Proceedings of the 4th IASTED International Conference
Internet and Multimedia Systems and Applications (IMSA'2000)
November 20-23, 2000 - Las Vegas, NV, pp.309-313

COLLABORATION USING HETEROGENEOUS DEVICES – FROM 3D WORKSTATIONS TO PDA'S

ALLAN MENG KREBS, BOGDAN DOROHONCEANU, IVAN MARSIC
{krebs, dbogdan, marsic}@caip.rutgers.edu

Center for Advanced Information Processing (CAIP)
Rutgers – The State University of New Jersey
Piscataway, NJ 08854-8088, USA

Abstract

With the recent proliferation of mobile computing devices we witness an increasing demand for applications supporting collaboration among users working in the field and in the office. A key component for synchronous collaboration in this domain, is real-time sharing and manipulation of information using very different display capabilities on the various devices. Providing support for WYSIWIS (What You See Is What I see) with past systems was a difficult but not impossible task because the groupware used common platforms. Developing groupware applications that are interoperable across diverse environments is significantly more difficult and costly.

Mobile applications involving synchronous collaboration are emerging in many fields, e.g., business, healthcare, military and transportation. The classic example is the provision of just-in-time assistance between a desk-bound expert and a mobile fieldworker using a portable device. The fieldworker may work with blueprints while the expert is using a 3D CAD model to repair a vehicle or the plumbing of a building.

The heterogeneity of computing platforms manifests itself in CPU speed, memory, display capabilities, and network bandwidth, with the last two accounting for the most prominent differences. This paper focuses on display differences since current developments indicate that they are likely to be the most variable. We take a data-centric approach, where conferees share the same data or a subset of that data.

Our work on the Manifold framework supports the development of collaborative applications for heterogeneous environments, ranging from 3D environments on workstations to 2D constraint environments running on PDA's (e.g. Palm Pilots). Our approach allows clients with different capabilities to share different subsets of data in order to conserve communication bandwidth. We also illustrate, via an extreme example of size and dimensionality differences, that heterogeneous collaboration does not appreciably affect task performance and that users perceive the task performance to be equivalent to homogenous environment collaboration.

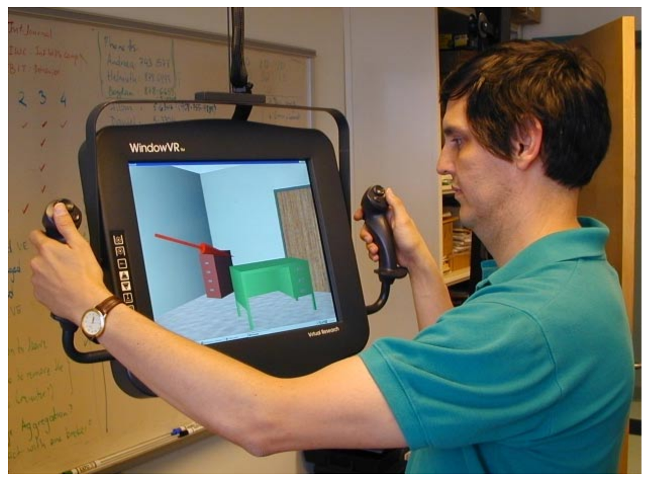

Figure 1 – Manifold based application displayed on Window VR.

Introduction

A key component for synchronous collaboration is real-time sharing and manipulation of information. Most collaborative applications provide synchronization support, the so-called WYSIWIS technique [1], which allows users to point or discuss an object on the screen that they are confident is visible to their collaborator. Providing support for WYSIWIS with past systems was a difficult but not impossible task because the groupware used common platforms. Developing groupware applications that are interoperable across diverse environments is significantly more difficult and costly. However, with recent anytime-anywhere proliferation of computing technology, support for heterogeneity is inevitable. Mobile applications involving synchronous collaboration are emerging in many fields, e.g., business, healthcare, military services and transportation. The classic example is the provision of just-in-time assistance between a desk-bound expert and a mobile fieldworker using a portable device.

The heterogeneity of computing platforms manifests itself in CPU speed, memory, display capabilities, and network bandwidth, with the last two accounting for the most prominent differences. This paper focuses on display differences since current developments indicate that they are likely to be the most variable. For example, we are seeing the development of displays mounted in eyeglasses, OptiScape technology, windshield-projected displays in cars, or foldable displays. These displays are invariably more limited in size and quality than those on the desktop are. Enlarging the screen real estate for mobile computers remains impractical because of the inconvenience of weight and size. Weight may also limit the compute power and bandwidth needed to display the graphics and animation that occurs in a desktop environment. Moreover, there is also a human information processing limitation that constrains the mobile user. A person in a private office can allocate vastly different cognitive resources to a display than a person in the field who needs to pay attention to the external environment, e.g., traffic [2]. All the above factors result in information displays that are both smaller and more limited in display capabilities for the mobile user than the stationary user. The problem then becomes one of enabling collaboration between users who have different views of the same information within an application and even run different applications.

We take a *data-centric approach*, where conferees share the same data or a subset of that data. Our Manifold framework defines a set of applications for data exchange on heterogeneous devices. We demonstrate the Manifold framework via a collaborative session involving office furniture arrangement between users with different display capabilities. Some of the users are viewing a 3D representation of the office, while other users are seeing the same office in two dimensions. In addition to providing an instantiation of our heterogeneity support, we also have evaluated the effect of the heterogeneous environment on the quality of the collaboration.

Our paper is organized as follows. We first review related work in this area. Then we overview the architecture of the Manifold framework for heterogeneous collaboration, supported by description of sample applications we have developed. Then we present results from a short study, made using these applications, which compares heterogeneous environment collaborations to homogenous environment collaborations. Finally, we conclude the paper.

Background and Related Work

The need to allow conferees to collaborate on dissimilar terminals was recognized early on by D. Engelbart, the pioneer of computer-supported collaborative work [3]. An early design for heterogeneous groupware is presented in [4], but it does not employ a model-view separation. Rendezvous [5], GroupKit [6] and several groupware toolkits thereafter use the model-view separation so that developers can create common models with different views. However, no implementation is attempted, and in some cases (e.g., [5]) the situation is greatly simplified by using centralized groupware architectures.

A recent approach to collaboration in heterogeneous computing environments is the CMU Pebbles project [7]. It is focused on single-display groupware, with the team being in a single meeting room. Multiple handheld Personal Digital Assistants (PDA's) provide simultaneous input (mouse, keyboard) to a single workstation. PDA's are not treated as equal partners in the collaboration, which belies the need for the heterogeneous data representation.

An important aspect of heterogeneity is interoperability among the groupware systems. Dewan and Sharma [8] address this issue. Another approach [9] is grounded on the premise that almost any information can be encoded as 2D pictures, and that humans readily understand such pictures. The focus is on a common pictorial surface that contains a rendering of graphical objects. Sharing between collaborators is implemented at the level of the pictorial surface (i.e., view), with PostScript as the communication standard. As the authors point out, editing the surface is cumbersome since application logic is lost in the rendering process. Our work is complementary to these efforts insofar as we focus on an architecture for developing new applications for heterogeneous environments, rather than interoperating the existing ones.

In addition to developing an architecture for heterogeneous collaborative applications, we have evaluated team performance characteristics for such applications [10]. 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 [1]. In subsequent work [11], 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.

Our essential principle for heterogeneous collaboration is that every user's action is interactively and continuously reflected in other users' workspaces, with a varying degree of accuracy or realism or through a qualitatively different visualization. Thus, the shared reference is maintained. We should also point out that non-WYSIWIS is quite common in collaborative virtual environments (CVEs) where collaborators navigate independently to accomplish their own goals (e.g., [12] and [13]).

System design

Our Manifold framework is based on the well-known *model-view-controller* design pattern, where the same model is used for implementation of applications with heterogeneous views. The application level of the Manifold was designed as a generalized editor, which can serve as support for implementing collaborative applications that can have user interfaces ranging from text/graphical editors to collaborative virtual environments.

The main characteristic of the Manifold generalized editor is the multi-tier architecture. The common three-tier architecture comprises the vertical tiers of presentation, application or domain logic, and storage. The Manifold's *presentation* tier is virtually free of the application logic and deals with visualizing the domain data and accepting the user inputs. The *domain* tier deals with the semantics of tasks and rules as well as abstract data representation. The third tier in our case comprises the collaboration functionality, which is mainly provided by the communication infrastructure, but a part of it resides in Manifold as discussed below.

The domain and the presentation layers of Manifold do not know about the communication infrastructure, but are glued together by the collaboration layer, which interacts with the communication infrastructure to send and receive collaboration events. Figure 2 shows the interactions among the layers. A local command is dispatched either simultaneously to the local domain layer and the communication infrastructure, or first to the infrastructure and then locally after the command is reflected back. Which strategy is used depends on the employed concurrency control algorithm, optimistic or pessimistic.

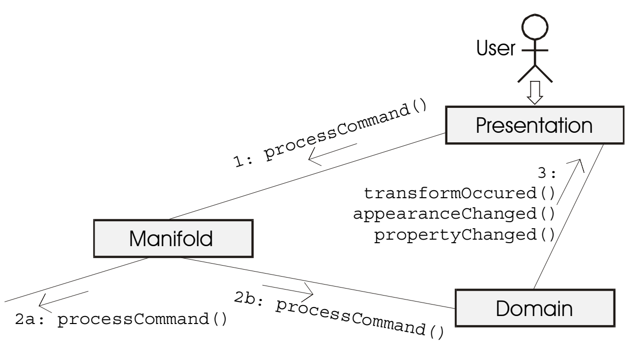

Figure 2 - Event exchange and interception in a Manifold-based application. Commands (generated by the local user) that affect the domain bean. Action 2a passes the command to the communication infrastructure, which broadcasts it to the remote peers.

Dewan [8] argues that any collaborative application can be seen as a generalized editor of semantic objects defined by it. A user interacts with the application by editing a rendering of these objects using text/graphics/multimedia-editing commands. In addition to passive editing, in a generalized editor, the changes may trigger computations or behaviors in the objects. In that sense, Manifold can be used to build arbitrary groupware applications that deal with structured documents. The architecture of the Manifold framework is lightweight, scalable, and extensible.

Example Applications

Using the Manifold framework we developed three complex applications: a 2D constrained graphics editor (Palmscape - Figure 3) running on Palm V equipped with a wireless modem, a 2D graphics editor (Flatscape - Figure 4) and a 3D virtual world (cWorld - Figure 5). Flatscape can run on a laptop or other lower-end personal computers, while cWorld can run on a desktop workstation, using optionally a Window VR (Figure 1) for display and interaction. The heterogeneity is simulated by both using different display applications and restricting the window size of the Flatscape environment to the screen size of a typical Windows CE handheld computer (320×240 pixels).

Figure 3 - A room floor plan as seen using Palmscape.

As the application on the PDA has to be very thin, we have chosen an approach where only the presentation tier of the application runs on the PDA, while a communication proxy, running on a workstation, implements the domain and collaboration tier. Manifold is relatively lightweight, but it can still be a problem to run on a Palm Pilot, mainly because of memory limitations. Our solution to this problem is to run only the presentation tier of the client on the PDA, while communication, collaboration, and domain tiers are running on a Personal Computer (PC). With the separation of domain, collaboration and presentation, its possible to develop very thin clients for the PDA (class-files occupy less than 20 KB).

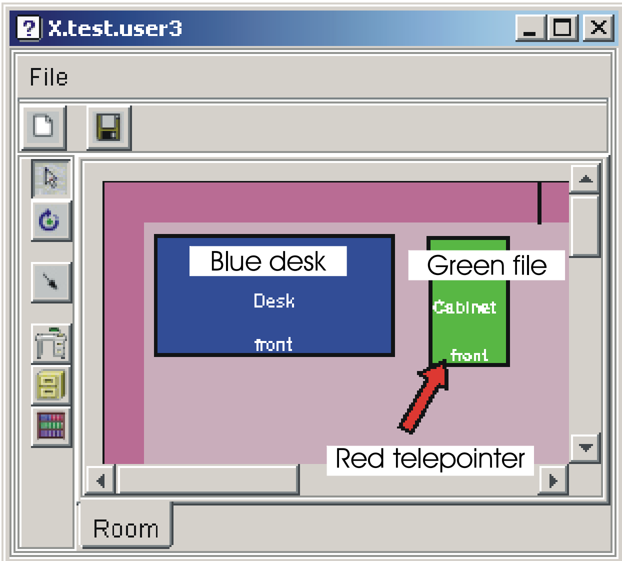

Figure 4 - A room floor plan as seen using Flatscape.

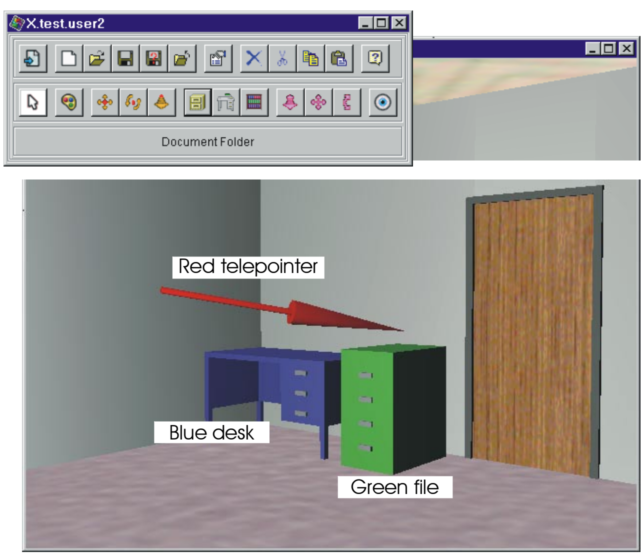

Figure 5 - A sample CVE built using cWorld, showing the same room as Figure 4.

Evaluation

Palmscape, Flatscape and cWorld demonstrate that our framework supports heterogeneous collaborative environments. Although we show that we can construct these environments, we do not know how effective they will be for collaboration. It is readily assumed that a common view is necessary for effective communication, but others have found that non-WYSIWIS systems do not impair collaboration and may even facilitate it [1]. Obviously this depends on how different the shared views are. Our 2D vs. 3D displays represent a common future difference expected in office-field communication. We have used Flatscape and cWorld to evaluate the effect of this difference in a short study. The study required each team to perform four office furniture arrangement tasks. In each task, a total of 9 pieces of furniture (bookcases, file cabinets and desks) had to be placed at specified positions in the room. Each team member was given a sheet of paper with a top-down view of the placement of 3 of the objects. They could not, however, place these objects themselves and needed to direct other team members in their placement. Color indicated the mover of the objects and the owner of each telepointer. For each of the tasks, we assigned a display configuration for each of the team members so that they were either in a 2D (Flatscape) or 3D (cWorld) environment.

Each view of the environment gives users certain advantages for creating and placing objects on their display. However, their different views of the world are likely to lead to misunderstandings in the communication that transpires. We expect these misunderstandings to be recognized (because of the observed changes on each user's display) and immediately repaired through the verbal channel. These repairs are likely to be quick and considered a natural part of the collaboration. However, they are also likely to increase the amount of time required to perform the collaboration task. In the study we examined whether this was true. We expected the homogenous collaboration to show better performance times and fewer repairs than the heterogeneous collaboration. However, we were not able to find these differences. We believe that this is because of two additional factors; (1) the difficulty of placing objects in the 3D environment and (2) the advantage of viewing objects in the 3D environment. The object placement times in the 2D–2D collaboration were, on average, 31 seconds faster than the 3D–3D collaboration. The full discussion of the study and numerical results are presented in [10].

Conclusions

The need for heterogeneous sharing in synchronous collaboration grows ever stronger with the proliferation of diverse computing environments, particularly wearable and handheld computers. This work presents a data-centric design for synchronous collaboration of users with heterogeneous computing platforms, by which we have implemented applications for target devices ranging from 3D workstations to PDA's. It is a significant departure from the traditional model of sharing, which is application- or procedure-centric. The communication architecture is purely distributed, i.e. there is no central server. Our approach allows clients with different capabilities to share different subsets of data in order to conserve communication bandwidth. We also illustrate, via an extreme example of size and dimensionality differences (Figures 3-4), that heterogeneous collaboration does not appreciably affect task performance and that users perceive the task performance to be equivalent to homogenous environment collaboration.

References

[1] M. Stefik, D. G. Bobrow, S. Lanning, and D. G. Tatar, WYSIWIS revised: Early experiences with multiuser interfaces, ACM Transactions on Information Systems, 5(2), 1987, 147-167.

[2] D. Kahneman, *Attention and Effort*, (New Jersey: Prentice-Hall, 1973).

[3] D. Engelbart, Authorship provisions in AUGMENT. Proc. IEEE Compcon Conference, 1984.

[4] A. Karsenty, C. Tronche, and M. Beaudouin-Lafon, GroupDesign: Shared editing in a heterogeneous environment, USENIX Computing Systems, 6(2), 1993, 167-195.

[5] J. F. Patterson, R. D. Hill, S. L. Rohall, and W. S. Meeks, Rendezvous: An architecture for synchronous multi-user applications, Proc. ACM Conference on Computer-Supported Cooperative Work (CSCW'90), 1990, 317-328.

[6] M. Roseman, and S. Greenberg, Building real-time groupware with GroupKit, a groupware toolkit, ACM Transactions on Computer-Human Interaction, 3(1), 1996, 66-106.

[7] B. A. Myers, H. Stiel, and R. Gargiulo, Collaboration using multiple PDAs connected to a PC, Proc. ACM Conf. on Computer-Supported Cooperative Work (CSCW'98), 1998, 285-294.

[8] P. Dewan and A. Sharma, An experiment in interoperating heterogeneous collaborative systems, *Proc. 6th European Conf. on Computer Supported Cooperative* Work (ECSCW'99), Kluwer Acad. Publ., 1999, 371-390.

[9] D. R. Olsen, S. E. Hudson, M. Phelps, J. Heiner, and T. Verratti, Ubiquitous collaboration via surface representations, Proc. ACM Conference on Computer-Supported Cooperative Work (CSCW'98), 1998, 129-138.

[10] I. Marsic, A. M. Krebs, H. Trefftz, B. Dorohonceanu, and M. Tremaine, Designing and Testing PC to Palm Collaboration, submitted, 2000.

[11] D. G. Tatar, G. Foster, and D. G. Bobrow, Design for conversation: Lessons from Cognoter, International Journal of Man-Machine Studies, 34(2), 1991, 185-209.

[12] J. Hindmarsh, M. Fraser, C. Heath, S. Benford, and C. Greenhalagh, Fragmented interaction: Establishing mutual orientation in virtual environments, Proc. ACM Conference on Computer-Supported Cooperative Work (CSCW'98), 1998, 217-226.

[13] Steed, M. Slater, A. Sadagic, A. Bullock, and J. Tromp, Leadership and collaboration in shared virtual environments, Proc. of the IEEE Virtual Reality Conference, 1999, 112-115.