Research in Distributed Mobile and Collaborative Systems: An Overview

Department of Computer Science | University of North Carolina at Chapel Hill | March 2005

Faculty Interests

Prasun Dewan, professor (Application Sharing and Mobile Computing)
Kevin Jeffay, S. S. Jones Distinguished Term professor (Application Sharing)
Ketan Mayer-Patel, assistant professor (Collaborative Mobile Systems)
Maria Papadopouli, assistant professor (Mobile Computing)
Don Smith, research professor (Hypermedia/Web)
John Smith, professor (Hypermedia/Web)
David Stotts, associate professor (Formal Models, Hypermedia, Mobile Computing)

Graduate Student Interests

Sasa Junuzovic (Application Sharing Architectures)
Dorian Miller (Assistive Collaboration Technologies)
Karl Gyllstrom (Paired Synchronous Collaboration)
Keith Lee (Paired Synchronous Collaboration)
Olufisayo Omojokun (Mobile Computing)
Jason Smith (Design Pattern Algebra)

Research in the Department of Computer Science at UNC–Chapel Hill, in the areas of mobile and collaborative systems, covers a wide range of topics from formal verifiable models of collaboration to dynamic migration and replication. These two areas are mostly contained within the larger area of distributed systems. The majority of our research lies in the overlap between these two areas. As shown in Figure 1 (above right), other related areas are multimedia and real-time computing.

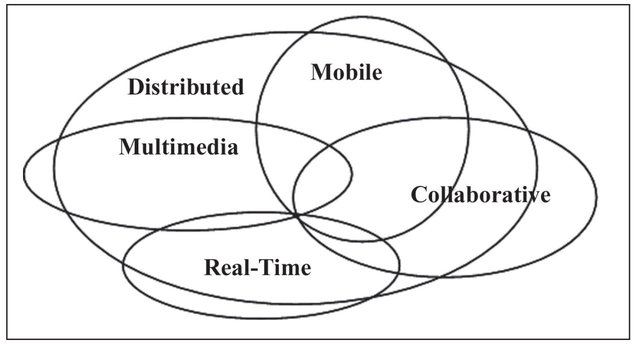

Figure 1. Areas of distributed systems.

Mobile Applications Research

Mobile applications are software systems that run on mobile computers, such as cell phones, palm-tops, and other hand-held devices. If such devices cannot be connected via a network to other computers, their uses are limited. Thus far, they have been used primarily to connect to regular desktop computers so that they can serve as small, portable caches of the data on these computers. Another emerging distributed use of these devices is to connect them to electronic appliances—such as TVs, washing machines, thermostats, lights and projectors—that are controlled by computing devices. Using a mobile computer, rather than directly using the controls provided by the appliance, provides several advantages, including: (a) remote manipulation of the device, (b) storage of preferences and previous interactions on the mobile device, which can be applied to different appliances of similar functionality, and (c) user-interfaces that manipulate multiple, related devices such as the thermostats in a building.

We are looking at some of the issues that arise in these two distributed uses of mobile systems: serving as a cache for desktop data, and controlling appliances. In particular, we are looking into techniques that allow the binding between the mobile computer and remote computer (desktop machine or appliance) to be made as late as possible. At a minimum, we want the mobile computer to be able to connect to any remote computer on the network that it knows how to communicate with, rather than require it to have established a partnership previously with that computer. There are many solutions to this problem, such as Sun's Jini system and Microsoft's plug and play architecture, and we currently are working to evaluate them. Ketan Mayer-Patel, assistant professor, is taking late binding a step further by allowing a mobile computer to communicate with other computers using a device-independent event model, thereby not requiring it to know about specific types of computers. For instance, a palmtop computer should be able simply to announce that a new appointment has been added to the user's calendar, without worrying about the specifics of how it should be communicated to the handheld device or desktop computer interested in the event. Prasun Dewan, professor, and Olufisayo Omojokun, graduate student, are trying to advance this idea for the special case of a mobile device controlling an appliance. They are looking into techniques that allow the mobile computer to learn (using a technique called computation reflection) about the events to which the appliance responds and, based on these events, to generate a user interface for controlling the device. Thus, the mobile computer would not be required to be bound to any kind of event (either device-dependent or independent), thereby allowing it to control arbitrary appliances.

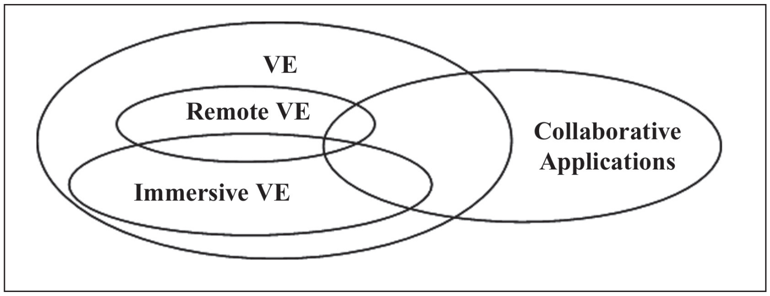

Figure 2. Overlapping research areas of collaborative applications and virtual environments.

Collaborative Applications Research

Collaborative applications are multiuser applications that couple their users; that is, they allow the input of one user to influence the output seen by another user. The obvious use of these applications is in supporting distributed collaboration; in giving users in different locations the sense of being there; of being present in the same location. Since an illusion can approximate but never equal reality, collaborative applications that try to support this sense of being there will always be second best to actually being there. It can be argued that for remote collaboration to be really successful, it should go beyond being there by supporting methods of collaboration (such as asynchronous collaboration) that cannot be supported in direct face-to-face interaction. These two goals—trying to approximate and to go beyond being there—define our research in this area.

Being there in a world other than the one in which the user actually is present, implies being in a virtual environment. In fact, this area of collaborative applications overlaps with the field of virtual reality, as shown in Figure 2. We use the term "virtual environment" loosely here to imply some approximation of a simulated real world. Immersive environments form an important subset in which the approximation is more precise. Environments that approximate a remote world form another important subset. As Figure 2 illustrates, all three sets can overlap with collaborative environments. Several current projects at UNC–Chapel Hill are exploring these intersections. The Distributed nanoManipulator Project (see separate handout) creates a collaborative environment that is immersive but not remote; it creates a view of some material at atomic scale that multiple users can share and manipulate. Researchers in the Office of the Future Project (see separate handout) are working on offering a sense of presence that will be as good as physically being present, by creating a remote collaborative virtual environment. Several Web-based projects are developing collaborative environments that try to go beyond the mere sense of being there by supporting asynchronous web-based collaboration.

General System-Level Collaboration

What distinguishes the work of this group of researchers from other researchers in the Department of Computer Science who are also exploring collaboration is that we are studying general system-level support for collaboration, rather than applications to support specific forms of collaboration. The system issues we are addressing concern performance/resource management (e.g., dynamic migration and replication of a server to improve performance), automatic support of some collaboration functionality (e.g., replica synchronization, access control, concurrency control, authoring on the Web), extensibility/composibility (e.g., development of collaborative beans), and formal models of collaboration protocols that can be theoretically verified. We are considering both synchronous, asynchronous, and mobile collaboration.

For More Information

Dr. Prasun Dewan
Department of Computer Science
University of North Carolina at Chapel Hill
CB#3175, Sitterson Hall
Chapel Hill, NC 27599-3175
Phone: (919) 962-1823
Fax: (919) 962-1799
E-mail: dewan@cs.unc.edu
Web: www.cs.unc.edu/Research/Colab/