Context-aware design and interaction in computer systems

by T. Selker
W. Burleson

As the human-computer interface becomes more pervasive and intimate, it will need to explicitly draw upon cognitive science as a basis for understanding what people are capable of doing. User experience and situation should be integrated into the computer system design process. Situational awareness can be used to reduce the amount of explicit input a person is required to give a computer. Contextual information of what and where the user task is, what the user knows, and what the system capabilities are, can greatly simplify the user scenario. Such use of contextual models in computers can also reduce the teaching needed for the user to accomplish tasks. An approach to a framework for design of contextual computing is illustrated through examples.

Real-world situations in which a computer can be useful and new approaches to design are changing the computer-user interface. Industrial, graphic, product, and fashion designers teach their craft through developing the "eye." Design maturity comes from developing a visual knowledge of the history of design examples and fabrication, a desire to visualize by creating, and a command of the generative skills (rendering and fabrication techniques, structured design, brainstorming, ideation, rapid-prototyping, and iteration[1,2]). Design evaluation, while important, will not be a focus of this paper. The fields of aesthetic design have profound impact on our experiences. Designers affect how buildings and their contents, roads and the objects on them, and even communities look and feel.

These design fields have been magnets for creativity. The best products are the results of creative experimentation and close attention to the human experience in specific contexts. Many products are created and valued entirely for their design appeal. For example, except for its seductive design, the highly successful iMac** has complete continuity with the previous Macintosh** products that looked like a standard PC (personal computer). Physical design has increased the iMac's popularity by applying the concepts of fashion design. The design of what people wear, drive, and own represents how they want the external world to view them. People may dress up and wear expensive-looking watches in an attempt to enter the business world. Only someone not "in the know" about business appearance wears a digital watch to a meeting. Is wearing a digital watch "saying" that function is more important than looking good? Is it saying, "I have other things to attend to besides being social"? Is it saying the wearer cannot afford gold jewelry and the cost of maintaining a fancy Swiss mechanism? Designers have functional and stylistic responsibilities. Design affects how people feel about the environment they are in and the tasks that they carry out in that setting. Designers vary materials, structure, and surface properties throughout an object to create functional, semiotic, and aesthetic characteristics.

As many objects begin to incorporate computers, the fields of design and computer science are beginning

0018-8670/00/$5.00 © 2000 IBM

to merge. Aesthetics, ergonomics, and cognitive sciences are increasingly important factors in user-interface design. This paper discusses design issues concerning what people do and where they do things as channels of communication with computers. A related paper[3] focuses on technology for contextually aware computing. Computer science and engineering design typically approach problems by breaking them down into parts and applying standard solutions. In contrast, the traditional design fields promote aesthetics, ergonomics, and their relationships to function. The traditional design paradigm is broadening the structured design process for human-computer interfaces, as shown in this paper.

As classic design sensibilities become valued in new fields such as computer science, products become easier to use and have more connection to people's lives. The steps of structured design must connect with the aesthetic sense and sensitivity of people. Issues of contextual interaction and knowledge about people's desired activities and actions further affect the design process. Design will continue to focus on recognizing and utilizing cultural and physical languages of space and objects as important contextual cues. These cues will drive computer-user interaction in designed objects and environments.

Context-aware systems sense or remember information about the person and the emotional or physical situation in order to reduce computer-user communication and effort.[4] To interpret such "input," these systems must create, maintain, and be driven by models of the task, user, and system.[3] The computer must remember things about a person, the way the computer has worked in the past, and the way a person is trying to engage with the computer in the present. The emerging field of context-aware systems supports people's continued acceptance of computational technologies entering their lives. Context-aware systems will respect and adapt to people's views of themselves. At the same time, however, these systems will motivate user input through contextual cues and patterns of use. For example, the Music Ball presented later in this paper is instrumented to enable users to collaborate in musical composition.

Instrumentation is important to understand events in the world. Computer science instrumentation is extreme in the sense that computers require all information to be digital in order to be used. Early automotive applications included continual beeping to inform those who did not perform an action such as closing a door or a seat buckle. Such contextual but socially insensitive applications highlight the brittleness of translating from the digital to the physical. The robotics, user-interface, and graphics communities value advances in input and output technology as much as they do in computational advances. Long-term research focus in the robotics community has been directed toward the development of sensors of position, touch, distance, material, etc. When computers were expensive and instrumentation was difficult, computer applications rarely interacted with the outside world. Computers are now relatively inexpensive. The development of high-quality sensors and effectors makes them equally affordable. People have been developing new and sophisticated sensors for the detection of distance, position (i.e., the Global Positioning System or GPS), object recognition (i.e., radio frequency identification or RFID), connectivity (i.e., personal area networks or PANs), the electromagnetic spectrum, diverse chemistries, and even smell. These new technologies open possibilities for context-aware systems. As better sensors for biological function and identification methods utilizing micro- and nano-technologies are developed, further applications will emerge.

The development of input and output (I/O) technologies has always been intertwined with digital technology. A whole set of high noise-to-signal areas of recognition, which once seemed undecipherable, are yielding fantastic successes. Speech recognition, handwriting recognition, and image understanding are now in widespread use as input sensors for computer-use scenarios. Sensors that focus a camera based on where the photographer is looking are now on the market.[5] Almost 30 years of work on voice recognition has yielded systems like IBM's ViaVoice*.[6] A few years ago, an untrained handwriting interface in a consumer product seemed unfathomable. Now products like PalmPilot** and CrossPad** depend on the ability to recognize hand gestures.[7] Recognition technologies are becoming reliable "smart sensors." These "smart sensors" put designers in a position to create increasingly complex scenarios. As personal digital assistants (PDAs) help people meet and greet each other, they become part of an increasingly sophisticated social fabric.

This paper addresses the following issues: In what ways can design reduce the complexity of computer tasks? What contexts are most improved by the application of computers with sensors, models of users, and models of tasks? These questions are answered

using a sequence of examples that rely primarily on contextual setting and use.

Framework

This paper describes creating context-aware systems using increasingly sophisticated task, user, and system models. The paper introduces a framework for analyzing them along with a discussion of design techniques and technology.

User and computer *task models*[3] are the acts that a person performs to accomplish a task with a system. As an example, consider a person crossing a threshold between rooms. The task at the threshold for a person is to open a door or to see if someone is inside. The function of the threshold is to keep people out and to let people in when appropriate. A new augmented task model might include a Digital Threshold (described later) telling a visitor that the occupant is busy and asking if the visitor would like to schedule an appointment. Task models for both user and computers allow modeling of expected behaviors.

Computer *user models* have been variously defined. The Grundy System developed by E. Rich[8] used a list of characteristics about a library visitor identified in a "stereotype." Deep Blue*[9] used a list of game starts and style of play to characterize a chess opponent. The user model, developed and stored in the computer, consists of task-relevant background information about the user (i.e., the Digital Threshold described later "knows" who its users are, what access they should be given, and their preferences). Users' needs are often intangible, affected by habit, self-image, and even issues of motivation (i.e., a person might be most efficient in the morning). Aspects of the user model that focus on what a person wants to accomplish have at least two elements: user comfort and user congruency. User comfort with certain thoughts, actions, and words naturally affects usage. User congruency conveys the match or compatibility between a user's preferences and the designed artifact. For example, a Web-based grocery delivery company may offer users a service congruent with their needs. However, the potential user may choose not to avail him- or herself of the service because of issues of comfort; for example, the user's discomfort could increase with letting others know what he or she purchases. Similarly, a person might use a personal digital assistant to appear technologically savvy, a decision based more on issues of congruency than of comfort, because the individual may not yet feel comfortable with many aspects of the PDA. This congruency also describes users' desires to be seen a certain way as they present themselves, through look and action. The user model, therefore, constitutes a composite of task-relevant background information, coupled with design that attends to comfort and congruency.

The *system model* refers to the capabilities of the computer system—its structure and ability to accomplish an existing task. For example, the Music Ball, described later, has the ability to play sound (speakers), and to track a three-dimensional input device (a Polhemus tracking device).[10] To this extent the system model for the Music Ball can help people relax and keep them entertained.

The fields of design and computer science are beginning to merge.

Context-aware computing relies on the task, user, and system models to meet the explicit congruency and implicit comfort needs of users while simplifying life. The examples later incorporate increasingly complex models, which are used to create the value and flexibility of the systems as interaction tools. The techniques and technologies used to develop context-aware systems include examples both with and without computation. Many context-aware scenarios can be spatial or mechanical without any need for electronic sensors. Household keys, for example, are made to work for people in particular places. Physical (nonelectronic) examples include a way to learn about a product as it is unpacked or a plug that works in any power outlet. Next in the sequence, analog sensors and switches are the simplest kinds of electronic sensors. Even an automatic door that opens when a mat is stepped on is context-aware. Slightly more advanced context scenarios develop when we couple a switch with an internal clock. An example of such a system, useful for intermittent interaction with the object or sensor, is an automated watering system that measures moisture and uses time to decide when to water plants.

Integration of numerous sensors in support of or driven by sophisticated task, user, and system models can support still more complex scenarios. Modern exercise machines use weight, height, heart rate, etc., to monitor and prescribe a training experience. The Electronic Wallet described later integrates many activities through sensors and scenarios chosen to expand the capabilities and contextual domains of a standard wallet. Memory, reasoning, and learning systems can be powerful elements for the design and development of context-aware systems. A simple form of memory context can be coded as a single program variable, but in many of the design examples the structure of memory and learning is complex. Artificial intelligence (AI) examples include the adaptive Music Ball and the COACH system,[11] both of which interact with users to develop a user model and dynamically adjust the system's interaction to facilitate the users' desired goals.

Context-aware system design relies centrally on sensing and modeling situations and recognizing rules for engagement, motivation, and productivity. If the system is not seductive, if its form and function do not meet the comfort and congruency needs of the user, then it will be used only when no alternatives exist. Physical materials and the way that they are displayed affect how they are perceived and what they do. For example, it might be nice to show visitors a video when they walk up to a door, but the form of a 40-pound CRT (cathode-ray tube) monitor mounted in a door would make the door unwieldy and dangerous. A flat-panel display or a rear-projected touch screen might be seductive alternatives. The surface of an object one would bring into the house matters; the texture of an object affects whether a person can or wants to touch it, wear it, or take it with them. Specifically, the spiky side of Velcro** does not feel as good against skin as the fuzzy side; the gray metal surface of a card punch would not fit nearly as well in a home library as the rubberized black of an IBM ThinkPad* computer. Design and aesthetics can affect a user's perception, belief system, and even task ability.[12] Function, style, perceived reliability, and ease of use of the designed object in the scenario are as important as the design's ability to support a task. Four elements that have been said to aid in creating seductive interfaces include creating instinctive emotional responses; connecting with personal values and goals; creating ongoing surprises, and enabling discovery of the unexpected or deeper value.[13] These can contribute to the design efforts for context-aware systems.

Design prototypes of context-aware systems

People enjoy personalization of their belongings, e.g., the name carved in a treasured pocketknife or the sports-rack put on a car. How is it that our lives have not been full of more things that get to know us, personalize themselves in more than physical ways, and improve as a consequence? There will be. We will make toys that get to know us as we play with them. We will make clothes that insulate us only when we want them to. We will have context-aware computers, which will learn and attempt to further our interests and goals.

The following examples demonstrate a progression of uses of technology to reduce the need for explicit communication between a person and the system. In contextual design, as in all design, the objective is to use the simplest solution that achieves the functional and communication goals. To demonstrate the framework, the examples that follow explore increasingly complex technology used to capture context.

Noncomputational models. Multimedia tutorial preloaded software often consumes sizable portions of new computers' hard disks when delivered to customers. Though beautiful, these tutorials sometimes seem particularly context-inappropriate in that the computer has to be operational for a user to view the tutorial on how to fix the computer. Packaging has always been an important part of product design. The following describes an attempt to make the most useful, simple, and accessible computer setup help system. This system resulted from a study conducted for the IBM ThinkPad product in 1996. The goal was to integrate setup instructions of a personal computer with the Out-Of-Box Experience (OOBE) (Figure 1). Design iterations developed the "wrapping paper" concept in which the step-by-step setup of the computer relied on a context-aware methodology. The wrapping paper puts the instructions on strips of paper that are inserted in the machine itself during packaging (one strip is on the outside, another becomes visible when the machine lid is raised, and another when the keyboard is lifted). This both keeps the instructions close to the product features and sequences the instructions. The user is presented with new actions only when they are useful. The three-inch-wide strip of paper uses graphics to draw a user through seven steps of setting up and turning on the computer as the user extracts the paper. Don Norman's similar concept of "forcing func-

Figure 1 The Out-Of-Box Experience presents instructional information in direct relationship to setup tasks using "wrapping paper."

Figure 2 The Electronic Necklace, without a power source or computation, uses interaction with motors, lights, and sounds to create a baseline application for context-aware research.

tions" reduces user error by restricting the number of paths the user can follow.[1]

Even the seal used in the wrapping paper example can have many implied values: a seal certifies the quality of the product, states that the computer is new, requires the user to break the seal to open the box, and, when broken, indicates acceptance of the licensing agreements. Setup instructions are presented in the order in which they occur and in close physical proximity to the product features they describe. A competing proposal was to develop a "quick setup guide" as a separate document. However, it suffered the typical disconnection between documentation and physical hardware. The design of the packaging anticipates the coincidence of information and the need for it. Giving the information with arrows, pictures, and diagrams increases the integration of the teaching materials into the context of opening the product. The OOBE user model assumes that a user who encounters it is a novice opening a computer for the first time. The task model is that the person is setting up a notebook computer. The system model is that the paper instructions are near the components they describe and are revealed as the person opens the computer. The process of education and discovery is seductive at every stage.

A context-aware system that uses sensors to respond to a user can be made without a computer or even a power source. The Electronic Necklace[14] (see Figure 2) is composed of motor/generator and other electronic bead devices such as squeaky enunciators or light-emitting diode (LED) lights. When a user turns one of the beads, the power is transmitted from the motion of the generator/motor bead through the wire necklace to activate the other beads. The turning beads massage the person wearing it and entertain others as well. Other special beads produce sound or light. The task model involves massage and entertainment. This is a simple yet compelling aesthetic interactive device. This device has no memory, requires no computation and no power. It requires the user to generate electricity to make physical events happen, and it is quite successfully responsive to the user. It raises several issues for designers to consider, for example, whether the system should have a task model that includes teaching a person about piezoelectric speakers, magnetics, and LEDs. The designer might also consider the effects of adding computation, memory, and learning algorithms. The necklace might be considered the baseline level of context-aware electronic systems. The necklace can do so much without a computer. When a computer is used in systems it must add more value to its context scenario than the necklace or OOBE achieve without computation.

The Universal Plug[15] (Figure 3) is a system that is capable of sensing and functioning in several contexts (in this case, prong configurations for electrical outlets in different countries) and uses mechanical logic to accomplish the task model, which is to

get power from an electrical outlet. When the plug is pressed against a wall it automatically selects the correct voltage and power prongs to fit anywhere in the world. If there are holes in prong positions for an American plug, the American prongs become fixed in the out position and can be inserted into the wall, while the European and British prongs retract into the plug base. Likewise, if the outlet is British, the British prongs are inserted into the outlet, while the rest retract. This plug will work around the world without the user having to attend to the differences among outlets. The selection is made by the force of the user's hand, the interaction of the holes on the wall, and the mechanical decision-making logic internal to the plug. The task model of getting power from an outlet is integral to the functionality of this device. The system model enables physical pressure to select mechanical prongs and pawls. The user model assumes that the user will reset the prongs and plug it into various outlets.

Context-aware computer systems. Context-aware computer systems include sensor-based systems, multifunction systems, and proactive model-based systems.

Sensor-based systems. The Flexor (Figure 4) is a device that recognizes contextual communication through arm motion. Socially, arm motions are significant. When people move their arms wildly we stand back; when they wave, we wave back. While costumes have employed blinking lights and sound for some time, they have not focused on distinguishing user goals. The Flexor consists of an analog force-sensing variable-resistor measuring elbow position, a microprocessor, lights, and an enunciator as effectors. In contrast to wearable games such as the electronic gloves that give feedback on a monitor,[16] Flexor has an integrated display. It is an electronic sleeve that allows a person expressive capability based on a user model of the person's arm movements. Flexor uses an arm-bending sensor to evaluate motions made by an arm. When the user is moving an arm as though he or she is dancing, it flashes a varying sequence of lights and sounds. When the user is moving an arm as in exercising, Flexor counts how many repetitions have been completed. This device can also be used to encourage and motivate specific patterns of physical activity and therapy. Flexor is an example of context-aware computing in which a model of what a person is doing allows the computer to help that person without any explicit direction from the user.

Figure 3 Universal Plug mechanically incorporates task model and situational awareness.

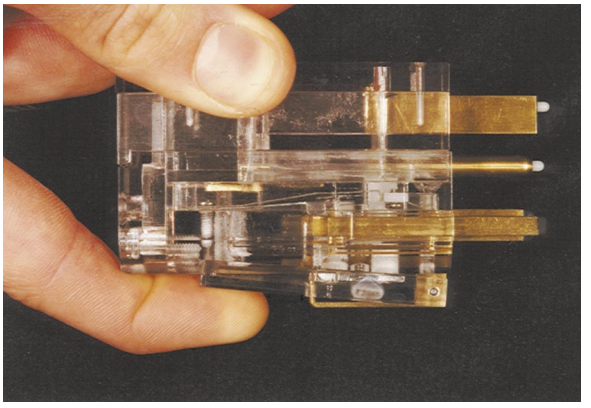


Figure 4 Flexor provides visual and audio feedback to motion for therapy and performance.

Typically, people wear jewelry both for their own satisfaction and for the effect it has on others. As society moves into the digital age, we increasingly use digital technology in wearable items. Sometimes these implementations are intrusive or confusing in social situations, but more often they are not. The Flexor's role is clear: it is an explicit, dramatic, seductive, and audacious use of lights and sounds in a wearable computer that augments one's personal expressive manner and physical activity. The flashing lights and audio signals seductively incite interest in the device and make visitors eager to try the Flexor themselves. Its clarity of feedback obviates the frustration of understanding difficult instructions. Users are able to quickly grasp the concept of the

Figure 5 Digital Threshold uses simple switches to create and maintain a meeting schedule for an office occupant.

sleeve and select among the scenarios included in the user model. The Flexor efficacy is surprisingly robust, even with only one sensor; ideas and scenarios are easily detected and implemented.

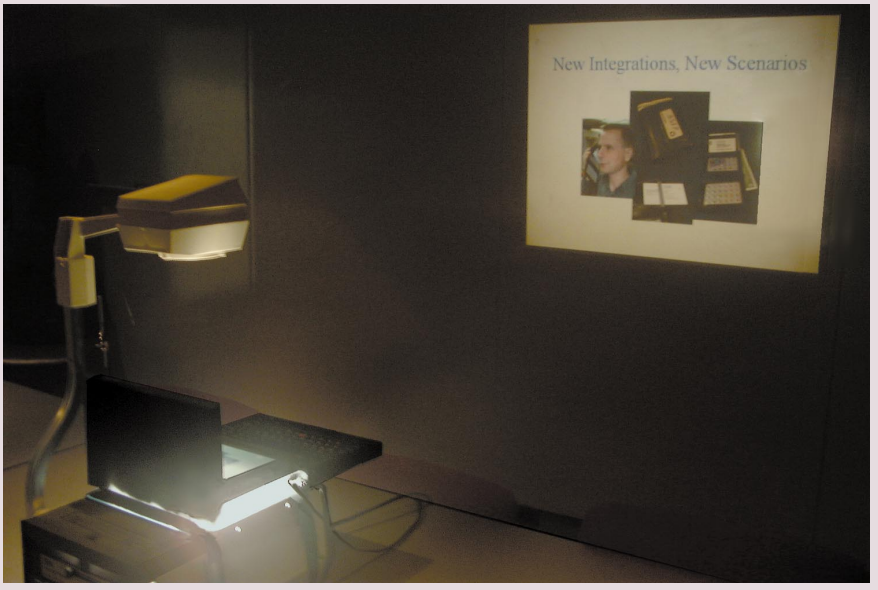
Computer-user interfaces have historically required users to explicitly type, send cards, or select actions through recognition. In contrast, the Flexor accomplishes context-awareness based on spatial positioning. In a similar development, fishing and golf toys sense a user's cast and swing. These interesting toys have been doing well simply by recognizing a single action per scenario and providing feedback. A highly developed example of this genre is Vadim Gerasimov's baseball bat, which depends on accelerometer data to provide a batting coach.[17]

Whereas the bat is designed to teach a user to hone a task, the Flexor distinguishes among many tasks. Flexor senses that a certain pattern of motion is being performed; it responds according to scenarios built into the user model. Each arm motion scenario is like the script in a story. Recognizing story scripts has been an important approach in many AI systems for selecting objectives.[18] A group of related motions, when performed in succession, trigger a corresponding script. The Flexor senses groups of motions and couples them with a category of action. This work demonstrates that single-sensor gesture recognition can be used to recognize actions through a user model. This enables the Flexor to distinguish and display the style and communication goals of a user.

Whereas the Flexor assumes things about one person, another more sophisticated device can assume and mediate information about two people. The threshold of a door might be the most important demarcation in a social environment. It is the space that a person walks by in order to find out if a colleague is in his or her office. Should the person enter or should a scheduled meeting be arranged? If there are people in the room is it all right for the visitor to enter? The Digital Threshold[18] relies on two switches, one on the outside and one on the inside of the door (Figure 5). Coupling these switches to a voice recognition system and Microsoft Office**, the system counts the number of people in a room, notices whether a person has an appointment, identifies people, allows the occupant to respond, and updates and manages the office owner's calendar. The visitor or office occupant can modify the task such as bringing someone in, rescheduling a meeting, or leaving a message. The Digital Threshold adapts its task model relative to people's positions, statements, and schedule to mediate meetings. This system adds two-way interaction with the office task model as well as with the office occupant to make a collaborative system with a dynamic modifiable task model.

Multifunction systems. A designer's task is to overcome constraints in order to find solutions to make function and form work together. Integration is a technique for making something more context-aware by combining many functions into one design. For example, the "multifunctional army knife" approach to many designs can be a productive way of making users believe that their particular situation has been taken into account. Making a tool that will do most everything is a risky although sometimes effective design goal. The personal computer attempts to be the multifunctional army knife of our time. It runs games, finds information, enables communication, types letters, does accounting, and checks paperwork.

Figure 6 ThinkPad 755CV provides multiple functions as a personal and presentation computer.



The IBM ThinkPad 755CV notebook computer (Figure 6) is a system designed to work differently for two diverse situations. In addition to being useful as a standard portable computer, it is capable of being used as a liquid crystal display (LCD) panel for an overhead projector. Its display was designed so that a person could remove the plastic back cover from the LCD panel and place the laptop computer screen on an overhead projector. The computer's LCD panel then displayed the presentation on a wall or screen. The user could place a finger on the LCD panel and, like a transparency, a shadow would appear on the projected image. Placing the notebook computer on top of an overhead projector produced an image that was excellent for presentations. However, a difficulty of the overhead projector computer was the invisibility of the new function. Some people never realized that it could project. Multiuse objects must be designed with care to articulate each function clearly. Multiple goals are sometimes poorly integrated. We all hate to have one feature of a system compromise another feature's goal; a multifunction army knife does not drive in a screw as well as a standard screwdriver does. "Creeping featurism" does not always make for objects that are easy to comprehend or use, even if they offer more functions and do not eliminate any primary functions. There are always many trade-offs in the design of a multifunction system (i.e., remembering why a tool is being picked up can be difficult if it can do several different things).

In a subsequent contextual design study, an additional feature was added to the overhead computer. A simple mechanical modification allowed sunlight rather than internal light to illuminate the display. The advantages of using the sunshine in this manner are that the screen can be seen outdoors, that the external lighting provides a brighter and richer display than fluorescent backlighting, and that the power requirements of the computer are reduced by the use of ambient lighting. The success of this integration is that the different modalities of display support each other. Manipulating the display itself improves it by choosing between standard and ambient modes. Outdoors, the notebook's main limitation, a dim display, is the prompt for using the ambient light mode. If one is giving a presentation to a group, people's need to peer over one another should prompt the use of the overhead projector.

Figure 7 Electronic Wallet integrates multiple technologies to support diverse identity and financial tasks.

Once people know about the capabilities, they can change between modes in a few seconds. This contextual approach was used by courtside judges at Wimbledon, England, in 1996 and 1997 to record and comment on tennis matches.

A smaller, more personal and powerfully integrated design is the Electronic Wallet. While today's wallet is eminently useful in many circumstances, the Electronic Wallet has features that expand its usefulness in these contexts and extend its functionality to new contexts. People typically carry a wallet, keys, and sometimes a watch or pen when not at home. Wallets today carry papers, money, and codes (i.e., credit and debit cards). A wallet represents the credentials and wealth the owner needs to make available while in public. Seen from a functional point of view, a wallet is a toolbox, sitting outside of the awareness of the user when not in use, but close at hand when needed. It is an integration of documents that people take from their pocket: lists for buying things, pictures for sharing, receipts and business cards, etc. The Electronic Wallet (Figure 7) reduces the physical materials in a person's pocket while improving the ability to identify the carrier, record financial transactions, and present personal effects with style. A context-aware system design approach can revolutionize the toolbox in several ways.

The design of the Electronic Wallet began with a use scenario consisting of the diverse contextual environments and capabilities it needs. Here is an example of how it might work. The Electronic Wallet contains a display on its outside surface that normally displays time in large numbers. Let us assume that I am carrying one. Now it vibrates gently in my pocket and I look at the external display to see that a friend is paging me. I open my wallet and it shows times when I received previous pages from this number. I lift it to my ear to call my friend back.

Later, the wallet vibrates again, displaying a reminder for me to meet a friend at a restaurant. When I arrive at the restaurant and meet my friend, I learn that he has a new job. He hands me his new business card and I scan it into the wallet. Optical character recognition (OCR) software updates the wallet's address book. After we sit down, I open my wallet to the central display inside and take notes on ideas to discuss later. We talk and I use the half-QWERTY keyboard to touch type with one hand[19] and enter notes for later. At the end of our meal, I hand the waiter the generic credit card, with my major credit card number encoded on the magnetic stripe. The wallet remembers my credit card numbers, so I need to carry only one plastic card. If I forget to retrieve the card, within a few minutes the card's volatile memory loses the number. My wallet vibrates to remind me that I have failed to retrieve the card. In the usual scenario, however, the waiter returns with a receipt, and I scan it into my expense database.

First attempts to make integrated wallet devices produced checkbooks containing calculators. The calculator added bulk and cost, however, while the simple calculations were easily done mentally. Consequently, they were not widely adopted. Early PDAs aggregated technology, rather than synergistic function. In contrast, new designs should have more util-

ity (e.g., be smaller, more convenient, and more functional) than the devices they replace. In particular, social and stylistic issues are central to deploying context-aware systems successfully. The Electronic Wallet starts with the task model of a standard wallet. It is full of personal items that make up its user model. Its system model is to deliver and record important documents, communicate wirelessly, and display information. To enhance user acceptance, the industrial design of the wallet encourages users to think of the functions they would expect a typical leather wallet to have. Such metaphorical implementation of technology has been shown to reduce learning effort and increase productivity.[20]

Proactive model-based systems. The above scenarios address issues of presenting, sensing, and interpreting activity in context. In these cases the system supports typical activities. More ambitious context-aware computing occurs when the system actively drives or trades off control with the user. The task model generated by the computer and based on its adaptive user model drives interaction in the next two examples. The first is a computer help agent that tutors users based on their abilities, and the second is a responsive Music Ball that creates music through a shared process of proposing music and interpreting user feedback.

COACH is a software agent that creates a dynamic user model by observing user actions with respect to frequency, duration, and efficiency of use.[3,11] The COACH system strives to navigate these stages by profiling its user's progress through them. The COACH system separates the task, user, and system models for learnable things into novice, intermediate, professional, and expert levels. The *novice* is concerned with what something is, the *intermediate* with starting to use it, the *professional* knows one highly efficient use, and the *expert* knows how to do everything. The system assumes that for each learnable thing, one must determine what it is (with a description), how to use it (with examples), and how it can be used in general (with syntax). The system adapts to the user at the user's level; for example, if the user of the system is a novice, COACH provides a lot of assistance. If, however, the user appears to be proficient, the adaptive help system stays out of the user's way.

Similarly, the Music Ball system takes the approach of proposing ideas to the user. This represents a rarely described computer-driven scenario in which the computer translates user reactions into computer actions. The user moves a ball; the system makes a subtle vocalization. The user either likes it and repeats the motion or does not like it and does something else with the ball. In this feedback loop, the ball draws on its user model and feedback to hypothesize sounds that the user will appreciate. The ball-user collaboration creates a language of sounds and associated motions. The user produces sounds by making known motions. The user and system expand their repertoire with new motions. When the user makes new motions, the ball tests its new theories about what the user likes.

The Music Ball system is written in the Java** language and uses a Polhemus, six-degree-of-freedom tracking device for input.

A stated goal of context-aware computing is to reduce the communication overhead for people. To the extent that AI modeling can track and change the role of people, the scenario can become more than mere delegation. The Music Ball demonstrates a possibility of creating smart relationships in which the computer and person together accomplish something outside the skill set of the person. The Music Ball uses the person's act of criticizing music to drive the creative act. The Music Ball's system model uses AI strategies for music teaching and structure. It also communicates with an enunciator and a motion sensor. In this example, the contrast between the user and Music Ball task models creates the power of the scenario. The user's task model is to make music. The ball's task model is that it wants to learn how to make a music language for the user. It is precisely the difference in task models that allows the user and ball to have different roles in their collaboration. The ball's user model is a set of music hypotheses and data about user-accepted sounds. The user actions are simple but the computer's decisions are complex and context-dependent.

Conclusion

The context-aware design framework focuses on design that accounts for contextual information or setting. This paper has shown how context can be embodied in the form and material of design, in the physical attributes, sensors, internal models, and in computer reactions. Context-aware system designs anticipate where and in what ways designed artifacts will be used. Watching and recording users' actions and needs can change users' experiences and introduce new and beneficial results. Memorable designs must be particularly useful, capable, aesthetically

pleasing, or powerful. Offering seductive qualities is an important way that a context-aware design can attract people. Systems can help people accomplish tasks and enable users to learn and accomplish new things. In order to achieve these goals, designs must fit into the lives that users have already created. Designers must remember that people bring into their lives things they find comfortable and things that are congruent with their expression to others of who they want to be.

Designers of contextually aware systems have a myriad of tools at their disposal; it is important that they choose the right tools for the right applications. Through examples we have shown how contextual awareness can be developed for diverse contexts and at different levels of complexity by using the appropriate tools. Even a physical system such as the Universal Plug adapts based on a task, user, and system model of how it will get power from a wall. A slightly more complex system like the Digital Threshold uses sensors to select and access task and user models to negotiate the meeting of a visitor with an office occupant. Drawing upon context-aware design, sensors and models can be aggregated into natural synergies like the Electronic Wallet. Integrated context-aware design can create powerful multifunctional tools that are easy to use and understand. On a more sophisticated level, the Music Ball demonstrates using an AI task and user model to enable simple sensors to drive an elaborate musical collaboration between user and systems. Context-aware design focuses on reducing communication barriers by analyzing what can be known about a user and how to support that information with task, user, and system models. This approach can be included in integrated design to combine the designer's aesthetics, the engineer's desire for efficiency, and the cognitive scientist's understanding of human potential.

For some years, we have been building laboratories to explore these ideas. The OOBE, Universal Plug, Electronic Wallet, ThinkPad, and COACH are a few of the context-aware projects built at the IBM User System Ergonomics Research group (www.almaden.ibm.com/cs/user/). The Electronic Necklace, Music Ball, the Flexor, and the Digital Threshold, are samples of the projects we have created in our new Context-Aware Computing Research Group at the MIT Media Laboratory (www.media.mit.edu/context/).

*Trademark or registered trademark of International Business Machines Corporation.

**Trademark or registered trademark of Apple Computer, Inc., 3Com Corporation, A. T. Cross Company, Velcro Industries, B.V., Microsoft Corporation, or Sun Microsystems, Inc.

Cited references and notes

1. D. A. Norman, *The Psychology of Everyday Things*, Basic Books, New York (1988).
2. V. Papenek, *Design for the Real World*, Pantheon Books, New York (1971).
3. H. Lieberman and T. Selker, "Out of Context: Computer Systems That Adapt To, and Learn From, Context," *IBM Systems Journal* **39**, Nos. 3&4, 617–632 (2000, this issue).
4. R. W. Picard, *Affective Computing*, MIT Press, Cambridge, MA (1997).
5. See www.canon.com for EOS-3 eye-controlled focus camera.
6. IBM's ViaVoice, see www.software.ibm.com/speech.
7. K. Silberger, "Putting Composers in Control," *IBM Research Magazine* **34**, No. 4 (1996); see www.research.ibm.com/resources/magazine.
8. E. Rich, "Users Are Individuals: Individualizing User Models," *International Man-Machine Studies* **18**, 199–214 (1983).
9. Deep Blue, see www.research.ibm.com/deepblue/meet/html/d.3.html.
10. See www.polhemus.com/isotrkds.htm.
11. T. Selker, "COgnitive Adaptive Computer Help (COACH): A Case Study," *Advances in Computers*, *Volume 47*, M. Zelkowitz, Editor, Academic Press, New York (1998), pp. 69–140.
12. M. Csikszentmihalyi, *Flow: The Psychology of Optimal Experience*, Harper Perennial, New York (1990).
13. J. Khaslavsky and N. Shedroff, "Understanding the Seductive Experience," *Communications of the ACM* **42**, No. 5, 45–49 (May 1999).
14. See www.media.mit.edu/context/projects/necklace/html.
15. Universal Electric Power Plug, U.S. Patent No. 5,423,690 (June 13, 1995).
16. T. Zimmerman, J. Lanier, C. Blanchard, S. Bryson, and Y. Harvill, "A Hand Gesture Interface Device," *Proceedings of CHI and Graphics Interface '87* (April 1987), pp. 189–192.
17. V. Gerasimov and W. Bender, "Things That Talk: Using Sound for Device-to-Device and Device-to-Human Communication," *IBM Systems Journal* **39**, Nos. 3&4, 530–546 (2000, this issue).
18. H. Yan and T. Selker, "Context-Aware Office Assistant IUI2000," *International Conference on Intelligent Interfaces*, New Orleans, January 2000, ACM New York (January 2000), pp. 276–279.
19. E. Matias, I. S. MacKenzie, and W. Buxton, "Half-QWERTY: A One-Handed Keyboard Facilitating Skill Transfer from QWERTY," *Proceedings of the CHI '93 Conference on Human Factors in Computing Systems*, ACM, New York (1993), pp. 88–94.
20. S. A. Douglas and T. P. Moran, "Learning Text Editor Semantics by Analogy," *Proceedings of the CHI '83 Conference on Human Factors in Computing Systems*, ACM, New York (1983), pp. 207–211.

Accepted for publication April 27, 2000.

Ted Selker *MIT Media Laboratory, 20 Ames Street, Cambridge, Massachusetts 02139-4307 (electronic mail: selker@media.mit.edu).*
Dr. Selker is an MIT professor focusing on context-aware computing. Before joining the MIT faculty he worked at IBM for over

a decade, created the User System Ergonomic Laboratory, and was named an IBM Fellow. During that time he also served on the faculty of Stanford University. He is recognized for the design of the "TrackPoint® III" in-keyboard pointing device, for creating the "COACH" adaptive agent that improves user performance (Warp Guides in OS/2®), and for the design of the ThinkPad 755CV notebook computer that doubles as a liquid crystal display projector. Dr. Selker obtained his B.S. degree from Brown University, his M.S. degree from the University of Massachusetts, and his Ph.D. degrees from the City University of New York in computer science and information sciences and applied mathematics. Prior to joining IBM Research in 1985, he worked at the Xerox Palo Alto Research Center, Atari Research Labs, and Stanford University, and was also a Stanford consulting professor. He is Chief Scientist for Vert Corporation, is on the Board of Directors for GetGoMail.com, and is on the Board of Advisors of FindTheDot.com and Xift.

Winslow Burleson *MIT Media Laboratory, 20 Ames Street, Cambridge, Massachusetts 02139-4307 (electronic mail: win@media.mit.edu).* Mr. Burleson is a Ph.D. candidate in the Context-Aware Computing Group at MIT's Media Lab. He has worked in the User Group at IBM's Almaden Research Center, lectured at Stanford in mechanical engineering, worked with the SETI Institute, and was a coprincipal investigator on the Hubble Space Telescope's investigation of binary asteroids. He holds a bachelor's degree in physics from Rice University and a master of science degree in engineering from Stanford University's Mechanical Engineering Product Design Program.