Abstract

As computers proliferate, becoming smaller, more mobile, more powerful, and more diverse, how will the ways in which we interact with them change? In this article we describe research in developing "hybrid user interfaces" that tie together the diverse displays and interaction devices that a user encounters in a mobile, shared environment. Controlling such a dynamically changing, heterogeneous mix of computers is a problem that we refer to as "environment management." We sketch some ways in which publicizing semantic information about computational objects and tasks can make it possible to automate environment management operations, and we describe research testbeds we are developing within which to explore these ideas.

Environment Management for Hybrid User Interfaces

Steven K. Feiner, Columbia University

Current computers work only or mostly on a desk or a lap, typically offering a single CRT or flat-panel display to a single user. Mobile systems, as represented by the early generations of PDAs, trade off power for portability, which is all too evident in the size and functionality of their present user interfaces. During this decade, however, advances in hardware and wireless networking will make it possible to radically change the shape of our computational environment to provide user-interface support for mobile, interacting users as they move about among large numbers of wall-sized, desk-top, hand-held, and head-worn displays.

For this to happen, it will be necessary to transcend today's one-user/one-display metaphor. We believe that future computing environments will include *hybrid user interfaces* [1], in which multiple heterogeneous 2D and 3D displays and interaction devices are used in synergistic combination to benefit from the advantages of each. For example, multiple users wearing lightweight, see-through, head-worn displays could participate in a shared virtual environment within which selected public information was presented on large shared displays, such as wall-mounted and virtual workbench displays. Each user's eyewear could privately overlay their personal annotations over the shared displays, while additional personal material might appear on a user's hand-held display. But, how can we control such a rich environment? We refer to the problem of managing large numbers of objects, on possibly large numbers of displays, in the virtual and real surroundings, as *environment management* [2], as an analogy to window management. Environment management will be an especially challenging task if it is to address the needs of mobile, collaborating users, whose proximity to other users and to displays and interaction devices may change rapidly as the users move about.

While managing relatively small numbers of windows on a conventional small desktop can be done entirely under direct user control, it quickly becomes tedious as the number of windows increases. For this reason, a number of researchers have experimented with automated strategies, such as the constraint-based approach to window management taken in RTL/CRTL [3] and SCWM [4]. When dealing with large numbers of user-interface objects (both windows and arbitrary other objects) that can be placed literally anywhere in the surrounding 3D world, direct control will often be impractical. Existing applications typically publicize only simple syntactic information about themselves for use by window managers that control the display of their user interfaces, such as window size and position. We believe that it would be useful to explore a different model that exploits semantic information about the content and purpose of user-interface objects. This additional information would be provided both by individual applications and by the framework within which the applications reside. It would make possible *environment managers* that use this information to determine how to lay out and actively maintain the user-interface objects that make up the collaborative information environment.

For example, two virtual objects may be related because they are different views of the same physical object. If this information relating the two objects were available to an environment manager, it could be used to position the objects near each other. Suppose further that the user is examining these objects on her see-through head-worn display and that she wants to talk with a nearby user, whose image the objects obscure, or to a distant user whose visual representation appears before her. If an environment manager knows that the objects are directly related to a task the first user is performing and wants to discuss, the objects could be automatically pushed aside to remain visible, allowing an unobstructed physical or virtual view of the other user. Alternatively, if the objects are not currently in use, they could be automatically dimmed, obscured, scaled down, stacked and pushed away, etc., much as the user might currently iconify them by hand.

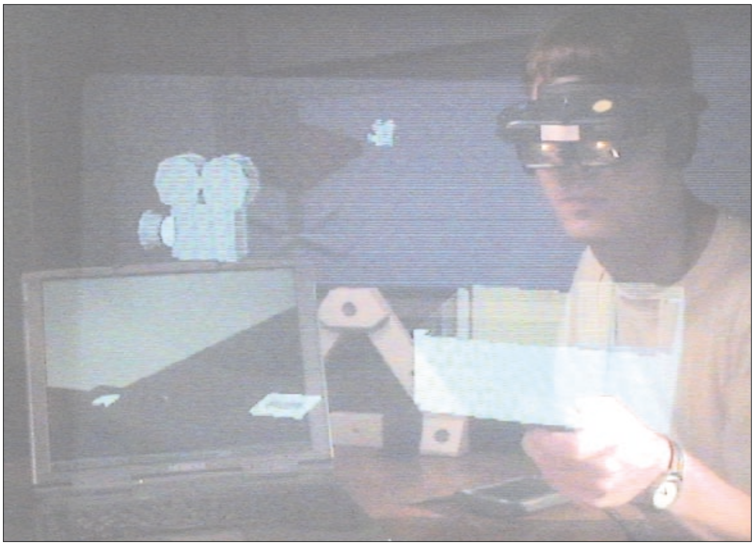
There are many kinds of semantic information that could be useful for effective environment management. For example, low-level semantic information may be provided by an individual application about its objects (e.g., the fact that a window contains an important error message). Higher-level information could reflect the context of a group of collaborating users and the many pieces of software they may be using to accomplish an assortment of tasks. In addition, some information may be domain-independent (e.g., the fact that a 3D model is being displayed), while other information may be domain-dependent (e.g., the fact that the stress analysis of a ship's hull is being displayed).

1070-9916/00/$10.00 © 2000 IEEE

Environment managers would be for distributed 3D hybrid user interfaces what window managers are for traditional 2D user interfaces. Window managers normally do exactly what the user tells them, though requiring that they be controlled at essentially the pixel level, within the typically static 2D space of the screen. In contrast, the environment managers that we propose are better suited to the dynamic nature of a hybrid user interface, interacting with the user's actions and the ever-changing environment and assisting the user in handling the management details. Therefore, we refer to such systems as *dynamic environment managers*.

The first step in building a dynamic environment manager is to build an object-management infrastructure that supports more powerful operations on top-level (i.e., management-level) objects. This infrastructure would:

- Allow applications to add explicit semantic information to objects that express their inherent properties and relationships to other objects, and infer basic semantic information when it is not provided. For example, transient windows might be assumed to be dialog boxes or alerts.
- Support the development of a set of high-level goals that the environment manager would need to fulfill. Examples include maintaining an object's visibility or legibility, keeping a set of objects from taking up too much display space, and keeping related objects near each other.
- Provide facilities for characterizing a set of user tasks, expressed in part in terms of the high-level environment-management goals. For example, a collaborative interaction may require mutual visibility between active participants, visibility and legibility to all participants of shared objects currently being referenced, and invisibility of private objects to all participants without access rights.
- Allow constraints to be specified on object properties. For example, constraints could support keeping a portion of an object visible or maintaining relative positions and sizes of objects. Constraints such as these are characterized by being solvable at interactive speeds, so that they can be maintained as the objects are moved by the user or by other parts of the system.
- Add a knowledge-based component that uses the high-level goal and task structure to organize the objects across available displays. Organization could take place on two levels. First, the system could do gross object organization by selecting reasonable initial positions and sizes for objects and rearranging displays when the current organization is sufficiently bad. Second, it could automatically add, remove, and modify constraints between objects to do fine-grained object organization.



■ **Figure 1.** *EMMIE allows multiple users to manipulate objects across a variety of displays.*

A general-purpose object management infrastructure would support: 3D objects, not just 2D windows; multiple, heterogeneous displays; and moving objects among displays.

Examples

We are developing several testbeds within which to explore these ideas, initially in environments that users construct entirely by hand. In our work on EMMIE (Environment Management for Multi-user Information Environments), we are experimenting with providing an integrated user interface, in the spirit of a conventional single-user, single-display GUI window manager, but addressing the needs of multiple users who manipulate information across many different displays [5, 6].

Figure 1 shows our EMMIE prototype, imaged through a video camera that is wearing a tracked see-through head-worn display. The user is shown manipulating a virtual model of our lab in one hand. Other virtual objects in the image include simple iconic 3D representations of application objects (e.g., a "movie projector" that can play videos) and data objects (e.g., a "slide" that represents a still image).

(a) (b) (c)

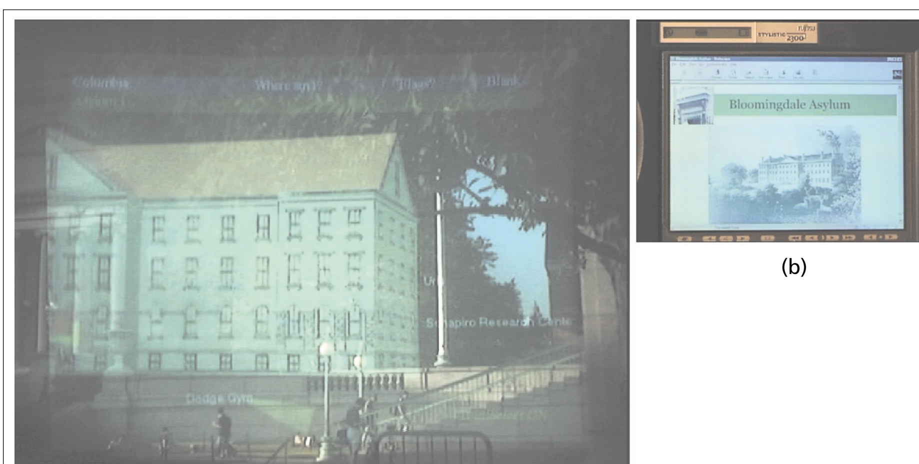
■ **Figure 2.** *Copying an object across displays in EMMIE. (a) The user selects an object displayed on laptop and (b) drags a visible representation of it through the ether, to (c) copy it to the wall-mounted display.*

■ **Figure 3.** *The "Touring Machine" mobile augmented reality system presents information using a tracked see-through head-worn display and a stylus-based handheld display. A handheld map of the campus is shown at left, an annotated real-world view through the see-through head-worn display at center, and the current backpack testbed at right.*

Dragging data objects to application objects allows them to be processed. Physical objects are also tracked; for example, the notebook computer has a 6DOF ultrasonic tracker on the back of its display. This makes it possible for objects from the notebook's GUI to interact with those of the surrounding environment. The 3D virtual environment, visible on the see-through displays, serves as an all-encompassing "ether" within which virtual and physical objects coexist. For example, as shown in Fig. 2, a user can grab an object displayed on a laptop and copy it to a wall-mounted display, viewing it as it moves through the ether. There is an interesting contrast between EMMIE and Rekimoto and Saitoh's augmented surfaces [7], in which objects dragged between displays are visualized for all users on a shared projected surface. Because EMMIE's ether is viewed on personal head-worn displays, material in it can be customized for an individual user; for example, to support privacy by selectively displaying private objects only to users who are allowed to access them [5]. EMMIE's infrastructure relies on a shared object directory that defines how objects appear and behave, and which is replicated across, and writable by, all participating processes. While this approach does not address the security requirements of a commercial implementation, it has made possible a flexible testbed for developing multi-user, multi-display interaction techniques.

We are investigating a different, but complementary, set of environment-management issues in our "Touring Machine," an outdoor, backpack-based, mobile augmented reality system, shown in Fig. 3, that assists a user exploring Columbia's campus [8]. As a user moves about, she is tracked through a combination of centimeter-level, real-time-kinematic GPS position tracking and inertial/magnetometer orientation tracking. The system's hybrid user interface includes a head-tracked, see-through, 3D head-worn display, and an untracked, opaque, 2D, handheld display with stylus and trackpad. Spatial information, extracted from campus databases, is overlaid on and registered with the mobile user's view of the real world as she walks around outside.

We have been using this system to create *situated documentaries* [9] that inform users about their surroundings. A situated documentary embeds a hypermedia presentation within the actual physical environment that it describes. The iconic flags shown at the center of Fig. 3 represent synchronized multimedia chunks of narrated imagery, video, 3D graphics, sound, and applets that can be selected for presentation by the user. For example, one of our documentaries, shown in Fig. 4, recounts the history of the Bloomingdale Asylum, the former occupant of Columbia's present campus, through 3D



■ **Figure 4.** *Situated documentary about Columbia's history uses (a) 3D models of demolished buildings overlaid* in situ *on head-worn display, and (b) text and graphics on hand-held display.*

models of its buildings overlaid on the head-worn display, accompanied by text and graphics displayed on the hand-held computer. Information is allocated between the system's two displays by the authors of our documentaries, taking into account the differences between the displays' capabilities. One of the goals of this research is to develop a better understanding of how multiple displays can be used together effectively, in preparation for the development of dynamic environment-management tools that can automate some of these design decisions.

Acknowledgments

These ideas have benefited from conversations with, and the development efforts of, many members of Columbia's Computer Graphics and User Interfaces Lab, including Blair MacIntyre, Tobias Höllerer, Andreas Butz, Elias Gagas, Drexel Hallaway, Sinem Güven, and Tachio Terauchi. This research was supported in part by ONR Grants N00014-97-1-0838, N00014-99-1-0249, and N00014-99-1-0394, NSF Grant EIA-97-2984, and gifts from IBM, Intel, Microsoft, and Mitsubishi.

References

[1] S. Feiner and A. Shamash, "Hybrid User Interfaces: Breeding Virtually Bigger Interfaces for Physically Smaller Computers," *Proc. ACM UIST '91 (Symp.User Interface Software and Technology)*, Hilton Head, SC, Nov. 11–13, 1991, pp. 9–17.

[2] B. MacIntyre and S. Feiner, "Future Multimedia User Interfaces," *Multimedia Systems*, vol. 4, no. 5, 1996, pp. 250–68.

[3] E. Cohen, E. Smith, and L. Iverson, "Constraint-based Tiled Windows," *IEEE CG&A*, vol. 6, no. 5, May 1986, pp. 35–45.

[4] G. Badros, J. Nichols, and A. Borning, "SCWM: An Extensible Constraint-Enabled Window Manager," *Proc. AAAI Spring Symp. on Smart Graphics*, Mar. 20–22, 2000.

[5] A. Butz, C. Beshers, and S. Feiner, "Of Vampire Mirrors and Privacy Lamps: Privacy Management in Multi-user Augmented Environments," *Proc. ACM UIST '98*, San Francisco, CA, Nov. 2–4, 1998, pp. 171–72.

[6] A. Butz et al., "Enveloping Users and Computers in a Collaborative 3D Augmented Reality," *Proc. IWAR '99 (IEEE and ACM Int. Wksp. Augmented Reality)*, San Francisco, CA, Oct. 20–21, 1999, pp. 35–44.

[7] J. Rekimoto and M. Saitoh, "Augmented Surfaces: A Spatially Continuous Work Space for Hybrid Computing Environments," *Proc. ACM CHI '99*, Pittsburgh, PA, May 15-20, 1999, pp. 378-85.

[8] S. Feiner *et al.*, "A Touring Machine: Prototyping 3D Mobile Augmented Reality Systems for Exploring the Urban Environment," *Personal Technologies*, vol. 1, no. 4, Dec. 1997, pp. 208–17.

[9] T. Höllerer, S. Feiner, and J. Pavlik, "Situated Documentaries: Embedding Multimedia Presentations in the Real World," *Proc. IEEE ISWC '99 (Int'l. Symp. Wearable Computers)*, San Francisco, CA, Oct. 18–19, 1999, pp. 79–86.

Biography

STEVEN K. FEINER (feiner@cs.columbia.edu) is a professor of computer science at Columbia University, where he directs the Computer Graphics and User Interfaces Laboratory (http://www.cs.columbia.edu/graphics). He received an A.B. in Music and a Ph.D. in computer science from Brown University in 1973 and 1987, respectively. His research interests include virtual environments and augmented reality, knowledge-based design of graphics and multimedia, information visualization, wearable computing, and hypermedia. He is a co-author of *Computer Graphics: Principles and Practice* (Addison-Wesley, 1990) and of *Introduction to Computer Graphics* (Addison-Wesley, 1993). He is an associate editor of *ACM Transactions on Graphics*, and serves on the executive boards of the IEEE Computer Society Technical Committee on Visualization and Graphics and the IEEE Computer Society Task Force on Human-Centered Information Systems. He was program co-chair of IEEE Virtual Reality 2000, and is conference chair of IEEE Information Visualization 2001 and program co-chair of the International Symposium on Mixed Reality 2001. In 1991 he received an Office of Naval Research Young Investigator Award.