CHAPTER 6

From GUI to UUI: Interfaces for Ubiquitous Computing

Aaron Quigley

CONTENTS

6.1 INTRODUCTION

The user interface represents the point of contact between a computer system and a human, both in terms of input to the system and output from the system. There are many facets of a "ubiquitous computing" (ubicomp) system from low-level sensor technologies in the environment, through the collection, management, and processing of context data through to the middleware required to enable the dynamic composition of devices and services envisaged. These hardware, software, systems, and services act as the computational edifice around which we need to build our ubicomp user interface (UUI). The ability to provide natural inputs and outputs from a system that allows it to remain in the periphery is hence the central challenge in UUI design.

For our purposes, ubicomp is a model of computing in which computation is everywhere and computer functions are integrated into everything. It will be built into the basic objects, environments, and the activities of our everyday lives in such a way that no one will notice its presence [Weiser, 1999]. Such a model of computation will "weave itself into the fabric of our lives, until it is indistinguishable from it" (Weiser, 1999). Everyday objects will be places for sensing, input, processing along with user output (Greenfield, 2006). Take, for example, the multiperson interactive surface in Figure 6.1. Here, dozens of people can interact simultaneously with a large historical record using gesture alone. Coupled with directional microphones, personal displays, and other forms of novel interface, one can imagine this as part of a larger system in the future. This future experience might bring schoolchildren back to past events not just in one room but throughout a city, country, or continent. Ubicomp aims to make information, applications, and services available anywhere and at anytime in the human environment, where they are useful. In keeping with Weiser's original vision of keeping technologies unnoticed (Weiser, 1999), a further aim is to have all this delivered in a fluid manner appropriate to our current context (Coutaz et al., 2005).

Around the world, research and development groups are exploring mobile and embedded devices in almost every type of physical artifact including cars, toys, tools, homes, appliances, clothing, and work surfaces.

FIGURE 6.1 Shared public display at the Cabinet War Rooms and Churchill Museum, London.

Indeed, anywhere computation will aid the user in solving a problem or performing a task in situ, ubicomp can be viewed as the model of computation. Ubicomp represents an evolution from the notion of a computer as a single device, to the notion of a computing space comprising personal and peripheral computing elements and services all connected and communicating as required; in effect, "processing power so distributed throughout the environment that computers per se effectively disappear" (Greenfield, 2006) or the so-called *calm computing*. It is important to note that the advent of ubicomp does not mean the demise of the desktop computer in the near future. The ubiquity of this technology took decades to advance and we can expect the same gradual evolution in ubicomp technology and scenarios of use.

Many ubiquitous computing scenarios suggest introducing new affordances, features, services, and modes of interaction into the simplest and most basic operations of our daily lives, from turning a door handle while getting haptic feedback to convey a status update through to the augmented cooking experiences in kitchens with radio frequency ID (RFID), computer vision, speech recognition, and projected and embedded displays. For example, Figure 6.2 shows a multidisplay gaming environment suitable for multiperson game playing, coordination tasks, or planning activities. Topics covered elsewhere in this text, such as communication, systems design, context awareness, and privacy, are all crucially important

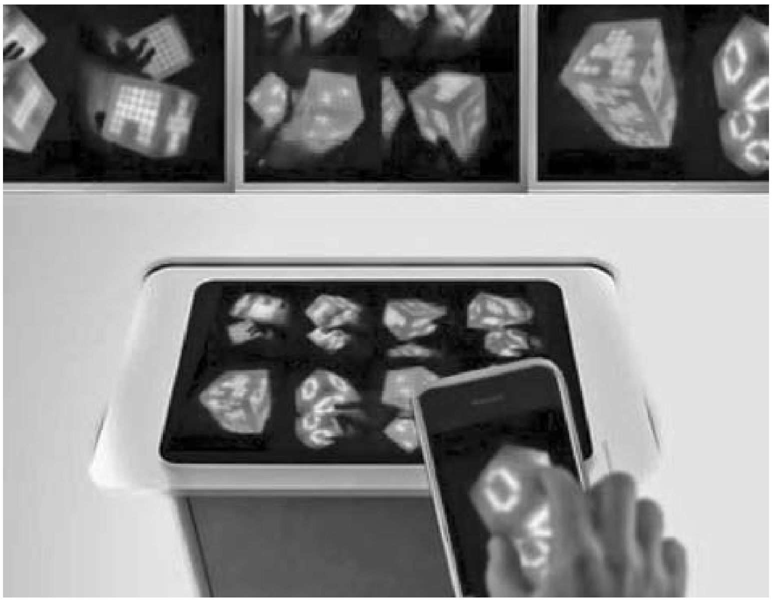

FIGURE 6.2 Coupled personal (iPhone) and public display (MS Surface) gaming environment.

to realizing such an infrastructure; however, so, too, are the interfaces with which we will interact with such systems.

It is clear that future ubicomp applications and deployments will rely on new devices, services, and augmented objects, each supporting well-understood affordances, but also simple computational adaptations to existing well-known and well-understood devices, services, and objects. As more and more computation is woven into the fabric of our lives, our interaction with such ubicomp systems cannot be the focus of our attention. As noted previously, if every physical object in our home demanded attention in terms of alerts, updates, and configurations in the way our current personal computers do, we would become quickly overwhelmed. Indeed, it has been noted that "… the benefits of inexpensive Ubiquitous Computing may be overwhelmed by its personal cost" (Heiner et al., 1999). Instead, the computational and activity support offered by ubicomp systems must reside in the periphery of our attention and should remain unnoticed until required.

It is important to note that myriad processing elements, sensors, displays, and outputs are not elements of a statically defined user interface akin to the keyboard, mouse, and screen of a desktop computer. Instead, these elements and their related technologies afford the user an experience and a place to interact. Keeping in mind that the goal is to develop and deploy systems and technologies that are calm and invisible, how do we provide UUI cues and feedback to users leveraging the full range of

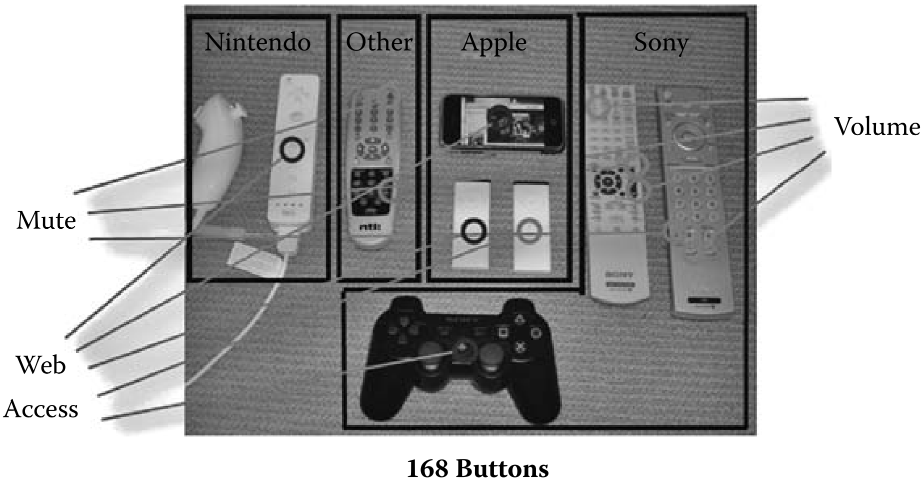

FIGURE 6.3 Three mute options, four volume controls, five web access methods, and 168 buttons for one television with six inputs.

their human senses? By contrast, take the standard appliance model one might find in a home. Figure 6.3 presents eight control devices that can be used to interact with one television that has six devices connected—DVD, AppleTV, Mac Mini (using VNC on iPhone, and Apple remote), PS3, Wii, and Cable Box. Aside from the 168 buttons available, there are in addition, a myriad of ways to access overlapping features. Figure 6.3 demonstrates there are three ways to mute the television, five ways to access a web browser, and four ways to alter the volume. Although not typically dealing with such a set of controllers, many people feel frustrated with just one or two. Such devices are often unclear and have too many functions available at the same level, which gives rise to the “controller hell” so many people feel. If we can learn anything from this, it should be that, a ubicomp system is made up of subsystems and we must design for the experience not the individual subsystem.

Although this chapter surveys the current state of the art to the user beyond the classical keyboard, screen, and mouse, it is important to also acknowledge that UUIs represent a paradigm shift in human-computer interaction (HCI) with input and output technologies not yet envisaged. UUIs are built around a next-generation technological paradigm that, in essence, reshapes our relationship with our personal information, environment, artifacts, and even our friends, family, and colleagues. The challenge is not about providing the next-generation mouse and keyboard, but instead making the collection of inputs and outputs operate in a fluid and seamless manner.

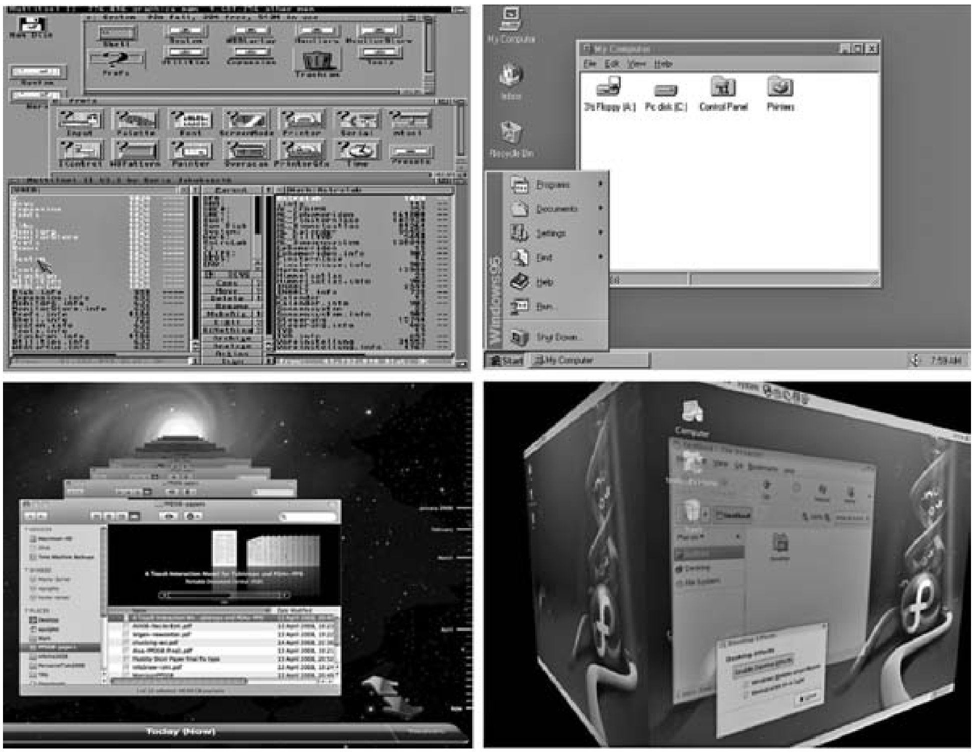

FIGURE 6.4 GUI interface evolution, Amiga UI, Windows 95, Mac OS X 10.5, and Compiz.

6.1.1 From Graphical User Interfaces to Context Data

The current graphical user interface (GUI) of the personal computer is built around keyboard, screen, and mouse devices. GUIs offer the Windows, Icons, Menus, and Pointer (WIMP) metaphor as a more intuitive view of the computer as compared with the more classical command line (textual) interface. GUIs have evolved considerably since they were first introduced, due to a better understanding of the user through HCI research along with advances in hardware and software. From rudimentary graphics through to the exploration of three-dimensional (3-D) elements as shown in Figure 6.4, GUI has grown to offer users better features.

Other technological advancements of the past 15 years have also seen the widespread adoption of mobile phones with keypads, personal digital assistants (PDAs) with styli, tablets with styli, and touchscreens with fingers. More recent developments have seen the adoption of game controller inputs or gesture-driven controls for game platforms. In spite of these advancements, many elements of the WIMP metaphor for HCI live on, be it in gesture, mouse, stylus, or finger form. The UUI draws on the GUI tradition but moves beyond the special-purpose device (PDA, phone, laptop, desktop) and into the support of activities of our lives, some of which are not currently supported by computation.

Drawing on technologies both in the research stage and also in widespread adoption, this chapter describes many of the foundational input and output methods, systems, and technologies that will help us realize future UUIs. Indeed, we must consider the full range of interaction styles available for input beyond the mouse, keyboard, stylus, or controller. Although not yet widely available, technologies and methods exist to both sense or infer a much wider range of human motion, activity, preference, and actions desired. In general a UUI must consider a broader range of inputs than current desktop, mobile or gaming devices, and applications. Examples of data or knowledge a UUI can rely on include spatial information (location, speed of movement), identity (users and others in vicinity), user model (profile, preferences, intentions), temporal data (time of day or year), environmental (noise, light), social (meeting, party), resource availability (printers, fax, wireless access), computing (network bandwidth, memory), physiological (hearing, heart rate), activity (supervision, interview), schedules, and agenda along with data that can be mined and inferred.

Broadly speaking, the range of implicit inputs to a ubicomp system via its UUI can be called context data whether sensed or inferred. Context includes information from the person (physiological state), the sensed environment (environmental state), and computational environment (computational state) that can be provided to alter an applications behavior. Explicit inputs from the user can include speech, gaze, or human movement of any type. Other chapters in this book focus on the processing and management of context data, whereas this chapter focuses on explicit user input along with explicit and peripheral outputs.

6.1.2 Inventing the Future

The time scale for the advancement and adoption of ubicomp systems and technologies may be unlike anything seen before in computing research and development. The adoption of this model of computation may reach a tipping point in key domains rapidly or it may be a slow process of features, functions, and indeed changes in expectations of what constitutes a computer, seeping into the general public consciousness. Weiser (1999) cites the symbolic representation of spoken language in written form as a ubiquitous technology. However, written language has developed over millennia and requires years of learning at a preschool, school, and even university level. Hence, the time scales involved in widespread adoption of a "ubicomp grammar" and hence interaction language may be longer than expected. Thankfully, researchers in academia, commercial laboratories, and industrial development have been producing ubicomp prototypes for more than 15 years. As a result, we have a rich assortment of research and experience to draw upon when considering UUIs.

Regardless of the eventual time scales of adoption, when planning a UUI for a particular ubicomp system, technology, or application, five fundamental questions typically arise:

1. Are the current generation of technologies applied to the problem sophisticated and aesthetic enough to afford a computationally enhanced experience that is realistic based on current practice?
2. Does the interface rely on well-understood metaphors and affordances? The WIMP metaphor has buttons, clicks, and drag-and-drop, but what can UUI designers and developers expect to use and build upon? In effect, what are the nouns, verbs, and grammar of ubicomp interfaces?
3. Does the proposed solution actually help people solve real problems or perform current tasks? Many of the technological scenarios used to motivate particular ubicomp methods, techniques, or technologies are aspirations of future needs and desires.
4. Will the costs or overheads involved prevent widespread adoption of these concepts into actual consumer use, that is, are the economics of this ubicomp solution sound?
5. When considering a UUI, if no one is expected to notice the presence of computation in the artifacts in their environment, then how are they expected to interact with them?

Even a cursory review of the ubicomp literature shows a range of technologies, methods, scenarios, and systems pushing the envelope of what is classically described as HCI. Some might describe many of these ubicomp systems as "solutions looking for a problem" but others classify them as attempts to invent the future. Alan Kay, the computer scientist, said, "Don't worry about what anybody else is going to do. The best way to predict the future is to invent it."

Current UUI development relies on a range of established and evolving user interface classes as described in Section 6.3. As new forms of sensing, actuation, technology, and interaction develops, we will see the creation of new interaction styles that may be hence classified as new classes of user interfaces themselves. Section 6.3 reviews the classical range of interaction styles considered in HCI and then describes three evolving classes of user interfaces that are central to the realization of UUIs. Table 6.1 describes

TABLE 6.1 Ten Rules for UUI Design

Rule	**Meaning**	**Example**
Bliss	Learning to interact with a new UUI should not require people to learn another skill or complex command language.	Good interaction design as discussed in Section 6.2
Distraction	Do not demand constant attention in a UUI. Inattention is the norm not the exception.	Ambient User Interfaces as discussed in Section 6.3.3
Cognitive Flow	Ubicomp systems that are everywhere must allow the user to retain total focus on the task at hand.	Multimodal interfaces as discussed in Section 6.4.3
Manuals	Do not require a user to read a manual to learn how to operate the current UUI. Do leverage prior experience.	Use of affordances (e.g., UI overlay) on real world in Figure 6.5
Transparency	Do not rely on users to hold application state in the mind to operate the UUI.	Tangible User Interfaces as discussed in Section 6.3.1
Modelessness	Avoid “modes” where the system responds differently to the same input stimulus dependent on some hidden state information.	State visible in SharePic as shown in Figure 6.8 (Section 6.3.2.1)
Fear of Interaction	Provide easy means to undo actions, otherwise users may become paralyzed with fear when using the UUI.	Use of well-understood design patterns as discussed in Section 6.2.4
Notifications	Feedback to the user can be piggybacked and layered into interactions with their physical environment.	Display of power usage as shown in Figure 6.12
Calming	Interfaces will support situated actions, interfaces will rely on a wide array of human inputs and human senses.	Surface interfaces as shown in Figure 6.6
Defaults	Good interfaces judiciously exploit what the system knows or can deduce.	Applications that reuse user input

the 10 key rules for UUI design. Each rule expresses either an aspect of the human state (e.g., maintaining bliss and not disturbing cognitive flow) or a base system requirement to ensure ease of use (e.g., no need of manuals and easy undo). Those developing a UUI should consider these rules from the start of the ideation process through to system design and development.

6.1.3 Chapter Overview

With all the artifacts in our environment imbibed with computation, input, and output, the future might become a very noisy, unpleasant, and controlling place. As such, it is important to study how to avoid the mistakes of the past in poor interface design while also relying on new technologies to help realize this vision of a useful future invisible "Everyware" (Greenfield, 2006) where we remain in control. Section 6.2 describes what is good interaction design for people in terms of affordances, metaphors, and the human action cycle. This is described in terms of user-centered design (UCD) in Section 6.2.1, systems design in Section 6.2.2, genius design in Section 6.2.3, and how patterns of interaction can be codified into design patterns in Section 6.2.4. The realization of a UUI can rely on a range of well-established user interface classes such as the GUI. Section 6.3 describes different classes of user interface. Three emerging classes of user interface that provide the natural interaction styles to allow interaction to remain in the periphery are described in Section 6.3.1 on tangible user interfaces (TUIs), Section 6.3.2 on surface user interfaces (SUIs) and particularly larger ones in Section 6.3.2.1, whereas ambient user interfaces (AUI) are discussed in Section 6.3.3. Three novel forms of input modality are described in Section 6.4, specifically sensor input in Section 6.4.1, gesture input in Section 6.4.2, and speech input in Section 6.4.3. We conclude this chapter with a brief discussion of some suggested UUI usability metrics in Section 6.5. These should be considered in the design, research, and development of any UUI.

6.2 INTERACTION DESIGN

Of particular note for UUI design is the field of *interaction design* (Shaffer, 2006). Interaction design is the discipline of defining the expected behavior of products and systems that a user can interact with. Donald Norman (2002) states, "far too many items in the world are designed, constructed, and foisted upon us with no understanding—or even care—for how we will use them." By contrast, interaction design aims to make products and systems usable and useful, and even engaging and fun. When considering the various aspects of interaction design for ubicomp, it is important to recall the breadth of the technological scenarios envisaged. Often, these scenarios revolve around people making connections to other people through ubicomp systems, not just connecting to the system itself. In addition, the breadth of the scenarios a UUI is required for suggests that interaction design should look to the fields of cinematography, kinesthesiology, and architecture.

Interaction design is a complex endeavor and draws on research, methods, techniques, and design guidelines from a range of overlapping and related fields and specialties including:

1. Cognitive psychology (metaphors, affordances, and mental models)
2. User experience design (storyboarding, personas, mockups)
3. Information architecture (shared data models, data stores)
4. Communication design (visual-auditory communication, graphic design)
5. User interface engineering (prototyping)
6. Human factors (human capability, ergonomics)
7. Industrial design (aesthetics)
8. HCI (new interface and interaction techniques)
9. Usability engineering (usability testing)

Good interaction design can be achieved in a number of ways depending on the complexity of the system proposed, its novelty, its degree of stability or ubiquity, and its cost. Design methodologies of interest to ubicomp include UCD, systems design, and genius design. The difference between these three approaches centers on the degree of user engagement in the process versus how much this can be abstracted away based on a whole system understanding or a confidence in the aesthetic of the designer. One approach to help bridge between these methods is to use documented design patterns. Such patterns can form a basis for exploiting lessons learned in each approach in the design of new systems based on a shared language for UUI design.

6.2.1 User-Centered Design

UCD focuses on the user's needs, problems, and goals to guide the design of a system. Users are involved at every stage of the process to help ensure that the system developed actually meets their needs and allows them to achieve their goals. As an approach to design, UCD dates back more than 30 years. It came from the realization that engineers frequently do not have the necessary skills required to develop user-friendly interfaces for computer systems. An important limitation of UCD is that most users typically cannot conceive of a radically different future and instead can only be relied on to inform design based on present needs. Henry Ford, the industrialist, is quoted as saying "if I had asked my customers what they wanted, they would have said a faster horse" (Jardim, 1970). UCD has been, and will continue to be, used in ubicomp research and development as a practical approach to entire systems design and user interface development (Fitton et al., 2005). UCD has its limitations in the face of evolving stated versus actual needs, user goals, technological shifts, or simply involving the wrong set or type of user in the process. Ubicomp research suggests that some of these limitations can be overcome by incorporating aspects of technology and cultural probes (Risborg and Quigley, 2003), contextual field research, intensive interviewing, and lag-sequential analysis (Consolvo et al., 2002) into the design process for ubicomp and its interfaces.

Many of the technological scenarios described in ubicomp literature are often beyond the current expectations, problems, and needs of users of today. From a design point of view, the solutions provided, given these scenarios, can be broadly classified as "systems design" or "genius design."

6.2.2 Systems Design

Systems design is a systematic and compositional approach to development, based on the combination of components to realize a solution in essence the development of a system of systems. A ubicomp system is typically composed of many systems including social systems (people), devices, appliances, computational artifacts, sensors, actuators, and services. In systems design, the documented or anticipated user needs and aims help set the *goal* of the system. Functional elements include computation, sensors, and actuators from the system itself. Inputs from the user are given from controls, which can be explicit (e.g., gesture) or implicitly defined (e.g., inference) on the system. Context data from the environment as a whole are sensed and matched with the goals, which drives the actuation of displays, services, or physical actuators. The actuation provides feedback (output) to the user, which should allow them to determine if the goal was met. If one removes the need for user control, then such a system can be described as self-governing or autonomic with the feedback loop helping to maintain control. Unlike desktop or Web application software development, systems design must consider and act upon a range of external factors from the real world as their inputs (disturbances, controls) and outputs (feedback, actuation). A systems design approach forces a designer to consider the entire environment in which the ubicomp system will be realized and not just one component of it. As the most analytical of the three design methodologies described here, it may be thought to appeal to ubicomp researchers who wish to ignore the user. This interpretation would be a mistake. As an analytical approach, it requires careful modeling and understanding of the implications of the user's goals, feedback, and controls provided.

6.2.3 Genius Design

Shaffer (2006) describes *genius design* as "the process of exclusively relying on the wisdom and experience of the designer to make all the design decisions." Based on their most informed opinion as to what users want and need, they design a product using their instinct and experience. This instinct and experience are developed over many successful projects and years of work. Many excellent devices, products, and systems in use today have come about from just this approach. Often, to maintain confidentiality, end users are not engaged in any manner, for example, in the design of the iPhone or indeed any Apple product. It is believed, but not confirmed, that designers in Apple must produce 10 versions of any new product feature for peer review and critique. We conjecture that this is how the majority of design decisions in ubicomp research and development are made today. A researcher invents a new physical component and interaction modality beyond the standard approach. Then, graduate students, academics, developers, or designers use their personal or collective skills to produce systems with a user interface. They do not have the time, resources, or inclination to engage a user cohort. Users may be involved at the end for testing or usability testing. Nielsen (2007) suggests that a quality user experience can be created with genius design *if* one starts by reducing risk and basing decisions on well-established usability data and studies. However, the contrived tasks often developed for usability testing are not suitable for the study of ubicomp systems and interfaces in authentic settings (Abowd and Mynatt, 2000). Instead, an iterative design process with evaluation and testing can remove the guesswork and inherent risk that the system and interfaces designed will fail. A description of the qualitative and quantitative user study techniques suitable for the evaluation and testing of a UUI can be found in Chapter 4.

6.2.4. Design Patterns

Design patterns are solutions to common design problems, tailored to the situation at hand (Tidwell, 2005). Introduced by Christopher Alexander and his colleagues in the field of architecture in *A Pattern Language: Towns, Buildings, Construction*, patterns represent a "shared language" for a discipline (Alexander et al., 1977). Patterns, now heavily used in software engineering (Gamma et al., 1994), are used to communicate common problems and appropriate solutions. Ubicomp researchers have started to describe common design patterns for this field (Landay and Borriello, 2003). Examples identified include Infrastructure (e.g., Proxies for devices), Privacy (e.g., Privacy Zones), Identification (e.g., Active Badge), Anticipation (e.g., "Follow-me Music"), Global Data (e.g., Interface teleporting), Discoverability, Capture and Access, Physical Space, and Location-based Services (e.g., Bluestar (Quigley et al., 2004)). A design pattern consists of 3–5 pages of text and provides general but descriptive methods for solving a particular problem. The text describes the background and how it might be fused with other patterns. A problem statement encapsulates a recurring domain specific problem. The answer identifies common solutions along with details, suggestions, and tradeoffs.

6.2.5 Discussion

With or without established patterns or an iterative process, UCD, systems design, and genius design approaches all have their strengths and weaknesses. Attempts to invent the future can be curtailed by the current generation of users who cannot imagine a future where this new device, service, or interface will be cheap, desired, or even required. However, this does not mean that the current generation of users should be ignored when researching and developing user interfaces for ubicomp systems. Instead, in each approach, user involvement should be layered in according to the assumptions around the eventual context of use for the system. This may be a UCD process where the ubicomp system solves a clear currently existing problem but relies on new techniques, methods, or infrastructure that are not yet commonplace or it may be an iterative systems design process where the ubicomp system affords users a new way to interact, relying on a novel combination of modalities with new appliances and affordances.

The process of design, or specifically interaction design, must draw on the available components, systems, interface elements, and modalities useful to realize a distraction-free, blissful, and calming experience. To understand the wide range of interface possibilities, we must first consider a range of "classes of user interface" currently available.

6.3 CLASSES OF USER INTERFACE

The user interface represents the point of contact between a computer system and a human, both in terms of input to the system and output from the system. The realization of this point of contact can exist in many forms and in many classes of user interface. In classical HCI texts, six classes are described that include command language, natural language, menu selection, form filling, direct manipulation, and anthropomorphic interfaces (Dix et al., 2003).

Direct manipulation embodied in the GUI has made computers accessible to many people. GUIs rely on visual and spatial cues, which are faster to learn, are easier to remember, and provide context for the user. Command languages require the user to remember concise yet cryptic commands. With rapid touch typing skills, these languages often appeal to experts but can strain one's power of recall. For example, consider the following UNIX commands

```
tail-n100 mynumbers|grep “^+353”
```

when issued at the command line will return any of the last 100 lines of the file mynumbers that start with string +353. Powerful indeed, but it requires detailed recall of command syntax to perfect.

Menu selection relies on recognition but not recall and can be easily supported on almost any visual device. However, this can create complex menu hierarchies and with audio systems users can easily get lost in (e.g., telephone menus). Form filling requires minimal training to use but is limited to data collection-style applications. Many of the most powerful web commerce applications are currently built around form fill-in interfaces. Research and usability has shown these reduce input errors and hence improve the user experience.

A natural language interface relies on a command language that is a significant subset of a language such as English. For example, IKEA offers an Online Assistant on their Web site that responds to typed English sentences. Related to natural language are humanlike or anthropomorphic interfaces that try to interact with people the same way people interact with each other. These are typically realized with interface characters acting as an embodied humanlike assistant or actors. Anna, the IKEA Online Assistant, has minimal anthropomorphic features through the use of 2-D animation.

From the WIMP paradigm developed in the Xerox Alto in the 1970s to the latest 3-D interface elements seen in Windows 7 and Mac OS X, the GUI has tended to dominate what is considered a user interface. However, clearly, a keyboard, screen, and mouse with GUI elements tied to every device affording computational interaction cannot be the future. The advent of the UUI will draw on elements from all these classes of interface and more.

For research and development, there are many issues in how elements of these well-established classes of user interface can be incorporated into future UUIs. There exist many further types of input technologies that we discuss in Section 6.4 which do not cleanly fit into any of these six classes because they rely on new devices. Examples include body movement in the form of gesture, speech, ambient feedback, surface interaction, and augmented reality (AR). Clearly, many new classes of user interface are being defined beyond these desktop-computing-bound six. Examples of this new class of interface include

- Tangible User Interface
- Surface User Interface
- Ambient User Interface

Alternate classes of user interface include gestural user interfaces, touch-based user interfaces, pen-based user interfaces, exertion interfaces (Mueller et al., 2007), and context-aware user interfaces. Two classes of interface, each deserving a chapter in their own right, which we do not explore in detail here, are AR user interfaces and multimodal user interfaces. An AR user interface (Figure 6.5) overlays computer graphics onto the image of a real-world scene typically with the aid of a supporting software interface framework (Sandor and Klinker, 2005). This scene can be viewed through a head-mounted display or a standard personal or desktop display. AR systems have been used to provide a graphical augmented-reality view of industrial equipment in the technician's immediate vicinity (Goose et al., 2003), Invisible Train Games (Wagner et al., 2005) as shown in Figure 6.5, or Human PacMan (Cheok et al., 2004). Multimodal interfaces attempt to take inputs from two or more of the inputs described here and "fuse" the disparate inputs into one. The motivation is to overcome the inherent limitations in any one modality with the expressiveness and error checking possible with multiple ones. For further details, we refer the reader to these guidelines for multimodal user interface design (Pavlovic et al., 1997; Reeves et al., 2004).

For the purposes of this UUI chapter, we limit our discussion to these three emerging areas (TUI, SUI, AUI) that rely on basic metaphors and

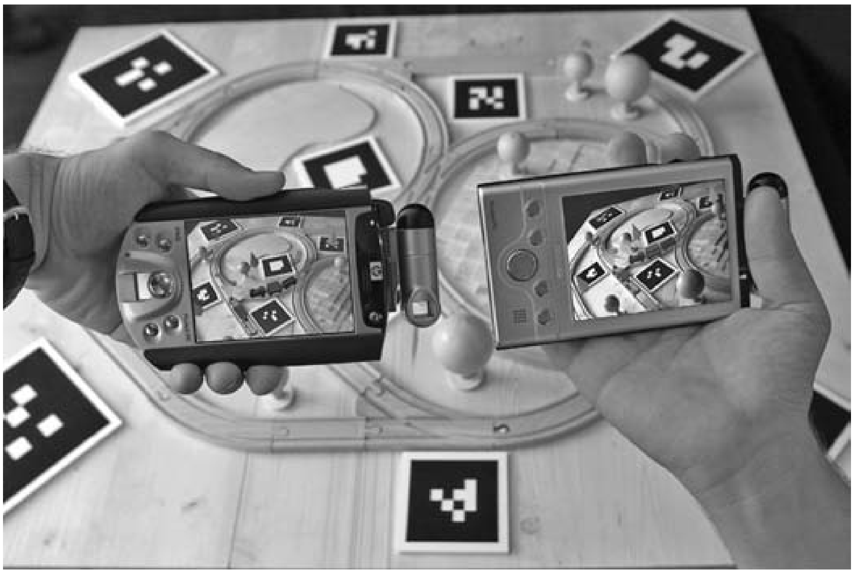

FIGURE 6.5 Augmented Reality invisible train game. (Accessed from Graz University of Technology, http://studierstube.org/handheld_ar/media_press.php)

well-understood affordances. These areas are both substantive and representative fields of user interface study for ubiquitous computing.

6.3.1 Tangible User Interface

Unlike a GUI, which presents manipulable elements virtually onscreen, a TUI integrates both representation and control of computation into physical artifacts. In essence, TUIs help provide physical form to computational artifacts and digital information. A "… TUI makes information directly graspable and manipulable with haptic feedback" (Ullmer and Ishii, 2000). The literal definition of the TUI as having both input and output factored into the same physical artifact has evolved over time (Fishkin, 2004). Accordingly, the accepted TUI paradigm is now: a user manipulates a physical artifact with physical gestures, this is sensed by the system, acted upon, and feedback is given. TUIs attempt to bridge the digital-physical divide by placing digital information in situ coupled to clearly related physical artifacts. The artifacts provide sites for both input, using well-understood affordances, and can act as output from the system. Unlike the GUI, a classical TUI device makes no distinction between the input devices and the output devices. Current TUI research and development attempt to form natural environments augmented with computation where the physical artifacts in that environments are digital embodiments of the state and information from a system.

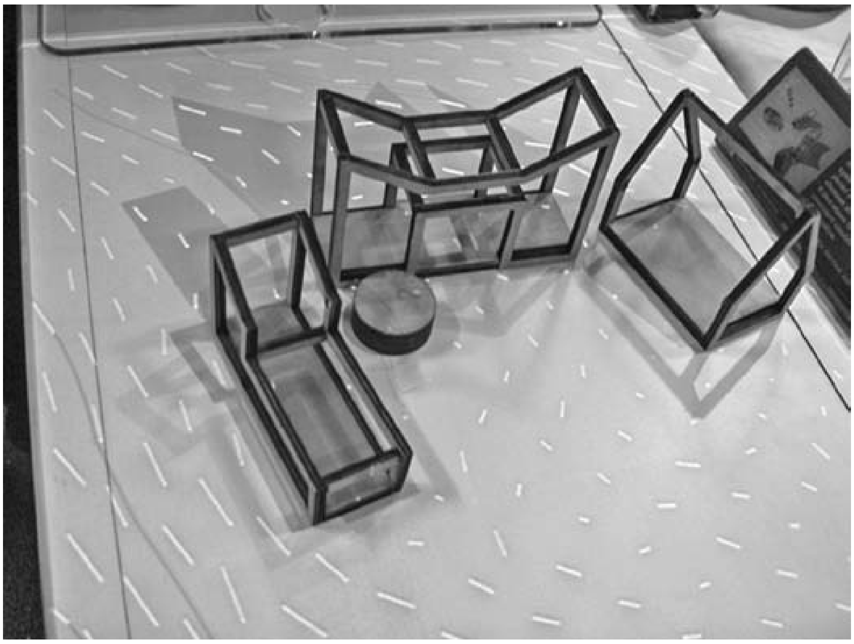

FIGURE 6.6 Urp, architectural 3-D shapes with projected shadows beneath. (Copyright Tangible Media Group. Photograph taken by Dana Gordon.)

A stated aim for ubicomp is to make information, applications, and services available anywhere and at anytime in the human environment. Although TUIs are not yet ubiquitous in our daily lives, point examples drawn from both research and deployment can be discussed with reference to Weiser's original vision of "computation in basic objects, environments and the activities of our everyday lives in such a way that no one will notice its presence" (Weiser, 1999).

For example, "Urp" is a tangible workbench for urban planning and design (Underkoffler and Ishii, 1999) shown in Figure 6.6. In Urp, the basic objects are physical architectural models (houses, shops, schools) that can be placed by hand onto the workbench. The environment for Urp appears as a typical design desk with the computational infrastructure for the I/O Bulb (projector, vision system, computation) hidden away. Indeed, even in the case of no power, the desk and models still provide a view of the design space. Layered on top of the physical artifacts (desk and building models) are aspects of an underlying urban simulation including shadow, reflections, and wind flow. The seamlessness of this system relies on the natural interaction, rapid response, and in situ nature of the simulation data presented so that its presence is not noticed. The simulation data and interactions appear natural, obvious, and in keeping with the task at hand. Although not "visually invisible," Urp may be "effectively invisible" in action or behavior to the designer. This distinction between literal and effective invisibility is important to understand.

Posey is a computationally enhanced hub-and-strut kinematic construction kit (Weller et al., 2008). As with all types of construction kits such as Lego, Meccano, or Tinkertoy, Posey allows for a wide range of physical creations to be realized. In Posey, the basic objects are plastic childlike toy pieces, consisting of struts with a ball at each end and hubs with from one to four sockets. The environment for Posey can be a typical child's play area and the computational infrastructure for optocoupling with infrared (IR) light-emitting devices (LEDs) and photosensors and ZigBee wireless transmission remains hidden away. As with Urp, even without power, computation, or other digital-physical applications, Posey remains a usable toy. It is, after all, a TUI, which affords construction kit building activities. Aspects of the computationally enhanced system are currently rudimentary with Puppet Show outputs shown on a local display. The current version of Posey is not as seamless as proposed future versions, where the posable actions of Posey will be fluidly linked to the creation of animated character in a 3-D world for, example. Here, the creation of an animated bear puppet from physical interaction may allow future versions of this TUI to be invisible in action.

The field of TUI includes research and developments including areas such as audio systems (AudioCube, Audiopad (Patten et al., 2002), and Blockjam), construction toys (Topobo, Posey, FlowBlocks), physical tokens with interactive surfaces (Microsoft domino tags, metaDesk, Audiopad, Reactable, TANGerINE), toolKits (iStuff (Ballagas et al., 2003), Phidgets), "edutainment" (DisplayCube), and consumer products (I/O Brush, Nabaztag). Core methods, technologies, and techniques have been developed that are suitable for integration into physical artifacts, each of which contains many challenging research and engineering questions.

Many ubicomp scenarios are currently realized with GUIs on handheld computers, embedded displays, mobile phones, laptop, and desktop displays. In comparison, TUIs rely on devices concealed into everyday objects and everyday interaction modalities. These modalities rely on our basic motor skills and our physical senses. Rather than simply weaving GUI displays into our world with devices such as chumbys (http://www.chumby.com) or augmented refrigerators, TUIs are sites for computation, input, and output within objects in our physical environment. For example, Phicons are a TUI whose physical appearance suggests natural mappings to corresponding computational actions (Fishkin et al., 2002;

Ishii and Ullmer, 1997). The physical appearance serves as a metaphor for action. In high-end BMWs, the shape of a car seat control resembles the shape of the seat itself. Thus, pushing the physical button that is shaped like the bottom seat forward will cause the actual seat to move forward.

Fishkin [18] details two dimensions of *embodiment* and *metaphor* to describe a taxonomy with which to describe TUIs. Embodiment is the extent to which a user thinks of the state of computation as being embodied within physical housing of the artifact. This dimension includes four categories: full, nearby, environment, and distant (Fishkin, 2004).

1. With full, the output device is the input device.
2. With nearby, the output takes place near the input object, typically, directly proximate to it.
3. With environment, the output embodiment is around the user.
4. With a distant embodiment the output is "over there" on another device.

The dimension of metaphor relates physically afforded metaphors due to an artifact's physical tangibility. This dimension includes five categories: none, noun, verb, noun and verb, and full (Fishkin, 2004).

1. For none, the physical actions with the TUI are not connected to any real-world analogy (e.g., command line style UI).
2. For noun, the look of an input object is closely tied to the look of some real-world object but this is a superficial spatial analogy only.
3. For verb, the analogy is to the gesture used.
4. For noun and verb, the physical and virtual objects still differ but are related with appeal to analogy (e.g., Urp (Underkoffler and Ishii, 1999)).
5. Full gives a level of analogy where the virtual system is the physical system. These is no disconnect.

This work unifies several previous frameworks and incorporates a view from calm computing to classical GUI interfaces. So, expanding on

Fishkin's taxonomy, we define the five key characteristics of a TUI for a UUI to

1. Provide a clear coupling of physical artifact to relevant and related computation, state, and information
2. Ensure contextually appropriate physical affordances for computational interaction
3. Ensure contextually sensitive coupling of physical artifact to intangible representation (audio/graphics)
4. Support "invisibility in action" (not literal invisibility) and natural behavior
5. Ensure a grounding of the TUI interaction design in the fields of ethnography, industrial design, kinesthesiology, and architecture

Moving TUI from research to deployment requires careful interaction design to ensure that the affordances and computational support can be invisible in action and support natural behaviors. "… the appearance of the device must provide the critical clues required for its proper operation—knowledge has to be both in the head and in the world" (Norman, 2002). It is important to recall that even the original command line interface on terminal computers relied on tangible interfaces in the form of a physical keyboard and screen. Tangibility may be necessary to help realize many ubicomp scenarios, but by itself is not sufficient, as the outputs cannot change to fit the context as the GUI can.

6.3.2. Surface User Interface

An SUI is a class of user interface that relies on a self-illuminated [e.g., liquid crystal display (LCD)] or projected horizontal, vertical, or spherical interactive surface coupled with control of computation into the same physical surface (e.g., a touchscreen). As with a TUI, the outputs and inputs to an SUI are tightly coupled. They rely on computational techniques including computer vision, resistive membrane, capacitive and surface acoustic wave detection, to determine user input to the system. They are often used in public places (kiosks, ATMs) or small personal devices (PDA, iPhone) where a separate keyboard and mouse cannot or should not be used.

The scale of an SUI can range from small personal devices such as the iPhone or PDA, through a Tablet PC up to large public interactive

FIGURE 6.7 Microsoft surface detecting device and user touch.

surfaces such as the MERL DiamondTouch (Dietz and Leigh, 2001), as shown in Figures 6.7 and 6.9, or the Microsoft Surface as shown in Figures 6.2 and 6.8. An SUI can rely on a range of input types including passive stylus, active stylus, fingers, or tangible objects, or it may be tied to just one, as is the case with the Tablet PC with its active powered stylus. Whereas the input to the SUI is simply being used as a surrogate for a mouse input, many SUI applications function as classical GUIs do and as such they can be subject to many of the same design and usability studies and evaluations. In addition, large rendered or projected keyboards can further make such systems less ergonomic and functional than their GUI equivalents with attached keyboards and mice. Clearly, this is to be avoided.

As with all types of interface, SUIs must be fit for purpose and not be foisted on users as a gratuitous replacement for desktop computing. Fitts' law still applies! This law, although published more than 50 years ago, can be used to predict the time to move to a target area, as a function of the distance to the target and the size of that target. Equation (6.1) states that the movement time T (or MT as originally described), can be computed based on A, the distance to the center of the target from the starting position, and W, the target width. The empirically determined constants a and b are computed using a regression analysis on the movement time data.

FIGURE 6.8 User participants sharing photos with gestures in SharePic.

Refinements to this formulation to ensure a more stable model for the Fitts law have also been proposed.

$$T = a + b\log_2\left(\frac{2A}{W}\right) \tag{6.1}$$

As a result, small targets far away from where one's hands currently are will be difficult to acquire rapidly. Instead, the research, design, and development of an SUI can be seen as an opportunity to move beyond the classical desktop, keyboard, and mouse setting, and into the realization of more natural interaction styles *as appropriate*. The current generation of SUIs built into LCD displays or form factored into coffee tables cannot be considered a "basic object" in Weiser's vision. However, display technologies are now ubiquitous and if SUI interaction styles can be woven into the environments and activities of our everyday lives, then they will have achieved invisibility in action.

Basic SUIs have been commonplace for more than 20 years in the form of interactive kiosks, ATMs, and point-of-sale systems with a touch-sensitive surface overlaid with the display. These systems rely on touch-screen technology with simple button style interfaces. The basic technologies in many products such as the Nintendo DS with its resistive touchscreen technology were established in hardware research decades ago. The recent heightened interest in SUI products such as the iPhone generally stems from the low cost of production and the interaction styles they can afford beyond the desktop paradigm.

Due to their low cost and form factor, PDAs with rendered SUIs have formed components of many of the published ubicomp systems in the literature for the past 15 years, for example, the use of a PDA as memory aid for elders (Szymkowiak et al., 2005). Siren is a context-aware ubicomp system for firefighting (Jiang et al., 2004). Each firefighter carries a WiFi PDA on which the Siren messaging application is running. Each PDA has a Berkeley smart dust mote attached that is used as both a local sensing device and a communication facility to gather data from other motes embedded in the environment. In the Invisible Train system, an AR interface (SUI) is displayed on the screen of the PDA (Wagner et al., 2005). The attached camera allows the PDA to determine its relative 3-D location and rotation with respect to printed fiducial makers in the environment. The tracking and rendering overlay provides an SUI to a 3-D information space embedded within physical reality. In effect, the SUI acts as a “magic lens” onto the world. Interface elements that appear overlaid on the world such as the train speed button or moving track switches can all be operated with a tap on the rendered item on screen. Here, the small PDA acts as both input with stylus, touch, or gesture, and visual output to effect an interactive SUI.

Embedded, personal, or public displays provide opportunities for the development of an SUI. Smart phones can be used in ubicomp for taking and displaying context-aware notes tied to visual codes, photos, dictation, and text (Konomi and Roussos, 2007). It is important to note that coupling of input with output sometimes limits the tasks for which an SUI can be used. Consider the Audi Multi Media Interface (MMI), which provides controls for the company’s high-end in-car navigation, radio, and car systems. The ergonomically designed remote control panel affords input while the two display outputs are in the driver’s primary field of view. These displays use well-considered typography, shapes, shading, and opacity to help clearly present the information to the driver. However, this decoupling of input from output means the MMI is not a form of SUI unlike the more ubiquitous TomTom or Garmin navigation systems seen in-car.

The PDA has represented the first truly small, low-cost SUI platform that researchers experimented with. The interfaces developed have drawn heavily from GUI and mobile computing design and interaction styles. The availability of GUI toolkits has further rendered many of the interfaces on

PDAs displays for the "small screen" and the stylus or finger as a surrogate for the mouse. Character input, however, has moved beyond the GUI style with input languages such as Graffiti and Unistrokes (Castelluci and MacKenzie, 2008). PDAs typically support single-point interaction and the inputs they can accept in terms of tap, double tap, and gestures reflect this. Tablet PCs, although classified as SUIs, have featured less prominently in ubicomp research due to their size, form factor, and supported styles of interaction. Weiser (1999) developed inch-, foot-, and yard-scale displays called "tabs, pads, and boards." Tabs are akin to iPhones and PDAs, pads to the Tablet PC, and boards to larger SUIs.

6.3.2.1 Large SUIs

Larger interfaces that can act as an SUI have been researched and developed for more than 20 years. Although they can be classified in a number of ways, here we consider them from the standpoint of front-projected, rear-projected, and self-illuminated. For front-projected displays, the Digital Desk (Wellner, 1993) is a seminal ubicomp example incorporating an SUI. Both computer vision and audio sensing are used, and physical artifacts placed on the desk (under the gaze of a camera) act as interface elements. An exemplar application of a calculator has a paper sheet with printed buttons. Pressing these buttons is registered by the vision system and the current total is projected into the total square. Urp, described in Section 6.3.1, is a form of SUI relying on tangible objects to interact with it.

Various devices and applications rely on a single mouse input to operate. Single-touch interfaces can readily adapt to emulate single mouse input once issues of finger size and double tapping are considered. However, people typically have four fingers and one thumb on each of two hands. As a result, multitouch and multihanded interaction has emerged as an area of active research, and advances in both hardware and software solutions are supporting multitouch gestures, two-handed input, and multiperson input.

The DiamondTouch from MERL allows multiple, simultaneous users to interact with it as touch location information is determined independently for each user [16]. The DiamondTouch operates by transmitting a unique electrical signal to an array of rows and columns embedded in the table surface as shown in Figure 6.9. Each user is connected to a pad (standing, sitting, or touching) so that when they touch the surface with any part of their body, signals are capacitively coupled from the particular rows and columns touched. The DiamondTouch supports multipoint interaction and multiuser interaction, and is debris-tolerant and durable. SharePic is

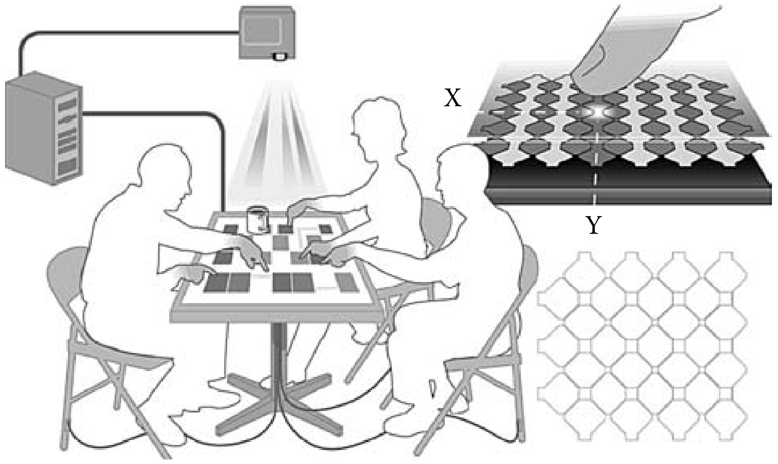

FIGURE 6.9 MERL Diamondtouch Multiuser input schematic. (Courtesy of Circle Twelve.)

a novel gesture-driven photo-sharing framework and application built on top of the DiamondTouch (Apted et al., 2006).

The Sony SmartSkin consists of a mesh of transmitter/receiver electrodes (e.g., copper wires) on the surface that forms a sensing layer (Rekimoto, 2007). The mesh can have a variable density, and it supports mouse emulation, shape-based manipulation, gestures, and capacitance tags on the bottom of tangible objects. When using capacitive sensing, both hand and finger interaction is supported along with physical dials and controls. AudioPad is a musical performance system using tangible music controls on a project surface (Patten et al., 2002). TANGerINE is a tangible tabletop environment, incorporating physical objects with embedded wireless sensors (Baraldi et al., 2007). For example, the Smart Micrel Cube (SMCube) is a wooden cube case with a matrix of IR emitter LEDs on each face. It embeds a WiMoCA node with a triaxial accelerometer and Bluetooth. The IR LEDs allow for computer vision methods to detect their position using simple noise removal, background subtraction, thresholding, and connected components analysis. Front-projected SUIs suffer from the problems of a person's body or hand occluding the display or blocking the line of sight for a camera system. In addition, the space required for the projector mounting can make this form of SUI difficult to manage. Short throw projectors like those used in commercial systems such as the latest SmartBoard go some way to overcoming some of these problems.

Rear-projected SUIs have made a large impression on the popular imagination in the past 5 years. They can be seen in nightly newscasts, during the Super Bowl, in the media, and new products from large multinational corporations such as Microsoft. The exact history of this part of the SUI field is both recent and deeply immersed in "patents pending." As such, we will review four representative systems with regard to the research and design methods behind them and interactions they support. The TouchLight relies on a rear-projected display onto a semitransparent acrylic plastic plane fitted with DNP HoloScreen material (Wilson, 2004). A touch image is computed on the plane situated between the user and a pair of cameras. An IR pass filter is applied to each camera and an IR cut filter to the projector if required. An IR illuminant is placed behind the display to illuminate the surface evenly in IR light. This configuration separates the projected light into the visible spectrum and the sensed light into the IR spectrum. Methods including depth mapping, sobel edge filters, mouse emulation algorithms, connected components algorithms, and template matching are used in TouchLight's computational support. The reacTable operates in a similar manner but it relies on a horizontal orientation and TUI physical objects with fiducial markers (Jordà et al., 2005). An example SUI is a collaborative electronic music instrument with the tabletop tangible multitouch interface. The Micosoft Surface operates in a similar manner to TouchLight, albeit with significant improvements in aspects such as industrial design, form factor, and software support. The IR lighting methods used here are typically referred to as *diffuse illumination*.

Fourier transform infrared (FTIR)-based displays such the multitouch work of Jeff Han (2005) relies on the total internal reaction (TIR) of light being frustrated. The frustration comes about when a object (e.g., finger) makes contact with the surface. The TIR is the same principal used in optical waveguides such as optical fibers. As with direct illumination, in FTIR IR illumination is used as the optical carrier. Here, a sheet of acrylic acts as a waveguide and is edge-lit by an array of high-power IR LEDs. IR light is emitted toward a pair of cameras situated behind the projected surface.

Self-projected systems are typically developed around LCD or plasma screen technologies. Given the space requirements for both front- and rear-projected SUIs, this class of device is drawing on both fundamental touchscreen technologies and the realization that alternate form factors are required. The digital vision touch relies on small cameras embedded in a device around the rim of the display. When an object enters the field of view, the angle within the field of view of the camera is calculated. Multiple cameras allow for triangulation and hence the location of the object (e.g., finger) onscreen to be determined. This is an area in its relative infancy as detecting multipoints as in FTIR or objects (e.g., the shape of a phone) as in TouchLight remains a significant research challenge for systems that do not have a clear view of the touch image from the back, or the front.

Regardless of the approach taken, the coupling of input with output into an SUI can cause a number of usability issues. These issues must be considered when developing an SUI component as part of a larger UUI:

1. The user's stylus, fingers, and hands may partially occlude the interface.
2. Interface elements may be difficult to select due to size of stylus, finger, or hand.
3. Users may suffer fatigue due to the range of human motion required.
4. The scale of the display may not be suitable for all aspects of the task at hand.
5. The screen surface can be damaged or dirty.
6. There is a lack of tactile feedback from passive screen surfaces.
7. Calibration between the display (projector or LCD) and the sensing elements can become misaligned.

Each of these problems can be addressed with careful design. For example, research is ongoing into the coupling of large displays with small personal displays (Terrenghi et al., 2009) to overcome the limitations of each type of SUI device. Other problems can be addressed with new technological advances such as haptic output to overcome the lack of tactile feedback on small SUIs. For example, Sony has developed tactile touchscreen feedback based on a very small actuator that bends the screen. The effect provides a haptic feedback along with a visual one to give the user the sense of buttons that actually click (Poupyrev and Maruyama, 2003).

Front, rear, or self-illuminated displays are becoming ubiquitous in our environment from ATMs to advertisements. Once coupled with support for HCI, they provide a key building block of the realization of always-on, always-available UUIs for ubicomp systems.

6.3.3 Ambient User Interfaces

SUIs require our engagement and involvement to operate correctly. Consider, however, a calm technology that can move easily from the periphery of our attention, to the center, and back again (Weiser, 1999). Ambient information displays or outputs are intended to be "ignorable" or "glanceable," allowing users to perceive the information presented in the periphery of their attention, but also to be bring this information (e.g., social reminders) into focus as required (Shannon et al., 2009). Although the periphery of our attention is a nebulous concept, it is grounded in the notion of our peripheral vision. Extensive research in vision science has demonstrated our abilities to recognize well-known structures and forms, to identify similar forms and particularly movements from outside our foveal line of sight. In practice, an ambient display, sound, movement, or even smell can convey background or context outputs from a ubicomp system (Heiner et al., 1999). Figure 6.10 demonstrates constant feedback through a tactile ambient display to express a mobile phone's state. Qualitative user studies demonstrated the ability to selectively "tune in and out" (Hemmert, 2009) status information was welcomed, even if the rudimentary form of actuation was not. In the future, a low-power heating element or other peripherally ignorable form of actuation could provide constant ambient feedback one could selectively attune to.

An ambient output (display or other) does not constitute a full user interface because it does not incorporate input. Many of the inputs we have described here would explicitly negate the glanceable nature of an AUI. As such, an AUI is a class of user interface where the output elements reside in the periphery of a user's awareness, moving to the center of attention only

FIGURE 6.10 Ambient information provided by a tactile ambient display in a mobile phone. (Courtesy of Fabian Hemmert.)

when appropriate and desirable, and the inputs come from nonintrusive sensing or inference from other actions. Various dimensions of interest including a user's information capacity, suitable notification level, display representational fidelity, and aesthetic emphasis have been explored to understand AUIs and their limits (Pousman and Stasko, 2006). Additional studies have demonstrated a person's inability to recall the structure of an ambient information display, whereas the essential details and information conveyed can be recalled (Rashid and Quigley, 2009).

Fully realized AUIs as defined are not yet commonplace in our daily lives. However, ambient displays (e.g., digital advertisements and signage (Rashid and Quigley, 2009)) are commercially available and the subject of active research and development as part of larger information systems. Aspects of, or objects in, the physical environment become representations for digital information rendered as subtle changes in form, movement, sound, color, or light that are peripherally noticeable. Such displays can provide constant/situated awareness of information. Again, we can discuss these with reference to Weiser's original vision of "computation in basic objects, environments and the activities of our everyday lives in such a way that no one will notice its presence" (1999). Weiser himself discussed the Jeremjenko Dangling String called Live Wire showing network traffic in Xerox PARC as one of the earliest cited examples of an ambient display. Network activity caused the string to twitch, yielding a "peripherally noticeable indication of traffic."

The power-aware cord, as shown in Figure 6.11, is designed to visualize the energy flowing through it rather than hiding it (Gustaffson and

FIGURE 6.11 Power-aware cord, a power-strip designed to actively visualize the energy in use. (Photograph by Magnus Gyllenswärd. Accessed from http://tii.se/static/press.htm)

Gyllenswärd, 2005). This AUI aims to provide increased consumer awareness of their energy consumption. Switching off/on or unplugging devices acts as an input to the system and the display consists of glowing patterns produced by electroluminescent wires molded into the transparent electrical cord. This is a very basic object, present in our everyday environment, and with people ever more concerned with energy conservation, this has become a natural activity of our daily life, that is, consideration of energy usage. By relying on patterns of movement, this AUI can be attuned to in our periphery via peripheral vision. Longitudinal research studies of both the effectiveness and the disturbance of our visual field are required for this AUI. Would a moving pattern ever truly blend its way into our environments so as we might not notice it?

The information percolator (Heiner et al., 1999) is an "aesthetically pleasing decorative object." Here, the display is created by the control and release of air bubbles that rise up a series of tubes. The effect appears as a set of pixels, which can be used to convey a message that scrolls up the display. Sample applications include a personalized clock, activity awareness, a poetry display, and an interactive "bubble painting" application. The first three can be classed as AUIs as they require no explicit user input to operate and can be truly glanceable. The bubble painting treats the percolator as a large display with gestural input, thereby rendering it an SUI in this case. This blending of multiple interface classes will be typical and expected of future UUI developments. The percolator is not a basic object, but as a piece of installation art it can go unnoticed and thus remain glanceable.

By contrast, described as *information decoration*, the Datafountain displays relative currency rates using three water fountains side by side as shown in Figure 6.12. The relative heights of each fountain change with

FIGURE 6.12 Datafountain displaying relative currency rates.

respect to the currency changes (van Mensvoort, 2005). Here, the fountain is a basic object one would expect to find in day-to-day environments. However, unlike the information percolator, it remains only an output with inputs provided by a web service call. The InfoCanvas paints an appealing and meaningful representation of user-identified information on a virtual canvas. This ambient display allows people to monitor information in a peripheral manner (Miller and Stasko, 2001).

Finally, the suite of commercial systems for Ambient Devices including the Orb and Umbrella as shown in Figure 6.13 along with the Energy Joule represents the best examples of ambient displays, which have gained widespread deployment. The Ambient Orb is a frosted ball that glows in different colors to allow user to peripherally monitor changes of the status of everyday information. The color range and movement can be tailored via a website to the user's preference to show information including real-time stock market trends, pollen forecasts, weather, wind speed, and traffic congestion. The Ambient "Energy Joule" glows red, yellow, or green to indicate current cost of energy. Further details can be determined upon closer inspection with a left status bar detailing the cost, and right bar showing your current energy consumption. With the Ambient Umbrella, if rain is forecast, then the handle of the umbrella glows so you will not forget to take it. The Umbrella, as shown in Figure 6.13, is an everyday object, and deciding to take it or not each day is certainly an activity for those who live in rainy places. It remains an output only, but can move to the center of our focus if we peripherally notice the glowing handle. The Orb and Joule represent special purpose devices, not everyday objects that exhibit Ambient display features. The Orb may function as a component of a larger AUI or UUI as developed or designed in the future. With the

FIGURE 6.13 Ambient Orb and Ambient Umbrella. (Courtesy of Ambient Devices.)

Joule AUI turning on or off, an appliance can act as an input, whereas the color on the Joule provides ambient feedback that can remain glanceable.

Further AUI examples include the context-aware Visual Calendar (Neely et al., 2007), AuraOrbs and ambient trolley (Stasko et al., 2007), Information Art (Stasko et al., 2004), CareNet (Consolvo et al., 2004), BusMobile, and Daylight Display (Mankoff et al., 2003).

An AUI may exist in a device or environment for long periods without a conscious user action, yet the user is getting information from it, and by his/her actions or colocation, is providing it inputs. As a class of user interface, the ambient one conforms closest to Weiser's vision of calm technology. The design and evaluation of an AUI is particularly challenging from many standpoints including engaging users, conveying the utility, determining the effectiveness, efficiency, and usability (Mankoff et al., 2003), and collecting longitudinal efficacy data. In practice, an AUI may transition from implicit use to being an explicit SUI or from support for interaction with one to multiple users (Vogel and Balakrishnan, 2004). Clearly, there are limited types of information where the use of a dedicated single purpose AUI is useful. Hence, AUIs will form only part of a larger UUI for the system as a whole.

6.4 INPUT TECHNOLOGIES

A UUI relies on a broader range of inputs and outputs from the system than the classical GUI, TUI, or even an SUI. Examples of these inputs include physiological measurements, location, identity, video, audio, gesture, and touch. In addition, environmental sensors, personal/embedded sensors, data mining, historical data, inference, and preferences can all act as inputs to a system. Examples of the outputs that can be used include haptics, ambient displays, environmental updates, actuators, automated actions and personalized behaviors, and multiple audio/video channels. Such outputs are all reliant on our senses including sight, taste, smell, touch, hearing, and balance. As we have seen from our discussion on interface classes, some user actions will be interpreted via the UUI as an input to the system without the user being explicitly aware of it. Likewise, the UUI can provide outputs that are only intended for the periphery of the user's attention. Of course, many inputs and outputs will require explicit user action to form the input and to process the output provided. As we have already reviewed a wide range of output technologies, we focus on three categories of input namely sensor input, gesture input, and speech input for the purposes of concluding this chapter.

6.4.1 Sensor Input

A sensor is a device that can measure a physical property from the environment. Sensors can reside in the environment or on the body. Environmental sensors are becoming more widely deployed as components of systems to monitor traffic, air quality, water quality, and light pollution. More locally, sensors can be found embedded in doorways, security systems, weight monitors, and health systems. When sensors are collecting measurements about the person they are attached to, they are called physiological sensors (e.g., heart rate or galvanic skin response measurements); when not used as such, they are called mobile sensors (e.g., light levels, pollution monitoring, temperature). Computational measurements such as network traffic or memory usage can be determined by software sensors. The measurements from all these sensors are aspects of the context data that a ubicomp system can use to function correctly. By a user's regular actions, they can affect the environment in which the sensors operate (e.g., opening a window, turning on a stove, streaming a video). These actions can hence be viewed as implicit user inputs to a ubicomp system. For further details on sensors and context data, see Chapter 8 on context for ubicomp in this book.

The SensVest is a physiological monitoring system built into a wearable form factor. It measures aspects of human physical performance such as heart rate, temperature, and movement (Knight et al., 2005). The ring sensor, which measures blood oxygen saturation via pulse oximetry, is a component of a 24-hour patient monitoring system (Rhee et al., 1998). In both cases, these measurements form inputs to a larger system. Some commercial sensor systems rely on inertial technology such as the Moven motion capture suit, which can capture full body human motion.

As with many systems that rely on the collection of sensor inputs, the iHospital system is supported by a location and context-awareness infrastructure (Hansen et al., 2006). PCs can detect Bluetooth-enabled badges that the staff are wearing when they come within a 10 meter range. No explicit input is required; instead, the location data are sensed from the wireless environment, interpreted, and acted upon with various feedback in the form of reminders and surgery status details, on personal and public displays. Body-worn sensors or sensors embedded in devices can also collect various environmental measurements. Infrastructures can also be adapted to subtly alter the environmental conditions that allows for novel sensing systems to be developed. For example, an economical transmitter circuit can be realized by adding few components to a commercial electronic ballast circuit for fluorescent lamps

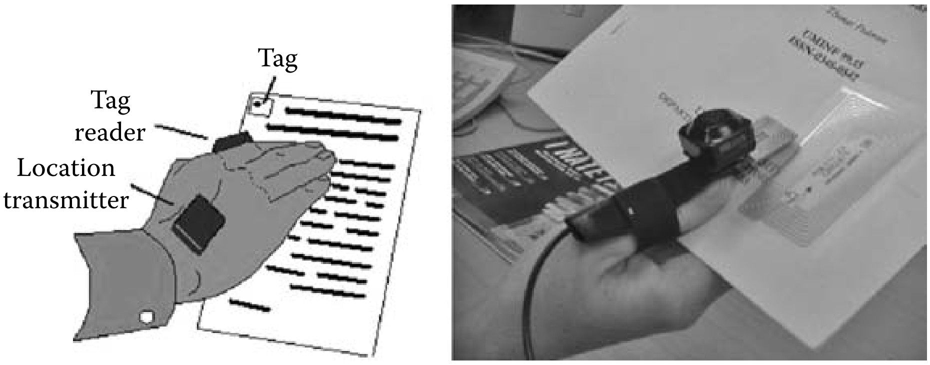

FIGURE 6.14 Magic Touch: use of RFID reader for tagged object identification and tracking. (Courtesy of Thomas Pederson.)

(Cheok and Li, 2008). A photo-receiver sensor can decode the pulse-frequency modulation that can allow this sensor to be localized. Once the location of the sensor, or more importantly the device or person it is attached to, can be determined, this can then provide an input to the system.

Active RFID systems can be sensed at a distance, whereas passive RFID requires closer contact and typically explicit user action for the sensor system to read the tag as shown in Figure 6.14. Examples of such large-scale sensor deployments include the Oyster card ticketing system used at the London Underground in the United Kingdom and retail applications deployed in Japan as examples, and other large-scale ubiquitous sensor deployments (Konomi and Roussos, 2007). For further details on such deployment, see Chapter 7.

6.4.2 Gesture Input

A gesture is the movement of a part of the body to express an idea of meaning. Typical gestures such as pointing, waving, or nodding are formed with the hand or the head as appropriate. For example, in a sign language a specific configuration and movement of the hands is considered a gesture. Figure 6.15 shows another common gesture, namely, a handshake, which can be detected and matched and used to infer social relationships (see Haddock et al., 2009). Both basic gestures such as pointing and complex gestures in sign language rely heavily on the appropriate cultural, geographical, or linguistic frame of reference for their interpretation. An innocuous gesture in one country can easily be interpreted as an insult in another. As such, gesture recognition is the process of interpreting human gestures using various inputs and computational processing. Almost any type of input device or system can be used to collect data

FIGURE 6.15 Matching accelerometer data for sensing handshakes for social network development.

on a user gesture. The difference between them is in the determination of how natural this movement actually is. Systems incorporating gesture recognition often suffer from the problems of feedback and visibility as described in Section 6.1.2—that is, how do I perform the gesture and how do I know what I just did was interpreted or not? Magic Touch, shown in Figure 6.16, is a novel gesture-driven object tracking and identification system (Pederson, 2001). Here, pages are tagged (either by the user or from the printer) and, as a user touches any page, the system develops a model that replicates the real-world organization of books, papers, and pages. Here, a gesture is simply the act of picking up and placing down a piece of paper, that is, very natural.

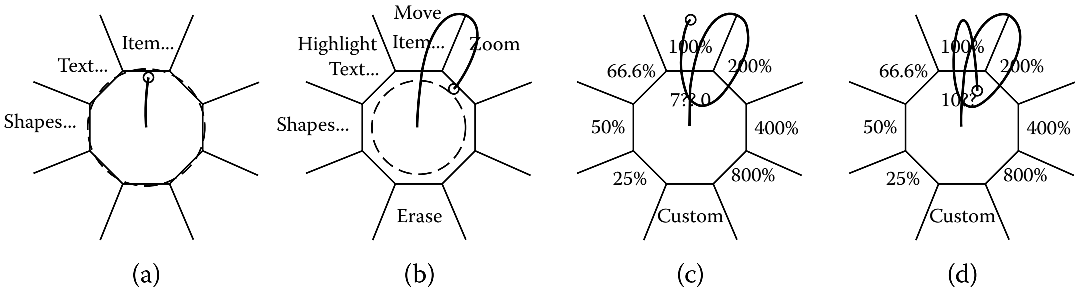

FIGURE 6.16 Flowmenu, single pen gesture to select option within a hierarchy.

Pen gestures using styli on PDAs and Tablets are the most ubiquitous example of gestural input to a computer system. Palm's Graffiti and Graffiti 2 are in widespread use and allow for rapid character and state changes (caps lock, shift) using simple gestures. However, they represent only a very small number of body movements. Pen input helped give rise to research on alternate forms of menu selection using gestural input, which do provide feedback and visibility. Circular pie, marking, and flow menus (Guimbretiére and Winograd, 2000) give each menu item a large target slice that is laid out radially and in different directions. Once you know the directions, you can quickly and reliably move ahead without looking. The user need not wait for the menus to render, so a single pen stroke can be a selection. Flow menus have a radial menu and overlaid submenus with eight octants. Experiments with it have shown that users can naturally learn gestures that require several strokes in one fluid gesture (Guimbretiére and Winograd, 2000). As single point inputs, all pen gestures can also be used with mouse input, although often with noticeable performance degradation.

Mouse gestures are commonly available for Web browsers such as Opera or FireFox via plugins. An example mouse gesture in Opera is for the user to hold down the right mouse button, move the mouse left, and release the right mouse button. This gesture encapsulates a series of movements (from current position to the back button) and a mouse click. Alternatively, a cheat sheet can be provided to a user on the make up of the pen and mouse gestures. In this case, the gesture can be considered a *command language* that must be memorized for efficient use. A specific problem with mouse gestures is the lack of standardization across applications, hence limiting the beneficial learning effect enjoyed by having visual widgets in a GUI.

The use of video input from a single or stereo camera array, relying on visible or near-visible light (e.g., IR) is a common method for collecting gestural input. This is a large field of research and development that is not limited to ubicomp scenarios. Classical UI interfaces augmented with video-based gestural input for gaming and sign language recognition are excellent examples of research in this field. The gestural input on SUIs such as the Microsoft Surface relies on the capture of IR light and subsequent processing with computer algorithms. Established algorithms and methods for processing images and video for gesture recognition include hidden Markov models (HMM) (Wilson and Bobick, 1999) and neural networks (Murakami and Taguchi, 1991). For further details, we refer the reader to the review by Pavlovic et al. (1997) on the visual interpretation of hand gestures. In ubicomp research, the FingerMouse (de la Hamette and Tröster 2008) relies on a button-sized camera that can be worn on the body, capturing the user's hand and supporting real-time segmentation and processing. Although not an everyday object, it is not a stretch to imagine such inputs factored into jewelry and other items commonly worn on the body.

Physical artifacts can be instrumented with sensors to provide accelerometer data. Body movement with the artifact can form an input that can be interpreted as a gesture such as shake, bump, tap, bounce, and rotate. TiltType relies on the combination of tilting actions with button presses to input characters to the device (Fishkin et al., 2002).

The Display Cube provides gesture recognition and output through six displays mounted at the sides of the cube (Kranz et al., 2005). A form of TUI, this is a playful learning interface for children. Simple simultaneous button presses affords SyncTap the ability to seamlessly authenticate two devices (Rekimoto, 2004). In contrast, simply holding two devices together and shaking them provides a method for automatic authentication in Shake Well Before Use (Mayrhofer and Gellersen, 2007). Smart-Its Friends are small embedded devices that become connected when a user holds them together and shakes them (Holmquist et al., 2001). The Wiimote is a handheld wireless game controller that can sense motion. Although this is a special purpose device and not an everyday object, its widespread adoption and ease of use can make the combination of game input and output invisible in action. The number of Wiimote devices that have ended up embedded in display screens is a testament to the fact that people forget it is a physical object that can damage other things and people!

6.4.3 Speech Input

Speech is the ability to produce articulate sounds to express an idea or meaning, and is typically realized through a spoken language. Speech recognition is the process of interpreting human speech using a variety of audio inputs and computational processing. As with video-based input processing, speech recognition is represented by a large body of research, development, and commercial deployment. The most widespread deployment of speech recognition can be found in phone call routing systems relying on a limited context sensitive vocabulary of words such as yes/no, numbers, and city names. The next most common example is in commercial dictation systems such as Dragon NaturallySpeaking. These rely on training periods with the user and a relatively noise-free environment in which to achieve high levels of speech recognition. Many high-end cars

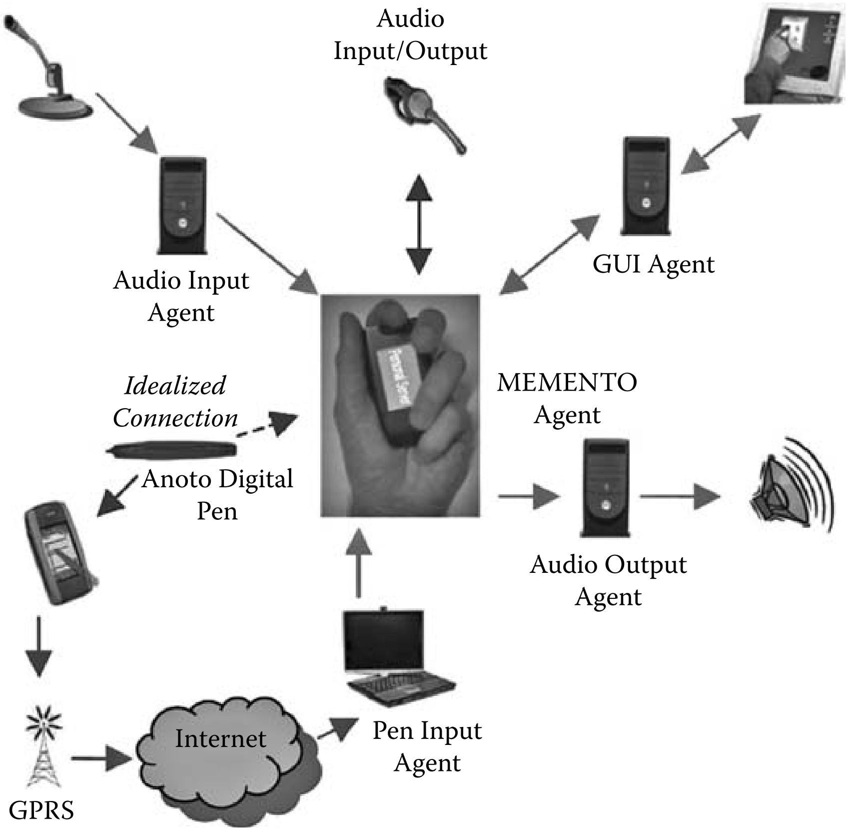

FIGURE 6.17 The MEMENTO speech agent in a user's personal area network.

use commercial systems such as IBM's ViaVoice for embedded speech recognition in automotive telematics and other basic controls.

Speech represents a popular view of how humans will interact with computers as evidenced in literature and film. From natural interaction with next-generation robotic systems to HAL seen 40 years ago in Arthur C. Clarke's *2001: A Space Odyssey*, speech represents one view of how people can interact naturally with future ubicomp systems. Speech recognition and natural speech output represent the backbone of natural language and anthropomorphic interfaces. In addition, speech is often used as a secondary input modality to multimodal systems. For example, Memento (West et al., 2007) uses a combination of speech and pen input for the creation of a digital-physical scrapbook as shown in Figure 6.17. Speech is recognized by Sphinx components (Huang et al., 1993), and gestures by the Microsoft handwriting and gesture recognition libraries. The fused input forms the basis for action by Memento, which in this case is a digital-physical scrapbook for memory sharing.

Speech forms a central mode of input in infrastructure-rich visions of ubicomp as in Project Aura (Garlan et al., 2002). Here, the environment consists of many agents with which one can communicate using natural language. Aura represents a class of Intelligent Environment research popular in ubicomp relying on a rich range of seamlessly coupled inputs and outputs. Beyond such next-generation environments, today there are many work environments where speech can provide a more natural interface. These are environments where the classic GUI with keyboard and mouse or even next-generation SUI or TUI cannot provide the required computational support for the task at hand in a natural manner. Environments such as surgeries, dangerous work environments, and driving all represent environments where UUIs with speech as input have been researched and developed. Ubicomp scenarios incorporating in-car elements are a natural environment for the exploration of speech input in a UUI (Graham and Carter, 2000). SEAR (speech-enabled augmented reality) offers a technician a context-sensitive speech dialogue concerning industrial equipment in the technician's immediate vicinity (Goose et al., 2003).

Speech recognition systems using statistically based algorithms such as HMMs rely on both an acoustic and language model. For a UUI researcher and developer, there exist a number of toolkits that can speed up the rapid prototyping and iterative design process with speech. VoxForge can be used to create acoustic models for speech recognition engines. Designed to collect transcribed speech, VoxForge aims to create multiuser acoustic models that can be used without training. The Julius large vocabulary continuous speech recognition engine is by the Continuous Speech Recognition Consortium in Japan (Kawahara et al., 2004). Julius can be incorporated into a UUI, although it requires the computation power found in modern desktop PCs to operate. The CMU Sphinx Group produces a series of Open Source Speech Recognition components (Huang et al., 1993). These include decoders, acoustic model training, and language model training. Other toolkits include the Internet-Accessible Speech Recognition Technology C++ libraries and the HMM Toolkit (Young et al., 2002). Other groups are attempting to realize the standardization seen in GUIs across "Universal Speech Interfaces." The learning from interacting with one such USI can be transferred to future experiences with others (Rosenfeld et al., 2001).

Although speech input is key, audio can be used to provide speech output of verbal instructions in a UUI (Bradley and Dunlop, 2005). ONTRACK provides navigation cues to a user navigating an environment by adapting the audio he or she is listening to. This is achieved by modifying the spatial balance and volume to help guide the user to his or her destination (Jones et al., 2007). NAVITIME, used by more than 1.82 million people in Japan, provides map-based routes, text routes, and provides turn-by-turn voice guidance for pedestrians on handheld mobile phones and PDAs (Arikawa et al., 2007). Gesture and speech interaction can be provided with patient information systems (O'Neill et al., 2006). Bluetooth headsets can be used to provide direct audio and speech outputs to the user, as in the feedback provided in Memento, where the headset operates for both speech input and output in a personalized manner to the user (West et al., 2004).

6.5 INTERFACE USABILITY METRICS

Usability is a quality attribute that assesses how easy user interfaces are to use. The word "usability" also refers to methods for improving ease-of-use during the design process. Usability is defined by five quality components: learnability, efficiency, memorability, errors, and satisfaction.

For a UUI, we can describe a set of interface usability metrics, as shown in Table 6.2.

TABLE 6.2 Seven Key UUI Usability Metrics

Metric	Meaning
Conciseness	A few simple actions in a brief time can achieve a task. This can be measured by time (attention or gaze), keystrokes, gestures, and taps.
Expressiveness	Does a combination of actions not anticipated give consistent results?
Ease	How much does a user need to learn or recall just to start using the interface?
Transparency	How much does a user need to remember about the state of his or her problem while using the interface telephone speech interface versus a GUI?
Discoverability	Can the user easily understand and form a mental model of the interface functionality?
Invisibility	How much does the interface make itself know when it could have inferred, deduced, or waited for the data required?
Programmability	Can the application, device, or service be used in repetitive tasks or can it become a component in a larger system?

6.6 CONCLUSIONS

This chapter has provided an overview of the classes of interface both new and old that may be suitable to realize interfaces of applications in environments imbibed with computation, input, and output. This chapter started by setting the context for what a UUI is and hence detailing 10 rules for UUI design (Table 6.1). We have shown contrasting approaches to interaction design and how these have impacted on each other and the relative merits of each approach in Section 6.2. Section 6.3 provided a rich description of different classes of user interface. Three novel forms of input modality are described in Section 6.4. This chapter concludes with interface usability metrics described in Table 6.2 for UUIs in Section 6.5 and how these may be applied to future interface design.

Although this chapter offers a snapshot of the state of the art in UUI development, it is best viewed as an entry point to the research papers cited here. Considerable research and development has been undertaken in both ubicomp and the research areas that underpin the subject. Developing a UUI as a prototype or proof of concept to demonstrate an idea or run some small-scale user trials is fundamentally different to the realization of a UUI and system for use in the wild. In the wild, an interface is expected to have all the properties shown in Table 6.1 and more. Anything less constitutes risk that people will not purchase, adopt, or use your system. General usability metrics and our ubicomp interface usability metrics are described in Table 6.2. When and where they are to be applied should be considered and reflected upon time and again in the research and development process.

The ultimate goal for ubicomp is to have interfaces and hence systems that seamlessly support the actions of their users. In the future, enhanced computational artifacts, environments, and full ubicomp systems will become so commonplace in restaurants, colleges, workplaces, and homes that no one will notice their presence. Our job as researchers, designers, developers, and ultimately users is to constantly question this vision to ensure we end up with calm and blissful digital-physical experiences.

ACKNOWLEDGMENTS

The authors thank Ross Shannon, John Krumm, and Marc Langheinrich for helpful comments and discussions on this text.

REFERENCES

Gregory D. Abowd and Elizabeth D. Mynatt. Charting past, present, and future research in ubiquitous computing. *ACM Transactions on Computer-Human Interaction* 7(1):29–58, 2000.

Christopher Alexander, Sara Ishikawa, and Murray Silverstein. *A Pattern Language: Towns, Buildings, Construction (Center for Environmental Structure Series).* Oxford University Press, Oxford, 1977.

Trent Apted, Judy Kay, and Aaron Quigley. Tabletop sharing of digital photographs for the elderly. In *CHI '06: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 781–790, New York, NY, USA, April 2006. ACM Press.

Masatoshi Arikawa, Shin'ichi Konomi, and Keisuke Ohnishi. Navitime: Supporting pedestrian navigation in the real world. *IEEE Pervasive Computing* 6(3): 21–29, 2007.

Rafael Ballagas, Meredith Ringel, Maureen Stone, and Jan Borchers. iStuff: A physical user interface toolkit for ubiquitous computing environments. In *CHI '03: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 537–544, New York, NY, USA, 2003. ACM.

Stefano Baraldi, Alberto Del Bimbo, Lea Landucci, Nicola Torpei, Omar Cafini, Elisabetta Farella, Augusto Pieracci, and Luca Benini. Introducing tangerine: A tangible interactive natural environment. In *MULTIMEDIA '07: Proceedings of the 15th International Conference on Multimedia*, pp. 831–834, New York, NY, USA, 2007. ACM Press.

Nicholas Bradley and Mark Dunlop. An experimental investigation into wayfinding directions for visually impaired people. *Personal and Ubiquitous Computing* 9(6):395–403, 2005.

Steven J. Castellucci and I. Scott MacKenzie. Graffiti vs. unistrokes: An empirical comparison. In *CHI '08: Proceeding of the Twenty-Sixth Annual SIGCHI Conference on Human Factors in Computing Systems*, pp. 305–308, New York, NY, USA, 2008. ACM Press.

Adrian Cheok and Yue Li. Ubiquitous interaction with positioning and navigation using a novel light sensor-based information transmission system. *Personal and Ubiquitous Computing* 12(6):445–458, 2008.

Adrian David Cheok, KokHwee Goh, Wei Liu, Farzam Farbiz, SiewWan Fong, SzeLee Teo, Yu Li, and Xubo Yang. Human pacman: A mobile, wide-area entertainment system based on physical, social, and ubiquitous computing. *Personal and Ubiquitous Computing* 8(2):71–81, 2004.

Sunny Consolvo, Larry Arnstein, and Robert B. Franza. User study techniques in the design and evaluation of a ubicomp environment. *UbiComp 2002: Ubiquitous Computing*, pp. 281–290, 2002.

Sunny Consolvo, Peter Roessler, and Brett E. Shelton. The CareNet display: Lessons learned from an in home evaluation of an ambient display. *UbiComp 2004: Ubiquitous Computing*, pp. 1–17, 2004.

Joëlle Coutaz, James L. Crowley, Simon Dobson, and David Garlan. Context is key. *Communications of the ACM* 48(3):49–53, 2005.

Patrick de la Hamette and Gerhard Tröster. Architecture and applications of the fingermouse: A smart stereo camera for wearable computing HCI. *Personal and Ubiquitous Computing* 12(2):97–110, 2008.

Paul Dietz and Darren Leigh. Diamondtouch: A multi-user touch technology. In *UIST '01: Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology*, pp. 219–226, New York, NY, USA, 2001. ACM Press.

Alan Dix, Janet E. Finlay, Gregory D. Abowd, and Russell Beale. *Human-Computer Interaction, 3rd Edition*. Prentice Hall, Inc., Upper Saddle River, NJ, USA, 2003.

Kenneth P. Fishkin. A taxonomy for and analysis of tangible interfaces. *Personal and Ubiquitous Computing* 8(5):347–358, 2004.

Kenneth P. Fishkin, Kurt Partridge, and Saurav Chatterjee. Wireless user interface components for personal area networks. *IEEE Pervasive Computing* 1(4):49–55, 2002.

D. Fitton, K. Cheverst, C. Kray, A. Dix, M. Rouncefield, and G. Saslis-Lagoudakis. Rapid prototyping and user-centered design of interactive display-based systems. *Pervasive Computing, IEEE* 4(4):58–66, 2005.

Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns*. Addison-Wesley Publishing, Reading, MA, 1994.

David Garlan, Dan Siewiorek, Asim Smailagic, and Peter Steenkiste. Project aura: Toward distraction-free pervasive computing. *IEEE Pervasive Computing* 1(2):22–31, 2002.

Stuart Goose, Sandra Sudarsky, Xiang Zhang, and Nassir Navab. Speech-enabled augmented reality supporting mobile industrial maintenance. *IEEE Pervasive Computing* 2(1):65–70, 2003.

Robert Graham and Chris Carter. Comparison of speech input and manual control of in-car devices while on the move. *Personal and Ubiquitous Computing* 4(2):155–164, 2000.

Adam Greenfield. *Everyware: The Dawning Age of Ubiquitous Computing*. Peachpit Press, Berkeley, CA, USA, 2006.

François Guimbretiére and Terry Winograd. Flowmenu: Combining command, text, and data entry. In *UIST '00: Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology*, pp. 213–216, New York, NY, USA, 2000. ACM Press.

Anton Gustafsson and Magnus Gyllenswärd. The power-aware cord: Energy awareness through ambient information display. In *CHI '05: CHI '05 Extended Abstracts on Human Factors in Computing Systems*, pp. 1423–1426, New York, NY, USA, 2005. ACM Press.

David Haddock, Aaron Quigley, and Benoit Gaudin. Sensing handshakes of social network development. In *Proceedings of the Artiàcial Intelligence and Cognitive Science 20th Irish International Conference.* AICS, 2009.

Jefferson Y. Han. Low-cost multi-touch sensing through frustrated total internal reflection. In *UIST '05: Proceedings of the 18th Annual ACM symposium on User Interface Software and Technology*, pp. 115–118, New York, NY, USA, 2005. ACM Press.

Thomas Riisgaard Hansen, Jakob E. Bardram, and Mads Soegaard. Moving out of the lab: Deploying pervasive technologies in a hospital. *IEEE Pervasive Computing* 5(3):24–31, 2006.

Jeremy M. Heiner, Scott E. Hudson, and Kenichiro Tanaka. The information percolator: Ambient information display in a decorative object. In *UIST '99: Proceedings of the 12th Annual ACM Symposium on User Interface Software and Technology*, pp. 141–148, New York, NY, USA, 1999. ACM Press.

Fabian Hemmert. Life in the pocket: The ambient life project life-like movements in tactile ambient displays in mobile phones. *International Journal of Ambient Computing and Intelligence* 1(2):13–19, 2009.

Lars Holmquist, Friedemann Mattern, Bernt Schiele, Petteri Alahuhta, Michael Beigl, and Hans W. Gellersen. Smart-Its friends: A technique for users to easily establish connections between smart artefacts. *Ubicomp 2001: Ubiquitous Computing*, pp. 116–122, 2001.

Xuedong Huang, Fileno Alleva, Hsiao-Wuen Hon, Mei-Yuh Hwang, and Ronald Rosenfeld. The SPHINX-II speech recognition system: An overview. *Computer Speech and Language* 7(2):137–148, 1993.

Hiroshi Ishii and Brygg Ullmer. Tangible bits: Towards seamless interfaces between people, bits and atoms. In *CHI '97: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 234–241, New York, NY, USA, 1997. ACM Press.

Anne Jardim. *The First Henry Ford: A Study in Personality and Business.* Leadership. MIT University Press, Cambridge, MA, 1970

Xiaodong Jiang, Nicholas Y. Chen, Jason I. Hong, Kevin Wang, Leila Takayama, and James A. Landay. Siren: Context-aware computing for firefighting. *Pervasive Computing*, pp. 87–105, 2004.

Matt Jones, Steve Jones, Gareth Bradley, Nigel Warren, David Bainbridge, and Geoff Holmes. Ontrack: Dynamically adapting music playback to support navigation. *Personal and Ubiquitous Computing*, 2007.

Sergi Jordà, Martin Kaltenbrunner, Günter Geiger, and Ross Bencina. The reactable*. In *Proceedings of the International Computer Music Conference (ICMC 2005)*, Barcelona, Spain, 2005.

Tatsuya Kawahara, Akinobu Lee, Kazuya Takeda, Katsunobu Itou, and Kiyohiro Shikano. Recent progress of open-source LVCSR Engine Julius and Japanese model repository and Japanese model repository—software of continuous speech recognition consortium. In *INTERSPEECH-2004*, pp. 3069–3072. http://julius.sourceforge.jp/, 2004.

James F. Knight, Anthony Schwirtz, Fotis Psomadelis, Chris Baber, Huw W. Bristow, and Theodoros N. Arvanitis. The design of the SensVest. *Personal and Ubiquitous Computing* 9(1):6–19, 2005.

Shin'ichi Konomi and George Roussos. Ubiquitous computing in the real world: Lessons learnt from large scale RFID deployments. *Personal and Ubiquitous Computing* 11(7):507–521, 2007.

Matthias Kranz, Dominik Schmidt, Paul Holleis, and Albrecht Schmidt. A display cube as tangible user interface. In *Adjunct Proceedings of the Seventh International Conference on Ubiquitous Computing (Demo 22)*, September 2005.

James A. Landay and Gaetano Borriello. Design patterns for ubiquitous computing. *Computer* 36(8):93–95, 2003.

Jennifer Mankoff, Anind K. Dey, Gary Hsieh, Julie Kientz, Scott Lederer, and Morgan Ames. Heuristic evaluation of ambient displays. In *CHI '03: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 169–176, New York, NY, USA, 2003. ACM Press.

Rene Mayrhofer and Hans Gellersen. Shake Well Before Use: Authentication based on accelerometer data. In *Proceedings of 5th International Conference on Pervasive Computing*, pp. 144–161, 2007.

Todd Miller and John Stasko. The InfoCanvas: Information conveyance through personalized, expressive art. In *CHI '01: CHI '01 Extended Abstracts on Human Factors in Computing Systems*, pp. 305–306, New York, NY, USA, 2001. ACM Press.

Florian Mueller, Gunnar Stevens, Alex Thorogood, Shannon O'Brien, and Volker Wulf. Sports over a distance. *Personal and Ubiquitous Computing* 11(8):633–645, 2007.

Kouichi Murakami and Hitomi Taguchi. Gesture recognition using recurrent neural networks. In *CHI '91: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 237–242, New York, NY, USA, 1991. ACM Press.

Steve Neely, Graeme Stevenson, and Paddy Nixon. Assessing the suitability of context information for ambient display. In William R. Hazlewood, Lorcan Coyle, and Sunny Consolvo, Eds., *Proceedings of the 1st Workshop on Ambient Information Systems. Colocated with Pervasive 2007*, Toronto, Canada, volume 254 of CEUR Workshop Proceedings, ISSN 1613–0073, May 2007.

Jakob Nielsen. The Myth of the Genius Designer, http://www.useit.com/alertbox/genius-designers.html, May 2007.

Donald A. Norman. *The Design of Everyday Things*. Basic Books, New York, NY, 2002.

Eamonn O'Neill, Manasawee Kaenampornpan, Vassilis Kostakos, Andrew Warr, and Dawn Woodgate. Can we do without GUIs? Gesture and speech interaction with a patient information system. *Personal and Ubiquitous Computing* 10(5):269–283, 2006.

James Patten, Ben Recht, and Hiroshi Ishii. Audiopad: A tag-based interface for musical performance. In *NIME '02: Proceedings of the 2002 Conference on New Interfaces for Musical Expression*, pp. 1–6, Singapore, Singapore, 2002. National University of Singapore.

Vladimir I. Pavlovic, Rajeev Sharma, and Thomas S. Huang. Visual interpretation of hand gestures for human-computer interaction: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19(7):677–695, 1997.

Thomas Pederson. Magic touch: A simple object location tracking system enabling the development of physical-virtual artefacts in office environments. *Personal and Ubiquitous Computing* 5(1):54–57, 2001.

Ivan Poupyrev and Shigeaki Maruyama. Tactile interfaces for small touch screens. In *UIST '03: Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology*, pp. 217–220, New York, NY, USA, 2003. ACM Press.

Zachary Pousman and John Stasko. A taxonomy of ambient information systems: Four patterns of design. In *AVI '06: Proceedings of the Working Conference on Advanced Visual Interfaces*, pp. 67–74, New York, NY, USA, 2006. ACM Press.

Aaron Quigley, Belinda Ward, Chris Ottrey, Dan Cutting, and Robert Kum-merfeld. Bluestar, a privacy centric location aware system. In *Position Location and Navigation Symposium (PLAN 2004)*, pp. 684–689, April 2004.

Umer Rashid and Aaron Quigley. Ambient displays in academic settings: Avoiding their underutilization. *International Journal of Ambient Computing and Intelligence* 1(2):31–38, 2009.

Leah M. Reeves, Jennifer Lai, James A. Larson, Sharon Oviatt, T. S. Balaji, Stéphanie Buisine, Penny Collings, Phil Cohen, Ben Kraal, Jean-Claude Martin, Michael McTear, T. V. Raman, Kay M. Stanney, Hui Su, and Qian Ying Wang. Guidelines for multimodal user interface design. *Communications of the ACM* 47(1):57–59, 2004.

Jun Rekimoto. Smartskin: An infrastructure for freehand manipulation on interactive surfaces. In *CHI '02: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 113–120, New York, NY, USA, 2002. ACM Press.

Jun Rekimoto. Synctap: Synchronous user operation for spontaneous network connection. *Personal and Ubiquitous Computing* 8(2):126–134, 2004.

Sokwoo Rhee, Boo-Ho Yang, Kuowei Chang, and H. H. Asada. The ring sensor: A new ambulatory wearable sensor for twenty-four hour patient monitoring. *Engineering in Medicine and Biology Society, 1998. Proceedings of the 20th Annual International Conference of the IEEE*, 4:1906–1909, 1998.

Peter Risborg and Aaron Quigley. Nightingale: Reminiscence and technology—from a user perspective. In *OZeWAI 2003: Australian Web Accessibility Initiative*. La Trobe University, Victoria, Australia, December 2003.

Ronald Rosenfeld, Dan Olsen, and Alex Rudnicky. Universal speech interfaces. *Interactions* 8(6):34–44, 2001.

Christian Sandor and Gudrun Klinker. A rapid prototyping software infrastructure for user interfaces in ubiquitous augmented reality. *Personal and Ubiquitous Computing* 9(3):169–185, 2005.

Dan Shaffer. *Designing for Interaction: Creating Smart Applications and Clever Devices*. Peachpit Press, Berkeley, CA, 2006.

Ross Shannon, Eugene Kenny, and Aaron Quigley. Using ambient social reminders to stay in touch with friends. *International Journal of Ambient Computing and Intelligence* 1(2):70–78, 2009.

John Stasko, Myungcheol Doo, Brian Dorn, and Christopher Plaue. Explorations and experiences with ambient information systems. In William R. Hazlewood, Lorcan Coyle, and Sunny Consolvo, Eds., *Proceedings of the 1st Workshop on Ambient Information Systems. Colocated with Pervasive 2007*, Toronto, Canada, volume 254 of CEUR Workshop Proceedings ISSN 1613–0073, May 2007.

John Stasko, Todd Miller, Zachary Pousman, Christopher Plaue, and Osman Ullah. Personalized peripheral information awareness through information art. *UbiComp 2004: Ubiquitous Computing*, pp. 18–35, 2004.

Andrea Szymkowiak, Kenny Morrison, Peter Gregor, Prveen Shah, Jonathan J. Evans, and Barbara A. Wilson. A memory aid with remote communication using distributed technology. *Personal and Ubiquitous Computing* 9(1):1–5, 2005.

Lucia Terrenghi, Aaron Quigley, and Alan Dix. A taxonomy for and analysis of coupled displays. *Personal and Ubiquitous Computing*, 2009.

Jenifer Tidwell. *Designing Interfaces: Patterns for Effective Interaction Design*. O'Reilly Media, Inc., November 2005.

Brygg Ullmer and Hiroshi Ishii. Emerging frameworks for tangible user interfaces. *IBM Systems Journal* 39(3–4):915–931, 2000.

John Underkoffler and Hiroshi Ishii. Urp: A luminous-tangible workbench for urban planning and design. In *Proceedings of the SIGCHI Conference on Human Factors in Computer Systems: The CHI Is the Limit*, pp. 386–393, Pittsburgh, VA, May 15–20, 1999.

Koert van Mensvoort. Datafountain: Money translated to water, 2005.

Daniel Vogel and Ravin Balakrishnan. Interactive public ambient displays: Transitioning from implicit to explicit, public to personal, interaction with multiple users. In *UIST '04: Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology*, pp. 137–146, New York, NY, USA, 2004. ACM Press.

Daniel Wagner, Thomas Pintaric, Florian Ledermann, and Dieter Schmalstieg. Towards massively multi-user augmented reality on handheld devices. In *Proceedings of the 3rd International Conference on Pervasive Computing*, pp. 208–219, 2005.

Mark Weiser. The computer for the 21st century. *SIGMOBILE Mobile Computing and Communications Review* 3(3):3–11, 1999.

Michael Philetus Weller, Ellen Yi-Luen Do, and Mark D. Gross. Posey: Instrumenting a poseable hub and strut construction toy. In *TEI '08: Proceedings of the 2nd International Conference on Tangible and Embedded Interaction*, pp. 39–46, New York, NY, USA, 2008. ACM Press.

Pierre Wellner. Interacting with paper on the digitaldesk. *Communications of the ACM* 36(7):87–96, 1993.

David West, Trent Apted, and Aaron Quigley. A context inference and multimodal approach to mobile information access. In *Artiàcial Intelligence in Mobile Systems 2004 (workshop of UbiComp 2004)*, September 2004.

David West, Aaron Quigley, and Judy Kay. Memento: A digital physical scrapbook for memory sharing. *Journal of Personal and Ubiquitous Computing* 11(4):313–328, 2007. Special Issue on Memory and sharing of experiences.

Andrew D. Wilson. TouchLight: An imaging touch screen and display for gesture-based interaction. In *ICMI '04: Proceedings of the 6th International Conference on Multimodal Interfaces*, pp. 69–76, New York, NY, USA, 2004. ACM Press.

Andrew D. Wilson and Aaron F. Bobick. Parametric hidden Markov models for gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21(9):884–900, 1999.

Steve J. Young, Gunnar Evermann, Thomas Hain, Dan Kershaw, Gareth L. Moore, Julian J. Odell, Dave Ollason, Dan Povey, Valtcho Valtchev, and Phil C. Woodland. *The HTK Book (Version 3.2.1)*. HTK, 2002.