In summary, we conclude the following:

—*Any combination with a platform having a communicator with a poor representation negatively impacts the problem solution even if the partner has a good representation.*

—*Any heterogeneous combination with a communicator having a good representation is still somewhat negatively impacted.*

In all cases, collaborators completed their tasks, just at different rates. We did observe significant communication differences across the different platform combinations and these differences were not the same as the performance time differences. We found the largest differences when the communicator had a poor representation and their partner did not. However, the poor representation on the PDA platform affected all collaborations. In summary, we found that:

—*Any doer with a PDA had a one-sided communication.*

—*Takeovers were more common when a doer had a platform that was comparable with, or better than, that of the communicator.*

—*The communicator in heterogeneous platform combinations used more descriptive statements.*

In short, we found that if one user has a bad platform and is in charge, the communication will change this status. Decisions will be received with less than enthusiastic approval or the role of being in charge will be taken over.

What we observed in the communication patterns in the different platform combinations was different than what we initially expected. We thought that the heterogeneous combinations would, in general, have more communication problems in terms of team members disagreeing with each other. What we actually found were significant communication variations in the heterogeneous combinations. On all but the PDA→PC combination, the person who was the doer primarily took orders from the person who was the communicator. In the other combinations, the doer followed the role assigned to him and the conversation exchange was coded mostly as One-Sided. Although we did not see a large number of Takeovers, the doer in the PDA→PC pair engaged in more communication, which we interpret as the doer taking a more active role in the problem solution. However, the type of activity the doer participates in is spread across many of our communication categories, from Takeover, to Collaborative, to Decision Approval, so that these differences become too small to assess. We see similar behavior in the PC→PDA combination. In this case, the communicator engages in more description to make the moves clearer to the doer. The One-Sided conversations for the PC→PC and PC→PDA combinations were approximately equal, but the PC→PC combination had much less Description communications.

We found common grounding differences between the heterogeneous and homogeneous combinations only for establishing naming conventions for giving and receiving instructions. Only two incidents of Common Grounding communication were recorded in the homogeneous setups. In contrast, the PDA→PC combination had three such incidents and the PC→PDA combination had one coded incident.

—*Given the number of exchanges made in this problem-solving task, this Common Grounding communication was insignificant and we cannot conclude that it had a large impact on our communication exchange.*

This is in contrast to our initial expectations.

Although the rotation parameters set up for each representation were natural to that representation, subjects had difficulty imagining their partner's task across the different platforms. Most of the subjects ended up with a command sequence that asked for a top- or a front-rotation. This would be the only way of describing a rotation in the 2D-environment. Teams that used other descriptors had their PDA partner asking them for repetitions. Thus, although subjects established a common ground, it was not reliably the best one for both pairs.

The PDA→PC combination created a difference in apparent problem-solving skill because of the poorer representation on the PDA. When the communicator had a PDA environment and was communicating to a PC environment, we viewed considerable waiting on the PC person's side. In three of these setups, the PDA communicator would pause for 1-2 minutes before giving another command. The person on the PC side, who could envision the next needed rotation, became frustrated with the slowness of his partner. The frustration was either expressed in a lukewarm response to the solution (Description) when it arrived, or in an attempt to take over the task (Takeover communication).

In contrast to the PDA→PC pairing, the PDA→PDA communication was predominantly One-Sided as was the PC→PDA communication. In either case, the problem representation for the PDA doer is so poor that this person is content to let the partner solve the task and give directions to be followed.

The PC→PC communication had a small number of Takeover communications, possibly because the doer in this setup was slightly more proficient at solving the problems than the communicator. We believe that we would see more collaboration in the PC→PC setup if we did not assign communicator and doer roles. Even though our instructions indicated that a doer could make move suggestions, most subjects did not take on this role. If we had not assigned roles to the heterogeneous platform combinations, our data strongly suggest that the individual with the better representation would assume authority over the problem solution and that the conversation would be predominantly one of giving and receiving orders.

The results from the post-test questionnaire support our communication results. In the PDA→PC combinations, the PC doer expressed the most dissatisfaction with the combination. This was the platform combination that had the most Takeovers and Descriptions, a result of the PC doer trying to convey moves to the PDA communicator. In all other combinations both PC and PDA doers are not very dissatisfied with the collaboration. Communicators who have a PC platform are not at all dissatisfied with the collaboration with their partners, quite likely because they are in charge and have no trouble accomplishing their task.

If we compare our communication differences to our performance time differences, we find no correlation. Any combination that has a PDA in it suffers in terms of performance time.

—*The performance time differences appear to be caused by the difficulty of using the platform with the poor representation.*

In contrast, the one communication combination that is different from all others is that of PDA→PC. This communication has the least One-Sided conversations, the most Takeovers, and the most Description.

—*The* PDA→PC *communication differences appear to be a result of the role asymmetries combined with the platform asymmetries, where the person of higher status is working on a poor platform (e.g., a manager in the field may be communicating with her secretary about appointments, using a PDA).*

Our results suggest that some of this status will be conferred on the "good" platform owner. In short, limited device capabilities can affect who is actually in charge.

We have looked at only one aspect of the types of group work that are likely to exist over future wireless networks. Our results suggest that attention has to be paid to the types of representations that are used on the mobile platform because poor representations may affect the collaboration relationship between the communicating colleagues. In particular, it may inadvertently give the person with the better problem representation significant advantage in the communication.

7. FUTURE WORK

Our key finding is that role-asymmetry, combined with platform heterogeneity, impacts collaboration. What we also observed in the communication exchanges was a flexibility of approaches as humans used the voice channel to work towards a viable exchange pattern that would help them solve the problems created by the platform differences. This suggests that changes in roles, expertise, task, and environment will uncover other collaboration impacts. We are therefore planning a series of additional studies to further explore this area. Our next experiment will continue to use the Slow Tetris game, but with simpler problems and a subject population that is not as video game-oriented. Since literature on gender differences suggests that women engage in more collaborative work practices, we plan on using female as well as male subjects. Instead of role-asymmetry, we are planning to encourage collaboration by setting up the system so that subjects will be unable to complete a problem without the aid of their partner. In addition, we will be observing what effect problem expertise has on the exchange and also assessing each team member's social attribution of their collaborative partner. Following this work, we are planning to move to text-based platforms and add a variety of collaborative support tools that have been suggested from our work in the 3D↔2D-environments.

Observing the problems the users had with the heterogeneous environments has generated a number of ideas for which we have already developed various small software tools. We briefly describe those that we consider most promising, which we intend to add to our collaboration platforms.

The first of these tools is TIWIS, which stands for "This Is What I See". Although the PDA computing power is too weak to support drawing the polygons for the 3D-display, it can readily display a 2D-screen capture of a 3D-object. TIWIS allows the user of the 3D-environment to circle any object on the screen and send it to the PDA. The object is displayed in the upper left hand corner of the PDA screen and allows the PDA user to receive a snapshot of the current 3D-view.

Since computing power is not at a premium on the PC platform, we also realize that it is relatively easy to provide the PC-communicator with a second window displaying what the person with the PDA is currently viewing. This could be labeled "partner's view". A telepointer system could also be developed to work with the partner's view so that the person on the PC could not only communicate by voice with their team member, but also provide gesture input. Throughout all of our experiment trials, we noticed our subjects using hand gestures to indicate the direction of the rotation when they talked with their partner. Being able to send this gesture electronically might have been very helpful. Such a telepointer would need to be a "semantic" telepointer similar to that developed by Greenberg et al. [1996], that is, it would need to adapt to the view being displayed. However, putting such a telepointer on the display of a PDA is problematic because of the size of the window. A telepointer, however small, would easily overshadow necessary detail in the display. We could highlight portions of the block that the 3D-person was pointing at, but this may actually add more complexity to the problem.

We have also designed a zoom tool that allows a user to select that portion of the screen they wish to enlarge. This is primarily for the PDA display. When we have more than two blocks laid out in a row for the wall-building task, the blocks become small and various features are difficult to see. Zooming will increase the number of steps the PDA user will take to solve the problem, but gaining better insight into the problem may reduce overall solution time.

Although not implemented, we have also considered making the 2D-task a scrolling task rather than a view-switching task. This may help the PDA user to better see the relationship between the top- and front-view and thus be more able to solve the problem. Scrolling on a PDA, however, adds to the complication of the user interaction with the interface.

Finally, we believe that adding interaction history to both interfaces will help users solve the problem quicker. We have also not implemented this tool, but are planning to do so with an arrow between the current state and the past state that shows the rotation that was taken. This implementation would allow the user to walk back through the rotation moves in order to better understand the path that had been taken.

The TIWIS, partner's view, and telepointer ideas are designed to help the collaboration between the two team members. The other ideas are proposed to improve the 2D-problem-solving capabilities of the PDA platform user.

ACKNOWLEDGMENTS

Dr. Maria Kozchevnikov helped with discussions on spatial tasks and Dr. Brian Whitworth helped with discussions on spatial skills of people. Dr. Erich Labouvie and Dr. James Graham guided our data analysis.

REFERENCES

ARK, W., DRYER, D. C., SELKER, T., AND ZHAI, S. 1998. Representation matters: The effect of 3D objects and a spatial metaphor in a graphical user interface. In *People and Computers XIII, Proceedings of HCI'98*, H. Johnson, N. Lawrence and C. Roast, Eds. Springer-Verlag, Heidelberg, Germany. 209–219.

BILLINGHURST, M., BEE, S., BOWSKILL, J., AND KATO, J. 1999. Asymmetries in collaborative wearable interfaces. In *Proceedings of the 3rd International Symposium on Wearable Computers*. IEEE, Piscataway, NJ. 133–140.

BROWN, P. AND LEVINSON, S. 1978. Universals in language: Politeness phenomena. In *Questions and Politeness: Strategies in Social Interaction*, E. N. Goody, Ed. Cambridge University Press, Cambridge, UK. 56–289.

CARD, S. K., MORAN, T. P., AND NEWELL, A. 1980. The keystroke-level model for user performance time with interactive systems. *Communications of the ACM 23*, 7, 396–410.

CLARK, H. H. AND BRENNAN, S. E. 1991. Grounding in communication. In *Perspectives on Socially Shared Cognition*, L. B. Resnick, J. M. Levine and S. D. Teasley, Eds. American Psychological Association, Washington, DC. 127–149.

COHEN, J. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement 20*, 1, 37–46.

DRYER, D. C., EISBACH, C., AND ARK, W. S. 1999. At what cost pervasive? A social computing view of mobile computing systems. *IBM Syst. J. 38*, 4, 652–676.

GREENBERG, S., GUTWIN, C., AND ROSEMAN, M. 1996. Semantic telepointers for groupware. In *Proceedings of the OzCHI Sixth Australian Conference on Computer-Human Interaction*. Hamilton, New Zealand. 24–27.

HINDMARSH, J., FRASER, M., HEATH, C., BENFORD, S., AND GREENHALAGH, C. 1998. Fragmented interaction: Establishing mutual orientation in virtual environments. In *Proceedings of the ACM Conference on Computer-Supported Cooperative Work*. ACM, New York. 217–226.

JONES, M., MARSDEN, G., MOHD-NASIR, N., BOONE, K., AND BUCHANAN, G. 1999. Improving Web interaction on small displays. In *Proceedings of the 8th International World Wide Web Conference*. ACM, New York.

KAMBA, T., ELSON, S. A., HARPOLD, T., STAMPER, T., AND SUKAVIRIYA, P. 1996. Using small screen space more efficiently. In *Proceedings of the Conference on Human Factors and Computing Systems*. ACM, New York. 383–390.

KRAEMER, H. C. 1982. Kappa coefficient. In *Encyclopedia of Statistical Sciences*, S. Kotz and N. L. Johnson, Eds. New York: John Wiley and Sons. 352–354.

KRAUT, R., MILLER, M. D., AND SIEGEL, J. 1996. Collaboration in performance of physical tasks: Effects on outcomes and communication. In *Proceedings of the ACM Conference on Computer-Supported Cooperative Work*. ACM, New York. 57–66.

KROKOFF, L. J., GOTTMAN, J. L., AND HASS, S. D. 1986. Validation of a global rapid couples interaction scoring system. *Behav. Assess. 11*, 65–79.

LANDIS, J. AND KOCH, G. G. 1977. The measurement of observer agreement for categorical data. *Biometrics 33*, 159–174.

LINDE, C. 1988. Who's in charge here?: Cooperative work and authority negotiation in police helicopter missions. In *Proceedings of the ACM Conference on Computer-Supported Cooperative Work*. ACM, New York. 52–64.

LOGITECH, INC. 2002. Magellan 3D-Space Mouse (Classic), Documentation. Available online at http://www.vrdepot.com/magelpg.htm.

LUFF, P., GILBERT, N., AND FROLICH, D., EDS. 1990. *Computers and Conversation*. Academic Press, London.

MACKENZIE, I. S. 1992. Fitts' law as a research and design tool in human computer interaction. *Hum. Comput. Interact. 7*, 1, 91–139.

MACKENZIE, I. S. AND SOUKOREFF, R. W. 2002. Text entry for mobile computing: Models and methods, theory and practice. *Hum. Comput. Interact. 17*, 2/3, 147–198.

MARSIC, I. 2001. An architecture for heterogeneous groupware applications. In *Proceedings of the 23rd IEEE/ACM International Conference on Software Engineering*. IEEE, Piscataway, NJ. 475–484.

MARSIC, I., KREBS, A. M., DOROHONCEANU, B., AND TREMAINE, M. 2002. Designing and examining PC to Palm collaboration. In *Proceedings of the 35th Hawaiian International Conference on System Sciences*. IEEE, Piscataway, NJ. 10pp. CD-ROM.

ROSENTHAL, R. AND ROSNOW, R. L. 1985. *Contrast Analysis: Focused Comparisons in the Analysis of Variance*. Cambridge University Press, Cambridge, UK.

SAVAJE TECHNOLOGIES. 2002. Java-based operating system for wireless devices. Documentation available online at http://www.savaje.com.

SILLARS, A. 1986. Procedures for coding interpersonal conflict: The verbal tactics coding scheme (VTCS). Department of Communication Studies, University of Montana, Missoula, MT.

STEED, A., SLATER, M., SADAGIC, A., BULLOCK, A., AND TROMP, J. 1999. Leadership and collaboration in shared virtual environments. In *Proceedings of the Virtual Reality Conference*. IEEE, Piscataway, NJ. 112–115.

STEFIK, M., BOBROW, D. G., LANNING, S., AND TATAR, D. G. 1987. WYSIWIS revised: Early experiences with multiuser interfaces. *ACM Trans. Inform. Syst. 5*, 2 (April), 147–167.

TATAR, D. G., FOSTER, G., AND BOBROW, D. G. 1991. Design for conversation: Lessons from Cognoter. *Int. J. Man-Machine Stud. 34*, 2, 185–209.

WEISS, R. L., HOPS, H., AND PATTERSON, G. R. 1973. A framework for conceptualizing marital conflict: A technology for altering it, some data for evaluating it. In *Critical Issues in Research and Practice: Proceedings of the Fourth Banff International Conference in Behavior Modification*, F. W. Clark and L. A. Hamerlynck, Eds. 309–115.

WEISS, R. L. AND SUMMERS, K. L. 1983. The marital interaction coding system-III. In *A Sourcebook of Marriage and Family Assessment*, E. E. Filsinger, Ed. Sage Publishers, Beverly Hills, CA. 85–115.

WILLIAMS, E. 1977. Experimental comparisons of face-to-face and mediated communication: A review. *Psycholog. Bull. 84*, 963–976.

WYNN, E. H. 1979. Office Conversation as an Information Medium. Ph.D. dissertation, Dept. of Anthropology, University of California, Berkeley.

Received June 2002; revised March 2003, December 2003; accepted March 2004 by Brad Myers