

EDUCATION

1997 - 2000
 Bachelor of Science in Architecture
 University of Toronto

EXPERIENCE

2000 - 2001
 Intern Architect
 Skidmore, OWing Merrill & Partners

2001 - 2002
 Intern Architect
 Skidmore, OWing Merrill & Partners

2002 - 2003
 Intern Architect
 Skidmore, OWing Merrill & Partners

2003 - 2004
 Intern Architect
 Skidmore, OWing Merrill & Partners

2004 - 2005
 Intern Architect
 Skidmore, OWing Merrill & Partners

2005 - 2006
 Intern Architect
 Skidmore, OWing Merrill & Partners

2006 - 2007
 Intern Architect
 Skidmore, OWing Merrill & Partners

2007 - 2008
 Intern Architect
 Skidmore, OWing Merrill & Partners

2008 - 2009
 Intern Architect
 Skidmore, OWing Merrill & Partners

2009 - 2010
 Intern Architect
 Skidmore, OWing Merrill & Partners

2010 - 2011
 Intern Architect
 Skidmore, OWing Merrill & Partners

2011 - 2012
 Intern Architect
 Skidmore, OWing Merrill & Partners

2012 - 2013
 Intern Architect
 Skidmore, OWing Merrill & Partners

2013 - 2014
 Intern Architect
 Skidmore, OWing Merrill & Partners

2014 - 2015
 Intern Architect
 Skidmore, OWing Merrill & Partners

2015 - 2016
 Intern Architect
 Skidmore, OWing Merrill & Partners

2016 - 2017
 Intern Architect
 Skidmore, OWing Merrill & Partners

2017 - 2018
 Intern Architect
 Skidmore, OWing Merrill & Partners

2018 - 2019
 Intern Architect
 Skidmore, OWing Merrill & Partners

2019 - 2020
 Intern Architect
 Skidmore, OWing Merrill & Partners

2020 - 2021
 Intern Architect
 Skidmore, OWing Merrill & Partners

2021 - 2022
 Intern Architect
 Skidmore, OWing Merrill & Partners

2022 - 2023
 Intern Architect
 Skidmore, OWing Merrill & Partners

2023 - 2024
 Intern Architect
 Skidmore, OWing Merrill & Partners

2024 - 2025
 Intern Architect
 Skidmore, OWing Merrill & Partners

EDUCATION

1997 - 2000
 Bachelor of Science in Architecture
 University of Toronto

EXPERIENCE

2000 - 2001
 Intern Architect
 Skidmore, OWing Merrill & Partners

2001 - 2002
 Intern Architect
 Skidmore, OWing Merrill & Partners

2002 - 2003
 Intern Architect
 Skidmore, OWing Merrill & Partners

2003 - 2004
 Intern Architect
 Skidmore, OWing Merrill & Partners

2004 - 2005
 Intern Architect
 Skidmore, OWing Merrill & Partners

2005 - 2006
 Intern Architect
 Skidmore, OWing Merrill & Partners

2006 - 2007
 Intern Architect
 Skidmore, OWing Merrill & Partners

2007 - 2008
 Intern Architect
 Skidmore, OWing Merrill & Partners

2008 - 2009
 Intern Architect
 Skidmore, OWing Merrill & Partners

2009 - 2010
 Intern Architect
 Skidmore, OWing Merrill & Partners

2010 - 2011
 Intern Architect
 Skidmore, OWing Merrill & Partners

2011 - 2012
 Intern Architect
 Skidmore, OWing Merrill & Partners

2012 - 2013
 Intern Architect
 Skidmore, OWing Merrill & Partners

2013 - 2014
 Intern Architect
 Skidmore, OWing Merrill & Partners

2014 - 2015
 Intern Architect
 Skidmore, OWing Merrill & Partners

2015 - 2016
 Intern Architect
 Skidmore, OWing Merrill & Partners

2016 - 2017
 Intern Architect
 Skidmore, OWing Merrill & Partners

2017 - 2018
 Intern Architect
 Skidmore, OWing Merrill & Partners

2018 - 2019
 Intern Architect
 Skidmore, OWing Merrill & Partners

2019 - 2020
 Intern Architect
 Skidmore, OWing Merrill & Partners

2020 - 2021
 Intern Architect
 Skidmore, OWing Merrill & Partners

2021 - 2022
 Intern Architect
 Skidmore, OWing Merrill & Partners

2022 - 2023
 Intern Architect
 Skidmore, OWing Merrill & Partners

2023 - 2024
 Intern Architect
 Skidmore, OWing Merrill & Partners

2024 - 2025
 Intern Architect
 Skidmore, OWing Merrill & Partners

GEORGE W. FITZMAURICE, ALIAS
 AZAM KHAN, ALIAS
 WILLIAM BUXTON, BUXTON DESIGN
 GORDON KURTENBACH, ALIAS
 RAVIN BALAKRISHNAN, UNIVERSITY OF TORONTO



SENTIENT DATA ACCESS

via a Diverse Society of Devices

Today's ubiquitous computing environment cannot benefit from the traditional understanding of a hierarchical file system.

It has been more than ten years since such "information appliances" as ATMs and grocery store UPC checkout counters were introduced. For the office environment, Mark Weiser began to articulate the notion of UbiComp (ubiquitous computing) and identified some of the salient features of the trends in 1991.^{1,2} Embedded computation is also becoming widespread.

Microprocessors, for example, are finding themselves embedded into seemingly conventional pens that remember what they have written.³ Anti-lock brake systems in cars are controlled by fuzzy logic. And as a result of wireless computing, miniaturization, and new economies of scale, such technologies as PDAs (personal digital assistants), IM (instant messaging), and mobile access to the Internet are almost taken for granted.

But while many of the components of UbiComp that were described and anticipated by Weiser are now commonplace, major aspects of the vision are still developing. A common language for these devices has not been

standardized, nor have current database solutions sufficiently captured the complexities involved in correctly expressing multifaceted data. In particular, XML is only now emerging as a viable backbone for communication within a diverse society of devices. CMSs that are now commercially available would be capable of appropriately expressing the data, but often still need to be custom-built for a given application domain. In this discussion, we focus on modeling the human aspect of interactions in the type of rich computing environment we envisage becoming commonplace.

FRAMING THE PROBLEM

The widespread growth of computational and communications technologies is obvious. But from our perspective, it is not the ubiquitousness of the technology per se that is of primary importance, but rather how its existence fosters changes in *who* employs "computation" (in the broadest sense), *where* they do so, *how* they interact, and

SENTIENT DATA ACCESS

via a Diverse Society of Devices

what it is used for. Technology is certainly important, but our perspective is shaped by the notion that its importance lies in its potential to serve as a motor-sensory, cognitive, and social prosthesis—not as an end in itself.

Ubiquitous computing is in some ways an everyday reality. However, *cooperative* ubiquitous computing is still in its infancy. New forms of interaction must be developed for this environment—interaction between two or more parties: people and people (both technologically mediated and not), people and machines, and machines and machines. Implicit in this formulation is the importance of location. Previously, transactions took place where the computer was anchored. The location of the computer was not a design issue. Now distance (both physical and social) and location are key considerations in understanding and designing systems. The underlying concept is perhaps best articulated in a famous quote from the architect Louis I. Khan: “Thoughts exchanged by one another are not the same in one room as in another.”⁴ This includes “thoughts” exchanged between people and/or machines, and implies that behavior is sensitive to location, and as a consequence of mobility, must adapt to changes in physical and social location.

With respect to location-based design, the particular input and output technologies being considered closely

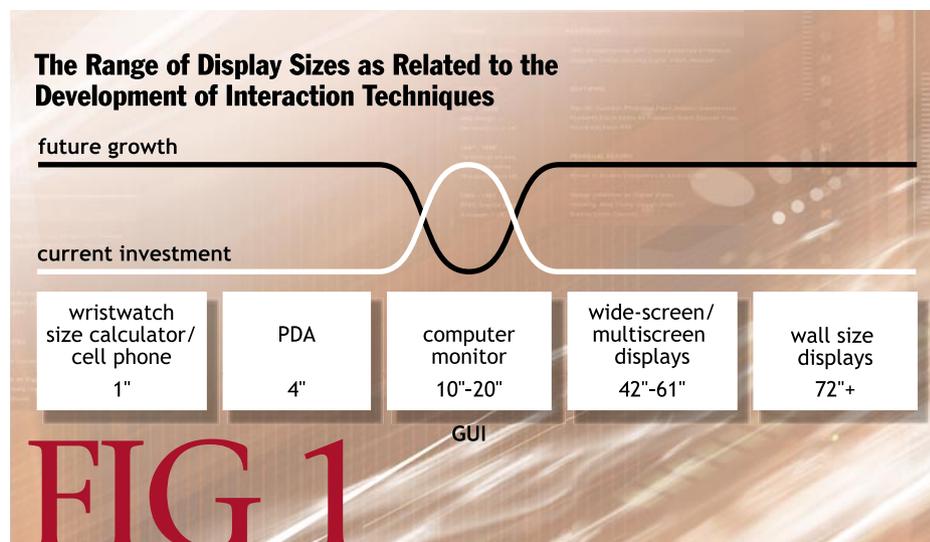
interplay with the choices made for the data formats and the ways to present the data. A wide variety of input technologies was developed during the dawn of UbiComp, and we now see a plethora of output devices also being introduced. Small displays are appearing everywhere, on appliances from watches, to pens, and telephones. Equally interesting is how the increasing penetration of plasma panels has led to large-format displays being used as general-purpose signage, such as electronic movie posters at cinemas. It is clear that this trend will only accelerate, given the progress and promise of organic light-emitting diode (OLED) technology, which is already finding its way into commercial products.⁵

While UbiComp is increasingly characterized by a growing deployment of small (mainly mobile) and large (mainly embedded) displays, our current store and our investment in interaction techniques are still dominated by the demands of the GUI running on a traditional desktop computer (see figure 1). The classes of devices illustrated are shown along a linear one-dimensional scale in a way that implies that they reflect a series of distinct, independent devices—which is largely consistent with current practice. However, at the PARC in the late 1980s, when we were developing the tabs, pads, and “Liveboards” discussed by Weiser, we were primarily exploring the relationships and interactions among these devices as they related to artifacts, and to people, in the physical world. It is these relationships we intend to explore in detail.

As computing devices expand from the status-quo keyboard and desktop to a variety of form factors and scales, we can imagine workplaces configured to have a society of devices, each designed for a very specific task.

As a whole, the collection of devices may act much like a workshop in the physical world, where the data moves among the specialized digital stations. For our society of devices to operate seamlessly, a mechanism will be required to (a) transport data between devices and (b) have it appear at each workstation, or tool, in the appropriate representation.

This vision has two aspects: the system and network architecture to



support transport and access, or *system model*; and the user's conceptualization of these activities, or *user model*. An example of this system/user model distinction is the standard desktop system in which file transfers have a "drag-and-drop" user model, while the underlying system model is a "file move" from one directory to another.

FIG 2

Simple Usage Scenario

(a) Scan a barcode on artwork using a PDA, (b) carry PDA to viewing terminal and (c) transfer item to Powerwall terminal for viewing 3-D model.



USER MODEL

Our user model draws inspiration from, and hybridizes, two related fields of research: wearable/mobile computing,⁶ and embedded ubiquitous computing environments.⁷ The idea is to use wearable/mobile computers to carry referential data to embedded computing environments at specific locations. This presents three fundamental questions for users:

What do you carry?

We depart from the graspable/tangible approach^{8,9} in which an individual physical artifact exists for every piece of digital data you wish to carry. Because this approach does not scale well, we take an ecological approach and consider what we reasonably expect a person to carry with them (e.g., a watch, PDA, or phone). While a person would only need to carry a single physical artifact, it would be capable of holding multiple data references.

What is in place at the location you are going to?

We assume there are task-specific devices at special locations. Taking the household as an example, locations such as the kitchen afford and imply a very different set of tasks from other locations such as the family room.

What is the relationship between the things you carry and the equipment at a given location?

We assume that all stationary devices are connected via a network, as are the mobile devices, at least when in proximity to the stationary equipment. Thus, mobile devices need only carry references to data because the network makes the data pervasive. We also assume a mobile device may act as part of a specific user interface to the computational elements at a particular location, as well as

a carrier of references to the data to be operated upon.

We illustrate this approach with a simple example from our experimental environment: an automotive design studio. In this example, a designer sees a physical picture of a car posted on a studio art board and would like to see the virtual 3-D model of the same car on the studio's wall-sized display device (called a *Powerwall*). The designer uses a PDA equipped with a bar-code scanner to record a bar code printed on the corner of the picture of the car, thereby capturing the reference to the data associated with the sketches. By carrying the PDA to the Powerwall, a 3-D geometric model of the same car is displayed on the screen when the user presses a *Send* button (see figure 2). Relative to the user, the system is sentient. It senses the relationship between the data and the terminal and acts accordingly, bringing up related yet terminal-specific data that the user would expect at a terminal of this type and location. Therefore, we call our user model *sentient data access*.

While this example does not show many of the complexities that can arise in different situations, it does demonstrate the basic components of our user model. Formally, this user model contains three components:

- Terminals—Fixed-location devices.
- Identifiers—Either physical or virtual pointers to data (such as a URL or bar code).
- Containers—Mobile wireless devices that can carry identifiers, as well as serve as a personal portable UI to computational devices (terminals) distributed in the physical environment.

As in our automotive design studio example, the terminals are fixed-location devices designed to perform specific, often complex, tasks. These terminals may include desktop workstations, touch-sensitive plasma

SENTIENT DATA ACCESS

via a Diverse Society of Devices

panels, large-display projection screens, and other more specialized devices. Each has a user interface that enables a person to interact with it directly. Typically, they also afford interaction through the UI of the portable container device. As we shall see, the complexity that would result from having to learn and interact with a number of diverse terminals can be reduced or eliminated by converging on a consistent approach to their user interfaces. Thus, due to its specialized nature, each device is less complex than the general-purpose alternative. At the same time, overall complexity is reduced if one can leverage the transfer of skills from device to device, due to the consistency of their UI design. We hope that our examples will illustrate that, with appropriate design, one can have one's proverbial cake and eat it too.

Note that we do not need completely different terminals to perform different tasks. For example, identical terminals at different locations may be dedicated to different tasks. This is analogous to an office on one floor being dedicated to accounting, while an identical office on a different floor is used for quality assurance. Departments (i.e., function) can be identified by location and terminal type, or both.

While the terminals are used to display and

interact with data, identifiers are keys to access the data. From the user perspective, identifiers include UPC symbols, RF (radio frequency) tags, and Smart Badge ID numbers that allow integration with physical artifacts, as well as (URLs), which allow integration with Web assets. When working with virtual assets already in the system, the displayed representation of the asset itself can act directly as an identifier (see figure 3).

Containers, or wireless mobile devices, primarily serve as a mechanism for easily transporting data identifiers among terminals. Sample containers include PDAs, cell phones, bar-code readers, and Smart Cards (see figure 3). Some devices can be both a container and a terminal. These types of devices not only hold and transport an identifier, they can also allow some interaction with the associated data. For example, a PDA transporting an

FIG 3

(a) Sample containers including Symbol PDA (<http://www.symbol.com>), cell phone, and tablet computer. The image displayed on the tablet computer can act as an identifier. (b) Sample identifiers including a Smart Card and bar code.



Spatial Layout of Our Trial Environment Including Terminal and Task Pairings

FIG 4



image identifier can also display and allow for machine manipulation of a version of the image itself. A container can also work in concert with a terminal, serving as an extension of the terminal's user interface. This is particularly useful when working with terminals that have limited input functionality.

There are two fundamental challenges in creating systems of these types. The first is having the system predict, given an identifier, which representation of the data should be loaded onto the terminal. The second is providing a way for the user to choose an alternate representation when the system does not correctly predict which representation the user desires. Given these fundamental concepts, we now explore their use in an experimental environment consisting of a heterogeneous society of devices.

AUTO DESIGN STUDIO AS TRIAL ENVIRONMENT

We have been working with automotive designers for a number of years and have a fairly deep appreciation of the problems they face in their workflow within this media-rich environment. Given this background, the automotive design studio is an appropriate application domain for our trial environment.

A typical automotive design studio supports a workflow that involves a myriad of data types, including: two-dimensional concept sketches; computer-rendered images; animations and movies of cars in various environments; 3-D clay and computer models at various scales; interior textures and fabrics; and engineering data. In addition, the studio needs to facilitate data flow among a divergent set of processes—including conceptual development; interior and exterior specification; engineering designs and constraints; design review and evaluations; and, finally, manufacturing. The different tasks in this workflow are typically performed by different people, at different locations, and often using very different and specialized hardware and software. This is an ideal environment to test our conceptual framework for sentient data access using a society of devices.

To facilitate this diverse workflow, our trial environment contains various terminal types, each suited for a specific task (see figure 4).

The largest terminal is a 6-by-8-foot rear-projection screen (see figure 5a). In real auto design studios, even larger display screens, called Powerwalls (see figure 5a), are being widely installed. These large displays function as *awareness servers*, which ambiently display imagery of two-dimensional and 3-D content, giving designers in the studio the context of their peers' work. Powerwalls

are also well suited for the evaluation of designs of 3-D car exteriors, especially when full-scale visualizations are desired. They can be used as general-purpose screens for presentations to large audiences.¹⁰

While the large scale of the Powerwall display facilitates full-scale viewing, the flat nature of the screen does not provide the viewer with any sense of immersion. Figure 5b shows our second large terminal: the Vision Dome—a 10-by-10-foot hemispherical concave display produced by Elumens.¹¹

The hemispherical display surface provides the viewer with a greater sense of immersion than a typical flat-screen display. When viewing designs for the interior of cars, for example, this enhanced sense of immersion provides a better idea of what it would be like to actually sit inside the car. Furthermore, since this immersion is facilitated without encumbering stereoscopic hardware, subtle human body-language cues, such as eye gaze, are not obscured. Viewers' ability to interact with one another while using the terminal is thus uncompromised. However, easy interaction with the surface of the display itself is precluded by the size and shape of this terminal, and the fact that viewers should stand several feet away from the display to get maximum immersion. To counteract these factors, we provide an auxiliary 15-inch touch-screen display, mounted at waist height in front of the terminal, to serve as an interaction portal.

Our third terminal is one of medium scale: a high-resolution 51-inch plasma display with an overlaid transparent digitizing surface (see figure 5c). We use this terminal primarily as an asset-awareness server. Running our PortfolioBrowser software, various digital assets such as images, 3-D models, animations, and movies can be

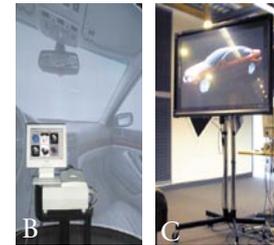


FIG 5 Terminals
(a) Powerwall;
(b) Vision-Dome; (c)
Plasma

SENTIENT DATA ACCESS

via a Diverse Society of Devices

easily accessed, compared, sorted, and annotated. Furthermore, when not actively being used, the terminal goes into an ambient mode that cycles through the various assets. Much like the corkboards of the past, only more dynamic, this provides for an ambient display that casually increases awareness of the various assets related to projects being worked on in the studio.

In addition to these three medium- to large-scale terminals, we have a more specialized terminal called the Chameleon^{12, 13}—a high-resolution touch-sensitive LCD panel tracked in 3-D space by an articulated arm (see figure 6). This terminal is a specialized viewer that makes inspection of a 3-D model intuitive by allowing a user to move around in 3-D space by physically moving the display. In effect, the display is a moveable window into the 3-D space.

Along with the specialized terminals described above, our space is populated by various status-quo PC workstations, used for engineering, design, and model-building applications.

Envisioned Usage Scenario. We envision a usage scenario that involves coordinated use of all these terminals. While they are all interconnected at the systems level, from the user's perspective, a seamless mechanism for transporting work from one device to another is highly desirable. For example, a user may first view a car's exterior design on the plasma display, and then move to the VisionDome to get a better understanding of the car's interior.

Using current status-quo user interfaces to accomplish this can be cumbersome. The user would first have to determine the name of the file that is related to the picture of the car's exterior, then determine the name and location of another file, which contains the data for this car's interior suitable for display on the Vision Dome. Finally, on the Vision Dome, the user would have to navigate through a file browser to load this file.

The intention of our sentient access user model is to alleviate the complexity of this transaction. A much improved user interface results by using off-the-shelf mobile devices, such as PDAs with wireless connections, as containers for transporting information between

terminals. In our previous example, a user could transfer a digital asset's identifier by tapping on the image of the car's exterior on the plasma display, then tapping the screen of the handheld PDA device that serves as a container. This pick-and-drop metaphor¹⁴ is an extension of the typical drag-and-drop action found on desktop interfaces. The user then walks over to the VisionDome with the container, and uses a similar pick-and-drop gesture from the container to the dome to load the files relevant to the given digital asset's identifier.

The key here is that the software has to be smart enough to know that the car's interior designs should be loaded on the VisionDome, despite having received an identifier from the car's exterior that was being viewed on the plasma display terminal. The representation most appropriate for a given location and a given terminal's affordances is chosen by default. The user, meanwhile, does

Chameleon Terminal

Includes touch-sensitive display on an articulated arm used for 3-D viewing of car models



FIG 6

Sample Container Device

Our sample container device, from Symbol Technologies consists of a ruggedized PalmPilot, wireless 1-Mbit network connection, and a barcode scanner.



FIG 7

not need to be concerned with low-level systems issues such as filenames and directory structures.

Just as we have a diversity of terminals, we also have a diverse set of containers. Some container devices may have mechanisms for dealing with different identifier technologies, such as UPC bar codes and RF tags. A PDA with a wireless network connection and a bar-code reader, for example, can be used to scan bar codes to access digital assets (see figure 7).

This identifier to the asset can then be transported to other terminals as described earlier, resulting in a “scan-and-drop” metaphor. An advantage to using bar codes is that we can also integrate physical assets into our system. For example, bar codes on 3-D clay models can be read and used as identifiers to access associated digital assets on appropriate terminals (see figure 8).

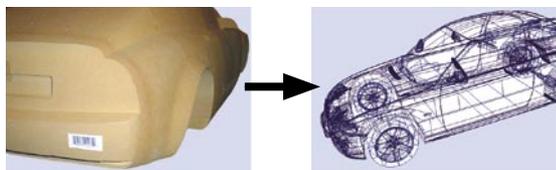
The glue that binds our diverse collection of terminals,

containers, and identifiers is a software infrastructure we call PortfolioBrowser (see figure 9).

The PortfolioBrowser currently deals with the traditional scenario in which a user has come to a terminal without an identifier in hand and needs to use the terminal as an asset browser. We envisage extending the PortfolioBrowser’s functionality to address the two challenges mentioned earlier: the need to deal with representation of the data to load, given an identifier, and addressing the case when the system does not correctly predict which representation the user desires.

As figure 10a illustrates, the default UI for our PortfolioBrowser organizes our assets by tabs. This is similar to an image-based file browser. Our intention is to extend this to organize and prioritize the data based on several criteria, including suitability of the data for a given terminal type, recent sessions, and the specific user.

FIG 8

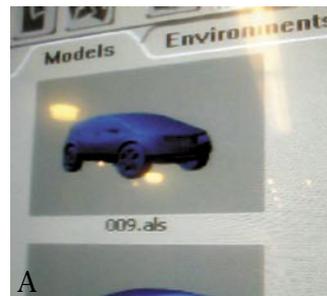


3-D Wireframe Computer Model

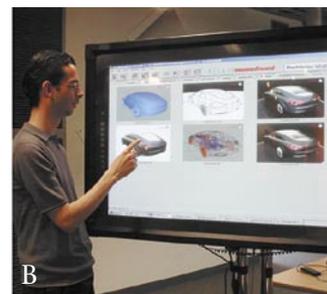
Identifiers on physical artifacts (such as the barcode on the 3-D car model) can be used to access alternative representations, as illustrated by this 3-D wireframe computer model.

Default UI for Portfolio-Browser

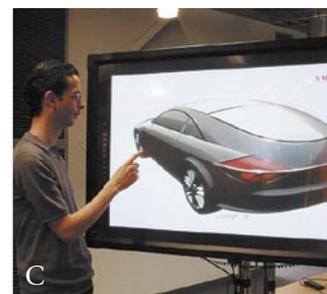
This consists of (a) user-selectable “tabs” and (b) a grid of digital assets. Touching an item in the grid selects and opens the item. Reversing the gesture (c) will return the user to the grid of digital assets.



A



B



C

FIG 9



Terminal Customized Portfolio Browser Software

This software runs on all of our different terminal types to serve as a simple and consistent UI for accessing digital assets.

FIG 10

SENTIENT DATA ACCESS

via a Diverse Society of Devices

The user can then select any data asset via this user interface for display on the terminal (see figure 10b, c). In contrast, we envision that when a user approaches a terminal with a container and sends an identifier to a terminal, the PortfolioBrowser will respond by automatically choosing the most appropriate representation and displaying the associated digital asset. If there are several choices for the appropriate representation, these choices

Coordinating the Society of Devices

(a) Using the Chameleon terminal to browse the 3-D model. (b) The Chameleon affords 3-D navigation and inspection for a single user. By establishing a link between the Chameleon and Powerwall terminals, groups can participate in the creation or review of annotations. (c) While the Chameleon device is well suited for creating annotations, reviewing and implementing the changes are best performed at a 3-D modeling workstation.



FIG 11

would be presented to the user by the PortfolioBrowser.

We maintain consistency and simplicity in the interaction by providing a common interface critical to the success of our sentient access user model. The inherent advantages of employing specific terminals for specific tasks would be defeated if moving from one terminal to another was complicated or time-consuming, and required log-in actions, along with learning a multitude of data access interfaces. Thus, the design of our PortfolioBrowser embraces our fundamental goal of minimizing transaction costs at all times, throughout the entire system.

We have focused on a user model of container-terminal interaction. Another scenario we are interested in is terminal-to-terminal communication in which the goal is to use the features of one terminal to enhance the capabilities of another. For example, a user could employ the Chameleon terminal to navigate around a car model in 3-D space, while others view the results of the navigation on a Powerwall terminal (see figure 11b).

Another important aspect to our user model is the management of connections among containers and terminals, based on proximity to one another. While terminals and containers are always implicitly connected via a wireless network, interactions between a specific terminal and container require that an explicit relationship be established. In the simplest case, when a container comes into physical proximity with a terminal, an explicit connection is automatically established without user intervention. In a more complex case, a container is in close proximity to multiple terminals. Ultimately, proximity alone may not be sufficient to determine appropriate connections. In this case, the user will need to be presented with a list of choices and confirm a connection. The important thing is to avoid having the user perform a series of initiation and setup tasks to establish a connection between a container and a terminal.

System Architecture. Given our envisioned usage and system scenario, several key underlying mechanisms are required. At the center of our system will be a relational database. The main objects in this database will be digital assets associated with an automotive design process—including sketches, 3-D models, photo-realistic renders, engineering data, market data, and animations. A content management system¹⁵ will present these assets grouped into projects. For example, a project encompasses a particular model of a car. In addition to this, the database needs to hold information on opening an object with a given application for a particular terminal. Associations between data type and application are normally handled

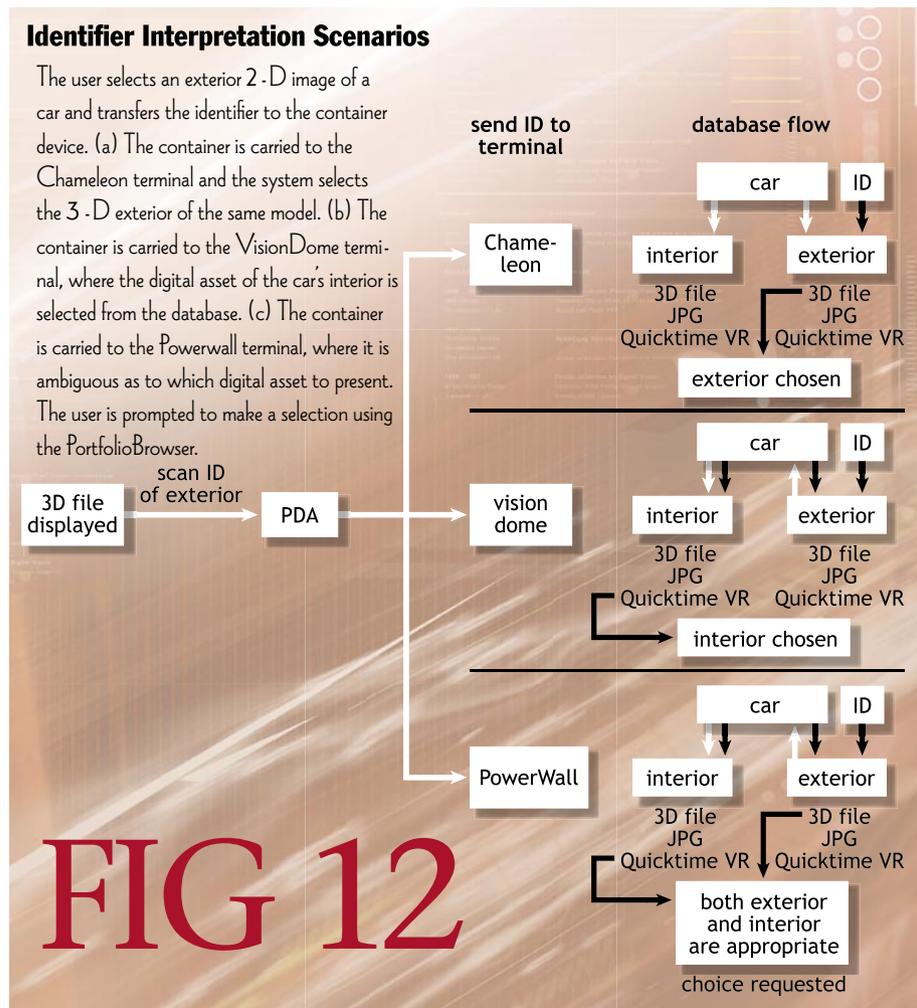
by the operating system. Unfortunately, current operating system mappings of data types to application do not factor in the terminal properties. Therefore, an important component of the database will be the mapping between data type and the target application that may depend on the terminal type. Having this terminal information will allow us to retrieve the correct assets for a particular terminal using database queries on given identifiers, as illustrated in figure 12.

The complexity of matching a given identifier with a particular terminal at a certain location, while accounting for a number of contextual states, requires adaptive, programmable heuristics to deliver the appropriate asset. To compound the complexity, the time of day or the presence of other people may influence the choice of asset presented. Initially, a set of preprogrammed rules will offer a default outcome. As usage knowledge is added to the system, a number of approaches may be blended to form an effective heuristic strategy.

Programming in a cooperative ubiquitous environment can be conceptualized as running an object-oriented simulator in which each computational element is abstracted into an object. Objects dynamically enter and leave the environment. A spatial layout consisting of the objects can be constructed to match the location-sensitive nature of the identifier-container-terminal user model. In this abstraction, all of the computational elements can be programmed holistically instead of individually. Furthermore, we speculate that diagnostic tools such as spatially oriented debuggers can be defined to facilitate development of sentient data access for a rich society of devices.

CURRENT STATE OF THE INFRASTRUCTURE
Within our current trial

environment we have set up the various terminals and physical stations described earlier (plasma display, Powerwall, Chameleon, VisionDome, traditional physical art board, and physical 3-D model). All of the computational terminals are functional and on a single network. A Symbol PDA acts as our container device, which is currently capable of scanning bar codes from physical artifacts, communicating to our network via a wireless connection, and serving as a portable user interface for terminals using the Pebbles software from Carnegie-Mellon University. The PortfolioBrowser software works on all of the terminals, and the architecture currently supports a shared database. However, more development is needed to fully support identifier transactions. We are continuing to develop the system infrastructure to fully support the sentient data access user model, including complete database support and customized PDA software (to support the pick-and-drop and scan-and-drop actions).





SENTIENT DATA ACCESS

via a Diverse Society of Devices

ENHANCING THE COOPERATIVE ENVIRONMENT

To some degree, data access methods have been rooted in the metaphor of accessing files in a hierarchical filesystem. Technological developments such as wireless networks, mobile computing devices, and specialized display terminals can be used to present a different, and possibly more effective, user model for data access in a modern cooperative ubiquitous computing environment. We have proposed a user model called “sentient data access,” which utilizes access context, location, and user information.

While we have used the automotive design studio as an application domain to motivate our discussion, our sentient data access model is clearly not limited to this domain. For example, other environments with a similarly rich set of tasks, assets, and media—including hospitals, biotech labs, special-effects studios, and industrial design companies—could benefit from a similar model. As the complexity in data access increases in these environments, we believe that the benefits of this seamless, intelligent user model will be all the more critical. □

ACKNOWLEDGEMENTS

The authors would like to thank Symbol Technologies, Elumens Corporation, Fakespace Labs, the Pebbles Project at Carnegie-Mellon University, the PortfolioBrowser product team at Alias, Alex Babkin, and Scott Guy for their assistance with this research project.

REFERENCES

1. Weiser, M. Some computer science issues in ubiquitous computing, *Communications of the ACM* 36, 7 (1991), 75–84.
2. Weiser, M. The computer for the 21st century. *Scientific American* 265, 3 (Dec. 11, 1991), 94–104.
3. The Anoto Group: see <http://www.anoto.com/>.
4. Toynbee, A. *Architecture, Silence, and Light: On the Future of Art*. (1970) Viking, NY, 20–35.
5. Kodak EasyShare LS633: see <http://www.kodak.com/US/en/corp/display/LS633.jhtml>.
6. Mann, S. Wearable intelligent signal processing. *Proceedings of the IEEE* 86, 11 (Nov. 1998), 2123–2151; see

- also <http://www.eecg.utoronto.ca/~mann/>.
7. Weiser, M. The computer for the 21st century. *Scientific American* 265, 3 (Dec. 11, 1991), 94–104.
 8. Fitzmaurice, G.W. Graspable user interface. Ph.D. dissertation, University of Toronto, 1996.
 9. Ishii, H. and Ullmer, B. (1997) Tangible bits: Towards seamless interfaces between people, bits and atoms. *Proceedings of ACM CHI (1997)*, 234–241.
 10. Balakrishnan, R., Buxton, W., Fitzmaurice, G., and Kurtenbach, G. Large displays in automotive design. *IEEE Computer Graphics & Applications* 20, 4 (July 2000), 68–75.
 11. Elumens Corporation: see <http://www.elumens.com/>.
 12. Fitzmaurice, G.W. Situated information spaces and spatially-aware palmtop computing. *Communications of the ACM* 36, 7 (1993), 39–49.
 13. Buxton, B., Fitzmaurice, G.W., Khan, A., Kurtenbach, G., and Tsang, M. Boom Chameleon: Simultaneous capture of 3D viewpoint, voice and gesture annotations on a spatially-aware display. *ACM CHI Letters* 4, 2 (2002) 111–120.
 14. Rekimoto, J. Pick-and-drop: A direct manipulation technique for multiple computer environments. *Proceedings of ACM UIST (1997)*, 31–39.
 15. Addey, D., Ellis, J., Suh, P., and Thiemecke, D. *Content Management Systems*, Glasshaus Publishers, Birmingham, UK, 2002.

LOVE IT, HATE IT? LET US KNOW:

feedback@acmqueue.com or www.acmqueue.com/forums

GEORGE W. FITZMAURICE, Ph.D. (gf@alias.com) is a senior research scientist in the Interactive Graphics Research Group at Alias and an adjunct professor of computer science at the University of Toronto.

AZAM KHAN (akhan@alias.com) is a Human-Computer Interaction (HCI) researcher at Alias and is currently pursuing his M.Sc. at the University of Toronto.

WILLIAM BUXTON (<http://www.billbuxton.com>) is principal of Buxton Design, a Toronto-based boutique design and consulting firm, and an associate professor of computer science at the University of Toronto.

GORDON KURTENBACH, Ph.D. (gkurtenbach@alias.com) is director of research at Alias and an associate professor of computer science at the University of Toronto.

RAVIN BALAKRISHNAN (<http://www.dgp.toronto.edu/~ravin>) is an adjunct professor of computer science at the University of Toronto.

© 2003 ACM 1542-7730/03/1100 \$5.00.

RESOURCES

Sentient Computing Project

Hopper describes the Sentient Computing project at AT&T Laboratories Cambridge. This project attempts to track the physical environment and the user's activity, and then react appropriately, depending on the user's location in the environment. For example, if a user moves into a new room, their terminal log-in session follows them to a local terminal in that room. [Hopper, A. The Royal Society Clifford Paterson Lecture: Sentient Computing. *AT&T Laboratories Cambridge Technical Report* (1999); see also <http://www.uk.research.att.com/abstracts.html>.]

Removable Media Metaphor

Ullmer and colleagues propose a "removable media" metaphor for dealing with the transport of data among devices. Their basic idea is to have physical objects, known as mediaBlocks, associated with pieces of data. These mediaBlocks, which need not have any computational power, can then be moved from one computational device to another for processing of the data. For example, to print a document, the document file can be carried on a mediaBlock from a desktop computer to a printer. The act of docking the mediaBlock in the printer initiates the print job. [Ullmer, B., Glas, D., and Ishii H. mediaBlocks: Physical containers, transports, and controls for online media, *Proceedings of the ACM SIGGRAPH* (1998), 379–386.]

i-Land Project

A similar mechanism for transporting data is described by Streitz and colleagues within their i-Land project, which interconnects computationally enabled furniture with large displays. Their mechanism allows for physical objects, called passengers, to act as a temporary container for data transport between these computationally enabled stations. [Streitz, N.A., Geißler, J., Holmer, T., Konomi, S., Müller-Tomfelde, C., Reischl, W., Rexroth, P., Seitz, P., Steinmetz, R., and i-LAND: An interactive landscape for creativity and innovation, *Proceedings of the ACM CHI* (1999), 120–127.]

Pick-and-Drop Metaphor

The pick-and-drop metaphor proposed by Rekimoto allows for transfer of data from device to device in a technique that is an extension of the typical drag-and-drop action found

on desktop interfaces. The idea is for users to identify (pick) an item on one device, move the input device to a second device, and insert (drop) the item onto that device, causing the data to be transferred. [Rekimoto, J. Pick-and-drop: A direct manipulation technique for multiple computer environments. *Proceedings of ACM UIST* (1997), 31–39.]

System with Goal of Intuitive Manipulations

Want and colleagues describe a system whose goal is intuitive manipulations based on the coupling of physical objects to representative virtual objects or actions. They do this by augmenting everyday objects with sensor tags. Actions take place when augmented objects are tapped on computational objects with sensor readers. [Want, R., Fishkin, K. P., Gujar, A., and Harrison, B. Bridging physical and virtual worlds with electronic tags. *Proceedings of ACM CHI*. (1999), 370–377.]

ParcTabs

The ubiquitous computing project at the PARC utilized small mobile devices, called ParcTabs, which were designed with four context-specific behaviors in mind: (1) stand-alone unit away from the network, (2) in the building as a networked appliance, (3) in a room with an electronic whiteboard and used as a telepointer, and (4) next to the electronic whiteboard used as a metaccontroller in the left hand, while a stylus is used in the right hand. [Want, R., Schilit, B. N., Adams, N. I., Gold, R., Petersen, K., Goldberg, D., Ellis, J. R., and Weiser, M. An overview of the ParcTab ubiquitous computing experiment. *IEEE Personal Communications* 2, 6 (1995), 28–43.]

In many ways, our work is similar to aspects of all of these previous systems. However, we propose a formal user model, and from an implementation perspective, we use networked computational devices as mobile containers rather than static physical objects to transport identifiers. Furthermore, while the identifiers in the previous systems serve as single, simple links to particular objects or actions, identifiers in our system are more complex because they serve as a pointer to a set of possible actions. From this set, the system intelligently selects the most appropriate action based on context. This context depends on several factors, including the type and location of each terminal, thus leveraging the configuration of our society of devices to promote seamless data access.