CHAPTER Two

Design Considerations for Information Appliances

MICHAEL F. MOHAGEG, PH.D.
User Experience Group
Consumer & Embedded Division
Sun Microsystems, Inc.

ANNETTE WAGNER
User Experience Group
Consumer & Embedded Division
Sun Microsystems, Inc.

2.1 Introduction

Current user interface approaches, developed for the desktop personal computer, are limited in their application to the design of information appliances. While portions of PC design approaches can be leveraged for information appliances, many of the design philosophies do not apply or simply cannot be used. Similarly, only a subset of the considerable knowledge from consumer electronics user interfaces applies to the information appliance domain. Therefore, new design approaches are needed to address the special design needs for these devices.

This chapter describes some of the practical design considerations for developing user interfaces for information appliances. These considerations have been derived from significant experience in designing for the information appliance space. In addition to its traditional server, workstation, and software businesses, Sun Microsystems provides software solutions for consumer devices and information appliances. These solutions have been in the form of a real-time operating system (RTOS), enabling platforms such as Java2 Micro Edition (for phones, televisions, automobiles), and complete applications for television set-top boxes, screen phones, and wireless devices.

The design considerations in this chapter are a sample of the principles yielded from our experiences. They are intended to provide the flavor of our new design approach. They are not design prescriptions, but they do offer a perspective for a user experience that is different from that of PCs. Additionally, the principles discussed in this chapter are supported by a number of design tactics and implementations, which, in the interest of space, are not included in this volume.

Information Appliances Defined

An information appliance is a computer-enhanced consumer device dedicated to a restricted cluster of tasks. A personal digital assistant (PDA), an Internet-enabled screen phone, and a pager are examples of information appliances. The concept of information appliances is borrowed from the traditional notion of an appliance: it is a device that performs only a few tasks, but does them well, efficiently, and with little conscious effort from the user. For instance, refrigerators are bought solely for the purpose of keeping items either cool or frozen; they do little else other than blend aesthetically with the rest of the kitchen. A dishwasher washes dishes. A microwave oven has the task of heating food. Information appliances apply this notion of a dedicated device to computing technology, with the goal of creating small, easy-to-use, low-cost devices that perform only a few tasks. Instead of using a PC that does

everything from email to 3D game playing, users can have inexpensive information appliances that perform a restricted collection of tasks. Proponents of these devices claim the user experience of information appliances will be superior to general-purpose PCs because all aspects of the product can be specially designed to meet the needs of the more-restricted device.

Some of the key characteristics of information appliances that differentiate them from PCs are

- Limited purpose and functionality
- Not necessarily extensible or upgradable
- Replacement expectation (the user may have to replace the entire device within a few years)
- Perceived as less expensive (versus PCs)
- Perceived as less complicated to run and maintain (versus PCs)
- Very easy to learn and use
- No expectation of "expert users"

There is another important factor contributing to the viability of information appliances. In comparison to mass market consumer products, PCs have not achieved high penetration and popularity among personal or home users; this is especially true outside North America (Computer Industry Almanac 1998). Many (AllNetDevices 1999) attribute this lack of popularity to the high cost and complexity of using and maintaining PCs. Information appliances offer a compelling alternative due to their low cost and ease of use. For instance, customers may legitimately wonder why they should spend over $1,500 on a PC (with a variety of maintenance complications) when their only interest is in a device to use email and access the World Wide Web.

The Need for a Different User Interface

Information appliances need different user interface solutions for several reasons, but the two most important are

- The consumer audience
- The characteristics of information appliances

Each of these is addressed below.

The Consumer Audience

Information appliances are intended for a very wide base of "consumers"—people who, unlike PC users, may have minimal computer experience. For consumers, user interface metaphors and models that are borrowed from desktop environments may not be appropriate. Pop-up menus, scrollbars, drag and drop, or the computer desktop may be quite unfamiliar. Consumers may feel uncomfortable dealing with anything they consider to be too "high tech" and tend to be unwilling to learn complex interaction models. On the other hand, they are familiar with appliances such as push-button phones, microwave oven control panels, and TV remote controls.

Consumer electronics appliances such as televisions, VCRs, and telephones are common today; however, widespread acceptance of a consumer electronics device can be difficult to achieve. Some products fail because consumers find them too difficult to use. And though some products sell well, consumers actually use only a small fraction of their capabilities. Modern consumers have little patience for learning how to operate new products, and without bothering to consult the user manual, they expect the interfaces to be self-evident. Moreover, our experience has shown that consumers—even those with computer experience—have very different expectations when they interact with a consumer device. There seems to be a low tolerance for learning how to use them.

The Characteristics of Information Appliances

The strength of consumer devices lies in their specialization for particular tasks or groups of tasks. Indeed, what makes certain devices compelling and successful in the marketplace is finding a targeted collection of tasks and supporting them with the right features in the product. Product designers are no longer bound by the generic technology offerings of a PC: CRT and speakers for output, keyboard and mouse for input, and fairly standard, "desktop-oriented" end-user environment to manage and access information. Products can be designed to provide an environment that is well suited to the requirements of the appliance. For instance, small monochrome LCDs with a touch panel may be appropriate for a portable device. A screen phone may not use a physical keyboard at all and instead rely mostly on gesture-based input with a stylus. A PDA user may not need a "desktop" metaphor with files and folders to manage his PDA "objects." Table 2.1 highlights some of the differences between a PC and two popular information appliances—screen phones and TV set-top boxes.

While designers of information appliances are freed from the trappings of the PC, there are still considerable design challenges, due largely to technical limitations of these devices. Compared to PCs, information appliances

TABLE 2.1

Some of the key differences among a screen phone, analog TV set-top box, and a desktop computer

Characteristic	*Screen Phone*	*TV Set-Top Box*	*Desktop Computer (personal computer or workstation)*
Applications: telephony, voice mail, address book, email, Web browsing		TV, EPG, Web browsing, email	word processing, spreadsheets, presentations, Web browsing, email, productivity applications, vertical applications
Input device: primarily finger, stylus, secondarily keyboard		remote control and remote (IR) keyboard	mouse, keyboard
Mouse support	none	none (some remote controls have trackballs)	yes
Keyboard support	on-screen and/or physical keyboard	on-screen and/or physical keyboard; keyboard usually infrared technology	physical keyboard
Viewing distance	1–1.5 feet	10–15 feet	1–2 feet
Display size (diagonal)	6–8 inches	13 inches to wall size	13–28 inches
Screen resolution: ¼ VGA (320 × 240) to full VGA (640 × 480)		broadcast television analog signal (roughly equivalent to 640 × 444 for NTSC)	640 × 480 to 1800 × 1440
Display colors	2-, 4-, or 8-bit color; both black and white and color	broadcast television; color characteristics depend on signal type (e.g., NTSC, PAL, SECAM)	8-, 16-, or 24-bit color
Pixel density	~102 dpi	N/A	~72 to 100 dpi
Multiple screens	no	no	yes
Audio input	telephone handset, microphone	microphone	microphone
Audio output	telephone handset, perhaps speaker	TV speakers up to full surround sound	computer speakers to high-end speakers
Data bandwidth	28.8 Kbps to ISDN	28.8 KBps to cable modem throughput	28.8 Kbps to T1
Printer connection	optional	optional	yes

have less memory (both storage and run-time), smaller displays, potentially less powerful processors, and different input and output devices. Especially in the case of input devices, information appliances tend to use lower-bandwidth mechanisms, which limit the richness of users' inputs. These factors place constraints on the human interface that do not normally exist on a desktop computer. So, ironically, the very factors that are deemed beneficial and essential to the product experience pose hurdles for product designers.

.2 Design Considerations

The design considerations described in this section are provided as an approach to the design of information appliances enhanced with computing technology. These considerations are not rules or prescriptions for design; rather, they represent a design approach that is quite different from that of traditional PCs and workstations. The five design considerations included here are not an exhaustive list but provide a strong notion of the new approach needed for this design space:

1. Account for the target domain
2. Dedicated devices mean dedicated user interfaces
3. Allocate functions appropriately
4. Simplify
5. Design for responsiveness

Account for the Target Domain

The user interface of an information appliance must suit the application domain for that appliance. A "domain" is characterized by the environment, the applications or tasks, and how the device is likely to be used. The device is intended to be dedicated to a set of very specific and related tasks in a defined setting. Both hardware and software are optimized for this domain.

A basic pager is a clear example of a dedicated information appliance. The pager's tasks are to receive pages and display the number of the caller. The device is used in very short sessions and can be carried to any environment. The user must be able to retrieve necessary information—either a phone number or a short message—quickly and efficiently. Clearly, the domain of use for the pager has considerable design implications for the

user interface. There is a wide range of possible domains for consumer devices. The three domains presented below cover the most relevant. Typically, an information appliance may fall into more than one of these domains. These categories are not mutually exclusive. However, the devices will tend to be used "more commonly" in one of these domain types; the user interface should be biased accordingly.

Entertainment

Information appliances used for entertainment, such as a TV-based Web browser or a game-playing device, are characterized by the following:

- Typically, these appliances will be used in a less "directed" setting compared to a PC. That is, the user is approaching the appliance with a more relaxed attitude and is more interested in a pleasant experience than in performing a specific task, such as writing a paper, completing a spreadsheet, or sending email. A game-playing device is an example of an appliance used more for fun and compelling content than for accomplishing a task.
- Use is characterized by long periods of interaction with the device (more than 30 minutes).
- Tasks are generally less structured than those performed on PCs. The user may not know exactly what sort of entertainment is of interest and may take some time to find something of interest.
- Users are likely to have varying levels of concentration and attention during the interaction. The tasks are such that interruptions (from the environment or even the content itself) are possible.
- Generally, interaction occurs in a relaxed, comfortable, and low-stress environment such as the living room. The relaxed nature of the interaction characterizes both the setting and the user's attitude.

These characteristics have several design implications for the user interface:

- Interaction with the content is more important than any other factor. Devote as much input/output bandwidth as possible to viewing, hearing, and interacting with the content.
- The human interface for the device (e.g., browser controls) should interfere as little as possible with the content.
- Task completion time is not a critical factor; the user is not under significant time pressure. Therefore, a pleasant experience is always preferred to an efficient one.

- Note: efficiency is a positive attribute of any system and should be sought after in user interface design as well. However, in the entertainment domain a pleasant and compelling experience is as important as efficiency. Above all, predictability is more important than efficiency.

Information Access and Communication

Consumer devices used for information access and communication, such as an Internet-enabled screen phone or PDA, are quite different from those of the entertainment domain. In particular, users of these devices will concentrate on getting a specific task accomplished quickly, such as creating a calendar appointment or finding a phone number. In fact, usage of these devices may be on a "one task at a time" basis; that is, the user may go to the device (or turn it on) just to perform quick information access or a communication task. Devices in this domain are characterized by the following:

- Users of these devices will be directed and goal oriented. The user is interested in performing a specific task rather than using the device for entertainment. For instance, the user of an Internet screen phone must be able to find the phone number of a restaurant, call, and make reservations.
- Interaction with the device is typically short (less than 10 minutes).
- Users are likely to concentrate on the task while using such devices. Interruptions are still possible, but not as likely as in the entertainment domain. Also, interruptions are more likely to be task relevant, such as receiving a phone call while trying to look up an address book entry.
- Tasks are typically structured and directed. As opposed to interaction with entertainment-oriented devices, users will be motivated to complete a specific task. For instance, the user may need to send an email to his brother to ask about a party. That is the main purpose of the interaction with the device; the user wants the experience to be quick, efficient, and easy.

The design implications for the information access and communication domain include the following:

- Ease of learning and long-term ease of use are more important than any other factor because the user must be able to get tasks done quickly and efficiently.
- Task completion time may be a critical factor. While the users are not concerned about a few extra seconds in performing a task, they are also not interested in spending much time to, say, send a simple email. Tasks should be designed to be accomplished efficiently.

TABLE 2.2

Characteristics of each target domain and the associated design implications.

Domain	*Characteristics*	*Example Design Implications*
Entertainment	1. Long interactions (> 30 minutes) 2. Less structured interaction (versus PC) 3. Not very "directed" tasks 4. More relaxed interaction 5. Various levels of concentration	1. Content is critical so devote significant I/O bandwidth to content. 2. Pleasant experience preferred to efficient one.
Information access and communication	1. Short interactions (< 10 minutes) 2. Structured interaction (versus entertainment) 3. Usually "directed" tasks 4. Various levels of concentration	1. Ease of learning and long-term use are critical. 2. Efficiency can be a key feature of the UI.
Assistant devices	Similar to information access and communication	Similar to information access and communication

Assistant Devices

Assistant devices are appliances dedicated to helping a user perform very specific tasks. Examples include

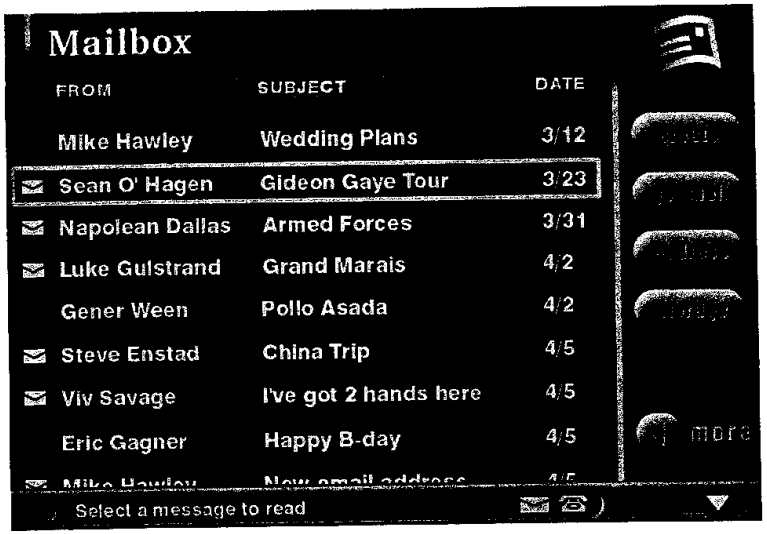
- A handheld tour guide device for museum patrons
- A device for a waiter to take orders
- A device to help an assembly line supervisor debug a problem
- A device to help delivery personnel collect information on the delivery/pickup schedules, plan the best route from one location to another, and so on

For the most part, the characteristics and design implications of the assistant devices domain are similar to those of the information access and communication domain (see Table 2.2).

Dedicated Devices Mean Dedicated User Interfaces

Consumer products use input and output methods optimized for particular uses. A PDA might use a pen to allow handwriting recognition and touch input. A screen phone may have a keyboard for text input, while a cellular phone may use the number pad. Additionally, the functions available on the

URE 2.1



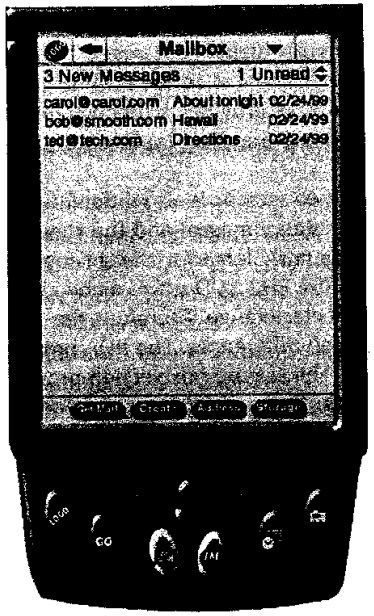
A sample inbox screen from an email application running on a TV set-top box.

devices are tailored to the needs of each device. TVs may have a dedicated electronic program guide (EPG), and a PDA may have an address book and calendar.

The market focus, user tasks, and product requirements for a consumer product should drive the user interface design for the product. This results in a user interface that is cleanly integrated with the applications, I/O, and other characteristics of the system. This tight focus and high level of integration work to the advantage of the user in consumer products because all components have been designed to work well together.

The tight focus also means the user interface may vary from one product to another. This variation can be true even for the same software application deployed on different hardware platforms. For example, consider an email application that must run on a TV set-top box and a PDA. In the case of the set-top box, the application will appear on a color TV screen (at roughly 640 × 444 resolution for an NTSC signal), and the user interacts with the product using a remote control. For the PDA, the application will appear on an integrated 160 × 120 monochrome LCD screen. The user interacts with the product using a touch overlay. Clearly, the display size and color differences, as

FIGURE 2.2



An inbox screen on a PDA device. This screen is functionally identical to that shown in Figure 2.1, but the user interface is altered to meet the requirements of the device.

well as the differences in input mechanisms, will have significant impact on the user interfaces for the two products. Figure 2.1 (see also Color Plate) shows an email inbox screen on a TV set-top box. Figure 2.2 (see also Color Plate) shows a functionally identical screen on a PDA device. While the two figures are similar in size for the purposes of presentation, note that in actuality the screen in Figure 2.1 appears on a large color television screen, while the screen in Figure 2.2 appears on a small handheld device with a much smaller screen.

A review of the two screens illustrates some of the contrasts in designing the same screen for these two very different devices. Notice the difference in layout, in particular the difference in the placement of command buttons (e.g., Get Mail, Create). For TV usage, buttons must be on the side to allow easy access with a remote control. With this design users must move the

highlighter (shown in Figure 2.1 on the message from Sean O'Hagen) sequentially from one target to the next; on this screen the targets are basically the email messages or command buttons. Highlighter navigation is done using a remote control that allows the highlighter to be moved on the screen and to select the currently highlighted target. Navigating to a target is a major constraint in this environment. In the PDA device, access to targets is not an issue because the PDA provides a touch screen, allowing touch-based access to any command (random input as opposed to sequential input).

Relative text size for the two screens is also different. Given the poor display quality of TV CRTs (for static images) and the viewing distance for TV usage, the text size must be fairly large for the set-top box version of the screen. On the other hand, the PDA LCD offers a crisp image, and the user will be viewing the screen at close range. Therefore, the text can be smaller.

Note also the visual design differences. The PDA provides a streamlined and "business-like" design. The set-top box version provides more whimsical and decorative elements. These differences are due to the set-top box being an "entertainment" device, while the PDA is more of an "access and communication" device. Textual elements on the PDA can be smaller, but the horizontal space on the screen must be used carefully because there is a limited amount. However, on the TV screen, horizontal space is plentiful by comparison.

While the underlying email application may be the same in both devices (perhaps exactly the same code), the user interface is dedicated and optimized for the requirements of each product. There is a long-standing approach to separating the user interface from the underlying computational side of software products. This approach, known as the model-view approach, contends that the "view" (or UI) can be divorced from the data or "model" component of an application. With this approach different views can be used with the same data set (for further discussion of this approach, see Buschman et al. 1996; Object Arts 1999).

Of course, the notion of a dedicated UI, based on the device, may violate some sensibilities regarding consistency of design. Some designers may be concerned that a dedicated UI, not shared across devices, undermines ease of learning. However, it is not practical to force a consistent user interface across devices with radically different hardware and usage characteristics. Our usability studies have shown that predictability is far more important than consistency or efficiency. The notion here is that the product should just "do the right thing." Imagine using a physical keyboard on a screen phone to create an email. This task is similar to doing the same task on a desktop computer with a keyboard. Now imagine performing the same task on a PDA. Entering text becomes a more tedious process. There are alternate

solutions, such as handwriting recognition, that can be used to make the process of entering text easier. But handwriting recognition (gesture-based input) doesn't have much in common with tapping a key on a keyboard. A different user interface approach is required to manage the task on the PDA, which results in a different yet better-suited user interface.

Allocate Functions Appropriately

Being a generic computing platform, the desktop PC must offer standard I/O mechanisms. Therefore, a CRT, a standard QWERTY keyboard with function keys (F1, F2, etc.), and a mouse are the typical I/O devices. Software developers usually design their products to work with these standard PC mechanisms.

However, information appliances offer the opportunity to depart from the standard I/O of a desktop PC. Information appliances provide dedicated user interfaces tuned to the needs of the device and its users (see previous section). This dedicated interface allows product designers to build in the I/O mechanisms that best meet the requirements of the device and to create software that is well matched. Therefore, there is significant freedom as to which features will be supported in hardware, in software, or in both.

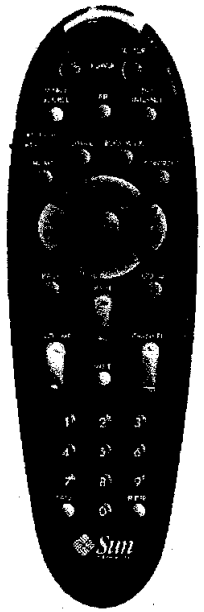
This newfound freedom places a special burden on the design. Care must be taken to ensure that features and functions are appropriately assigned (allocated) to hardware versus software. A good example for discussing function allocation is the function of launching applications. Generally, a PC allows users to launch applications using the desktop (typically by double-clicking an icon) or entering the "executable" name. This approach is sensible since the PC is designed to handle all sorts of applications. The PC manufacturer and providers of system software make no specific assumptions about how the device will be used or what applications will be needed.

On the other hand, PDA device designers can make assumptions about the applications used on the device and can support those assumptions in the design. For example, buttons can be included in the hardware to provide one-touch application launching. In this example, launching applications has been allocated to hardware. This decision is justifiable because the number of applications is limited and users need quick access to them.

While the above example is somewhat oversimplified, it is an important approach in designing information appliances. The design freedom exists to allocate functions to best meet the needs of the device and its users. These design decisions have considerable impact on the user interface.

Two of the key factors in determining function allocation are usage frequency and importance.

FIGURE 2.3



An example of a remote control with a number of available functions. These functions are provided either instead of being on the screen or in addition to what is available on the screen.

Frequency of Use

Tasks that users perform frequently should be assigned to very visible and easily accessible locations in the user interface; this approach applies for both hardware and software. The "location" may be different depending on the type of appliance. For example, in designing a TV set-top box user interface, commands with high frequency of use may need to be represented with physical buttons on the remote control (or IR keyboard). Figure 2.3 provides an example of a set-top box remote control layout. While many of the functions appear on the TV screen, there are certain functions that must be available on the physical remote control as well. For instance, one-touch access to a TV picture-in-picture (PIP) window is essential. Users will frequently access and dismiss the PIP window (for TV viewing) and should be able to

quickly do so at any time. Associated with PIP is switching between full TV viewing and Internet functions. Users will switch between these two modes and should have easy access to this feature.

Importance of Use

Important or critical features must also be made highly available to the user. Importance and frequency of use are somewhat related, but they should be considered separately. There may be functions that are not performed frequently but are quite critical to successful use of the product. A classic example of this distinction is the installation of devices (after first-time purchase). Users of information appliances must be able to install and set up the device easily. For instance, a consumer who returns home with a new Internet-enabled screen phone should have no trouble installing the unit and connecting it to the Internet. Significant design effort must be applied to ensure the process is simple, which often entails making certain features or functions highly visible and available. Typically, users will not perform the installation task frequently (perhaps once or twice total). But successful completion of this task is extremely important for such devices.

Simplify

Simplifying user interfaces has been a prime objective for most designers. However, despite repeated attempts, the industry has been largely unsuccessful in keeping PCs and workstations simple. There are three main reasons for this increasing complexity in products:

1. Business model
2. Continual obsolescence
3. Implementation-driven design

The basic business approach to selling computing equipment and software is to offer more features, better performance, and more quality (stability, user experience, support, etc.) at a lower price than the previous product (or the competition's product). While this is an oversimplification of how the industry sells and markets, it is generally true. The result of this business approach is that products require progressively more complex user interfaces to support the additional features.

Tightly coupled with the business model is the relentless effect of continual obsolescence as a result of Moore's law—every 18 months computing

prowess can be doubled. This phenomenon has driven the hardware and software industries to continually "improve" the technology. Improvement means the ability to add features and functionality that the previous technologies were unable to offer. Accordingly, older versions of products quickly become obsolete and ill equipped to handle the new product offerings and features.

Lastly, in the computing industry, there is a tendency toward implementation-driven design. This tendency is derived from a technology-centered prespective, where developers may look for a problem to solve with a collection of technologies. Often this approach leads to an unnecessarily complex product. Of course, a user-centered approach would attack the problem by first defining what users need. At a later stage the proper technologies would be identified to address the product design challenges.

These three biases must be avoided in building consumer devices. Simplification is key. There are two important dimensions to consider when trying to make products simple (or decrease complexity): the functionality versus simplicity trade-off, and the choice versus simplicity trade-off.

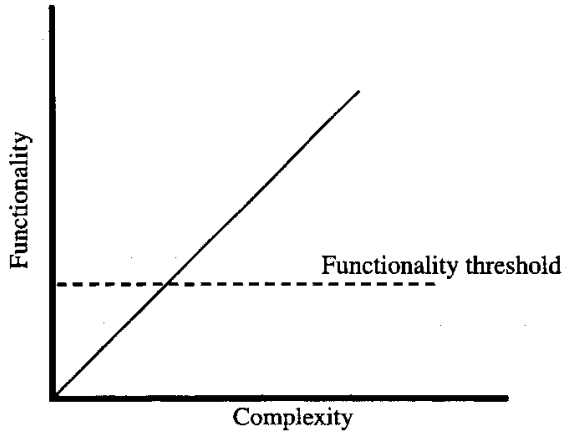
Functionality versus Simplicity and the Functionality Threshold

A long-standing truth in the industry is that the more functionality included in a product, the more complex it becomes (Figure 2.4). In designing information appliances, it is critical to reduce functionality to the most essential/needed functions. Consumer devices offer a unique opportunity to justifiably eliminate unneeded functionality because these devices will be targeted for a limited set of tasks and will have dedicated user interfaces. A good way to consider this trade-off in the interface design is to think of the 80/20 rule. For each application or feature set, it's helpful to identify the 20% of functions that will meet 80% of the users' task needs. Those are the functions to support in the product and around which to optimize the design. The remaining 80% of functions are proposed based on other criteria (utility, necessity, competitive edge, price, etc.), and it may be necessary to exclude most of these functions.

A second key part of this approach is to optimize the user interface around the absolutely key features in that 20% of functions in the product. User interface decisions are predicated on making that 20% of functions as accessible as possible. The design should not be driven by providing equal access to all functions. Instead designers should "relegate" some features to a secondary role in order to make the primary functions as accessible and easy to use as possible.

Clearly, gratuitous elimination of functions is not useful either. Removing too much functionality can lead to a product that is either too limited to support users' tasks or simply uncompelling. Therefore, it's important to provide

FIGURE 2.4



The functionality versus complexity trade-off. As more functionality is added to a product, the level of user interface complexity also rises. The "functionality threshold" is the concept that information appliances should limit the functionality to the essential few (the threshold) that provide for a compelling product without leading to unmanageable complexity.

enough functionality to have a compelling and useful product without complicating the user experience. We refer to that "sweet spot" as the functionality threshold: It is just the right collection and number of features to strike the balance between functionality and simplicity.

Choice versus Simplicity

As with functionality, a trade-off exists between simplicity and the number of choices available to a user. A simple example of this concept is the menu at the local restaurant. Say you visit the local restaurant with the intention of ordering a hamburger. In the "old" days you'd have a choice between a Hamburger, a Cheeseburger, and perhaps a SuperBurger, which was basically a larger version of the Hamburger. But in today's choice-filled restaurant you may be faced with choosing from a Hamburger, Cheeseburger, the SuperBurger, the AvocadoBurger, the Special WesternBurger, the Chef's ExtraBurger, the SlapJack Burger, the Big Daddy Burger, and the "Bring Down

RE 2.5

An example of a set-top box appliance where the user is browsing the Web and viewing television in a PIP window. The user can access or dismiss the PIP window by pressing a button on the remote control.

the House" Burger! Too much choice and a complicating, perhaps overwhelming, situation.

Choice is not an inherently negative construct. Choice is useful when it is appropriate, but information appliances can benefit from reduced choice. This approach is contrary to traditional design practices in the industry, where it is the norm to provide users with choices on everything from their desktop color scheme to the "toolbars" they want in the UI. Given the reduced functionality, targeted user population, and focused nature of these devices, reasonable defaults can be provided and nonessential options removed completely. An example of reasonable choice reduction is found in the TV set-top box picture-in-picture window. Most Internet set-top boxes allow users to watch TV in a PIP window while performing standard Internet functions, such as browsing the Web or reading email (see Figure 2.5; see also Color Plate). However, this PIP window is limited to being accessed and put away—that's it! Users do not have the "choice" to move the PIP window, select its size, or change any of its characteristics. Some set-top boxes provide slightly more choice by allowing the user to select a quadrant of the

screen where the PIP window should appear. However, the user is still limited to only one of four quadrants as opposed to positioning the window anywhere on the screen. This approach, while limited, meets the needs of the overwhelming majority of users. Simplification is achieved by removing nonessential choices from the UI. While some users may find the lack of choice limiting, the design bias is toward simplicity and supporting the most common scenarios.

Reducing choice is an approach to yield simple designs. It is more of a strategy than a rule. Clearly, there are functions and features for which users should have choices. However, we believe in limiting choices to only the essential few to reduce the complexity of the design.

Design for Responsiveness

An interesting element of designing for the consumer space is that users expect immediate and reliable responses to their inputs. Certainly, responsiveness is a positive quality of any system and is a goal in the PC domain as well. However, for the consumer audience responsiveness takes on characteristics that are unconventional from a PC perspective. The two most critical aspects of responsiveness that information appliances should support are interruptability and continuous feedback.

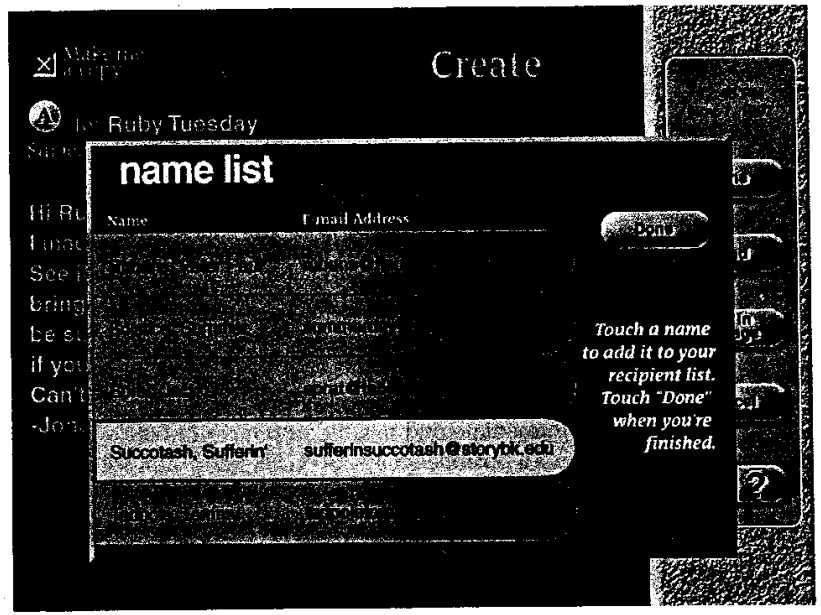
Interruptability

In an information appliance, almost all operations should be immediately interruptible. Users have built expectations of behavior through years of using other consumer devices, such as TVs and stereo equipment. Additionally, it's simply good design to allow the user to be in control.

Consider a PC application that has posted a dialog box to the screen asking for confirmation on a command (e.g., delete). Generally, in PCs, the user must deal with the confirmation dialog before doing anything else. This modal approach is definitely true of major system functions like shutting off the PC; the user must deal with the lingering confirmation dialog before the shutdown can proceed. On a set-top box, however, the approach should be different. With few exceptions, designers should not require users to deal with modal dialogs. If the user wants to turn off the set-top box, she should simply press the "Power" button to turn it off. The lingering dialogs or commands must be handled gracefully by the system.

Dialog boxes on a screen phone provide another example (Figure 2.6; see also Color Plate). Most screen phones have a touch screen, allowing users to make inputs with their fingers or a stylus. Consider a situation where a dialog box has been posted to the screen asking the user for confirmation on an action or simply providing information. A button exists inside the dialog box that the user can touch to dismiss the dialog; this user interface is essentially identical to that of a PC. However, in a touch environment (and with information appliances) it is best to allow for additional mechanisms for interrupting or dismissing the dialog. For instance, touching anywhere outside the dialog can be treated as a dismiss command. This approach provides a solution that presents the product as responsive. Certainly, there may be situations where the user did not intend to dismiss the dialog. However, the product behavior is biased toward the most common cases.

GURE 2.6



Sample screen from an Internet-enabled screen phone. The screen is an email composition screen, and a dialog box has appeared over the top (based on the user's input). With a touch screen device, it is possible for the user to dismiss this dialog by pressing the "Done" button in the dialog or by simply touching anywhere outside the dialog.

Continuous Feedback

Information appliances should respond continuously so users are confident that their inputs are being processed. However, with these devices, feedback

takes on added meaning beyond the typical audiovisual feedback for user inputs and becomes more task related.

Users should have feedback and progress indication even while the product is functioning normally (or as designed). For instance, in usability testing a screen phone, we found that users needed feedback that mail had been sent successfully. Clearly, receiving this type of feedback on a desktop system would be annoying, but computer-novice screen phone users seemed lost without it. Therefore, users of these devices need continuous, unobtrusive feedback about their progress with the device, even when activities are going well!

Of course, many of the feedback principles from the PC domain apply in the information appliance arena as well. For instance, where possible, delays should always be minimized. It is important to provide appropriate feedback in cases where completion of an operation is delayed, for example, when downloading information. Displaying an animated indicator, preferably one that shows how the operation is progressing, is crucial; a static status message is not adequate for such situations. Providing dynamic, live feedback to the user during any operational delays keeps the user engaged and ensures they don't mistakenly perceive the system to be broken. Anecdotal evidence shows that people will perceive a product to be faster if it continues to show activity as opposed to not providing feedback until the operation is done.

Responsiveness starts with feedback but goes further, to "intelligent" responses. For activities that users perform frequently, the product should respond in a way that makes the experience easier for the user. An example of this intelligent responsiveness can be seen in some car navigation systems. When the user starts to enter an address to receive directions, the system turns off the keys on the on-screen keyboard that do not map to real addresses. This helps the user be more accurate in entering addresses and makes the task quicker to complete.

Generally speaking, if products are not perceived to be responsive, users can become annoyed with the product, often repeatedly pressing buttons, and/or assuming the device is broken. When users become frustrated, they stop using a device, or worse, return it to the store.

Summary

The five design considerations discussed above provide a good overview of the design approach we have found successful for information appliances. Some of the considerations are novel, such as the new approach to simplicity. Others are a new twist on a familiar UI design tenet, such as responsiveness. All of the design considerations are extremely helpful in approaching the practical design of information appliances.

2.3 A Proliferation of Wildly Different Interfaces?

If information appliances are successful and ubiquitous, as many predict they will be, then an interesting question arises: Will the proliferation of appliances lead to a proliferation of wildly different user interfaces?

The answer is: probably not. With any new technology/product, new UIs are inevitable. Take a simple example like a wine bottle opener. Initially, there were a variety of tools available for uncorking wine bottles. However, over time users of these devices settled on the one or two interfaces that work best. There is one corkscrew that seems to be the overwhelming choice of waiters and waitresses the world over, despite the variety of available alternatives. One corkscrew mechanism has emerged as the leader because it offers the best usability while meeting all the necessary requirements.

A similar "settling of the dust" will occur with information appliances. Initially, a variety of user interfaces and features will be available on a multitude of devices. However, over time, many of the devices will fail. Of the devices that succeed, only a limited number of user interfaces will remain viable for each device. For any given class of device, a particular approach will be accepted or followed as a de facto standard.

Additionally, there are certain user interface mechanisms that have proven to be successful across different devices and products. For instance, selecting items from lists, pressing buttons, and progress indicators are UI elements that seem to work well universally. There is no logical reason to reinvent mechanisms that users understand and can easily use in different devices. Therefore, these items will likely become standard parts of most cross-device user interfaces.

Note that this process of de facto standardization does not necessarily guarantee a usable or high-quality user interface. In the past, poor interfaces or user interface environments have been accepted as "standard." Hypertext Markup Language (HTML) is a perfect example. HTML became popular and an accepted standard in the early 1990s despite the fact that it did not support many of the rich features available in the user interface technologies of the time. But HTML did offer one very compelling feature: cross-platform access to the same content. Therefore, this one compelling feature drove standardization of content around HTML despite its shortcomings.

Some standardization is already taking place in the information appliance design space. For instance, consider the input mechanisms in TV-based user

interfaces, such as those for Internet-enabled set-top boxes. Early versions of set-top boxes provided either a pointer (cursor) or a rectangular highlighter on the screen. The user would then use arrow keys on a remote control or another physical input device to manipulate the on-screen element. The pointer behaved much like a pointer on a PC. The user could move the pointer to any location on the screen (using the input device) and press a button to make a selection (click). Conversely, the rectangular highlighter moved in discrete steps from one "hot spot" (or target) to the next. While the pointer approach provided more flexibility, the highlighter solution has proven easier to learn and use for consumers without much computer experience. This fact is especially true given that arrow keys are used to manipulate the element on the screen. Users found the highlighter to be extremely simple to manipulate (using remote control arrow keys), and its behavior highly predictable—not necessarily efficient, but very predictable. Most TV-based user interfaces now use some version of a discrete, step-by-step input as opposed to a random input (like that of a pointer on a PC). Therefore, while many different TV-based products have been produced recently, their basic approach to navigating a screen is not substantially different.

Clearly, as content accessed and used on TVs becomes richer and more interactive, the input mechanisms may need to be enhanced as well. For instance, the highlighter approach may be too limited for the day when all the neighborhood kids are playing networked interactive games on their television sets. However, with the types of content and bandwidth currently available on set-top boxes (from Internet to satellite boxes), the highlighter approach is practical.

2.4 General versus Specific Trade-off

As mentioned in the section on simplification, designers of information appliances must strike a tenuous balance between offering a product that is simple enough to use, but functional enough to be useful. This balance or "functionality threshold" may be different for various classes of users even with the same device. The benefit of information appliances is in their dedication to a narrow range of tasks. However, picking that range of tasks and providing the appropriate compelling solution is difficult. There are two potential disadvantages in building devices that are too specific:

- The devices may be too limited to be useful or compelling.
- Assuming the devices are useful, users may need to own many such devices for various domains (collection of tasks), which may be unwieldy.

On the other hand, making a device overly general will have the following disadvantages:

- It can lead to a product that is too complex.
- It may yield a product that doesn't appropriately meet the task needs of users.

Consider the example of a personal digital assistant. Generally, PDAs include applications such as an address book, "to do" list, calendar, and notes, among a handful of others. This collection of features/applications is sufficient for many users. But there are users who would benefit from having a telephone integrated into the PDA to allow for real-time voice communication. Additionally, there is a collection of users who would like email.

Now the originally simple PDA has taken on new telephony and email functions. Certainly, each of these in isolation is not a complicated application, but once they are brought together on the same platform a number of complexities are introduced. For instance, the issues associated with managing a wireless connection to collect and send email can have a significant impact on the complexity of the user experience. Also, such a complex product will no longer meet the more practical needs of the users who needed only the original PDA functionality. Of course, in practice, product lines are created to provide different versions of a product to meet the needs of a diverse user base.

Interestingly, this issue of general versus specific can be seen in the "traditional" appliances as well. In some cases appliances become too complex, and in other cases the functionality is inappropriate. For instance, the only purpose of a microwave oven is to heat food. However, there are various models of microwave oven, some of which provide a number of features and functions. These high-end microwave ovens can become so complex that users can't easily determine how to simply heat a dishful of food. Moreover, most users end up drawing on only a limited subset of the myriad functions.

Or consider the case of the combination VCR/TV products. This product was envisioned as a convenient combination of two solid consumer electronics products. However, the VCR/TV has found limited success, largely because, to keep cost and size down, the TVs in the product are not high quality enough to be used regularly and the VCR can't easily be connected to other TVs. Therefore, this more feature-rich product doesn't meet the real needs of users.

Some amount of consolidation of information appliances will occur naturally. As the market for these devices matures, product designers will have a better understanding of the collection of tasks people prefer to perform on, say, a TV as opposed to a screen phone. It will also become more clear as to which types of tasks need to be performed together and which collections of

functions and features best support these tasks. However, care should be taken not to create hybrid devices that do too much and are so complex that they are useless to the consumer.

2.5 Conclusions and Implications

There is a third industry that has not been addressed in this chapter: service and content providers. Much of what users want to do with these devices (and therefore their tasks) will be determined by the type of content available on a device. The technology and user interface for information appliances will be driven by the nature of the services and content. It is important to view an information appliance as the work of a triumvirate: manufacturers, technology providers, and content/service providers. In some cases one company may perform the role of two of the parties, but the key components are still the same. To scope these discussions, we have taken a largely device-based view of the user interface challenges. However, content and services will also be factors in this space.

Information appliances provide an opportunity for new, ground-breaking user interface work. The application of computing technology and the traditional notion of appliances to small consumer devices are an exciting challenge for both the computer and consumer electronics industries. Clearly, most of the user-centered design approaches and tools that have matured over the past 20 years of research and practice still apply in this domain. However, information appliances will challenge two very different industries to converge in many areas. User interface and usability will be key components for the products and services resulting from this convergence.

2.6 References

AllNetDevices. 1999. Latest net device news and analysis [online]. Available: *www.allnetdevices.com* (accessed June 7, 1999).

Buschman, F., R. Meunier, H. Rohnert, P. Sommerland, and M. Stal. 1996. *Pattern-Oriented Software Architecture: A System of Patterns.* Boston: John Wiley and Sons.

Computer Industry Almanac, Inc. 1998. Nearly 600 million computers-in-use in year 2000 [online]. Available: *www.c-i-a.com/19981103.htm* (accessed June 7, 1999).

Object Arts, Ltd. 1999. Model view controller [online]. Available: *www.object-arts.com/EducationCentre/Overviews/MVC.htm* (accessed June 8, 1999).