pigment—that is, red dyes tend more toward blue tones. The quality of the water used influences the dye as well, especially if chlorine, acids, alkalines, or heavy minerals are present. A weight ratio of 1:1 of mushroom to fiber is recommended at the start. Some fungi have a lot of pigment, whereas others have very little. Thus, with experimentation, careful observation, and note taking, one can decide the best proportions on a case-by-case basis.

Since the First International Mushroom Dyes Textile Show—FUNGI and FIBERS was held in Mendocino in the summer of 1980, interest in mushroom dyes has continued to spread worldwide rapidly, with particular enthusiasm from the Scandinavian countries, Scotland, and Western Australia. In 1985, the International Mushroom Dye Institute (IMDI) was established. The nonprofit research institute was founded to encourage the use of fungal pigments, to further research on their extraction and employment, to encourage research on cultivation of especially desirable fungi, and to provide financial aid to artists and researchers to enable them to participate in the international mushroom dye symposia and exhibitions.

Other developments. In the 1980s, Miriam Rice explored the possibility of making paper out of the fungal detritus that is left over from the dyes. Always a passionate advocate of recycling, she saw this as the natural solution for disposal of the fungal residue from the dye process. She tried many mushrooms and polypores (shelf fungi typically lacking stalks), finding that the polypores that contain chitin as well as cellulose produced the best paper—all colors and textures, and with no additives necessary. Thus, in 1985, the concept of papermaking from fungi was introduced.

Further experiments in the 1990s led in another direction: specifically, making watercolor paints from many of the mushrooms that had also been used for dye. By extracting the fungal pigments and combining them with a variety of media, a variety of watercolor paints have been generated (Fig. 5).

In addition, recent research into developing the mushroom pigments into some form of medium for artists to use in drawing and sketching to supplement the watercolor paints has resulted in a drawing medium called “Mycostix™.” This medium uses pigments extracted from fungi and combines them with a variety of binders to form crayons that can be used as drawing, watercolor, pastel, and encaustic (hot wax) media.

For background information see Color; Dye; Dyeing; Fungal Ecology; Fungi; Mordant; Mushrooms; Paper; Pigment (Material); Printing; Wool in the McGraw-Hill Encyclopedia of Science & Technology.

Dorothy M. Beebee


Human-computer interface design

Advances in computational technology have made computerized devices an integral part of our daily lives. Unfortunately, poorly designed interfaces often make these devices difficult and awkward to use. A conceptual framework for the design of effective human-computer interfaces will be outlined.

Cognitive Systems Engineering (CSE). CSE provides a systematic approach to interface design. The interface is considered a form of decision support: it is used by a person with the goal of completing a task within a work domain. Thus, there are three system components (domain, user, and interface) that contribute a set of mutually interacting “constraints” (Fig. 1). How well these components “fit together” will determine the case of use and the effectiveness of the device.

Domain constraints. Because the interface is viewed as decision support, the logical starting point for interface design is an analysis of the constraints in the work domain. The unfolding events in “law-driven” domains such as process control (for example, a power plant) arise from the physical structure and functionality of the system. In the case of a power plant, the laws of thermodynamics are a fundamental source of regularity. At the opposite end of the
spectrum are "intent-driven" domains, where the unfolding events arise from the user's intentions, goals, and needs (for example, information retrieval).

**User constraints.** The capabilities and limitations of users constitute another set of constraints. For example, humans possess extremely powerful skills for perception (obtaining visual information) and action (manipulating objects), but they have extreme limitations in terms of attention and memory (only a few items can be maintained in working memory). Additional constraints are introduced by those who will use the device. Power-plant operators are highly trained and will have homogeneous (uniform) skill sets; this is in contrast to mobile cell phone users, who do not require this level of training and skill.

**Interface constraints.** A third set of constraints is introduced by the interface. In the early years of computer technology, the interface was designed using an ineffective "conversation" design metaphor: cryptic verbal instructions were communicated through an intermediary (that is, a command line) to the computer. This imposed a severe set of constraints on interaction. Interface technology has evolved considerably since then (high-resolution screens, bit-mapped graphics, precise pointing devices, and so on). This has allowed the use of design metaphors that are predominantly spatial in nature and are far more effective (for example, the desktop metaphor).

**Principles of design.** The fundamental goal of interface design is to build virtual ecologies of work domains that allow the powerful perception-action skills of the human to be leveraged (in other words, that do not require the use of limited-capacity resources such as working memory). The ease of use and effectiveness of a device will ultimately depend upon the relationships (the "fitting together") between these three sets of constraints (Fig. 1). Three principles of interface design can be used to help these three system components fit together well.

**Direct perception.** The interface should allow the user to comprehend the current (and perhaps the future) state of the system through powerful visual processes. The first step is to conduct work domain analyses to reveal the domain constraints (physical, functional, and goal-related properties). Effective graphical representations (the virtual ecology) then need to be developed. The visual properties of these representations must match the perceptual and cognitive abilities of the observer (for example, is the user sensitive to variations in the visual features that were chosen?). They must also match the domain constraints (for example, do variations in the visual features accurately reflect variations in, or properties of, the domain?).

**Direct manipulation.** The interface should allow the user to execute control input through powerful action capabilities applied directly to objects in the interface. A good example is file deletion in the Macintosh OS X operating system. An object in the work domain (a computer file) is represented in the interface (its icon). The user manipulates this representation directly (that is, point, click, drag, drop) to delete it. In contrast, the deletion of a file via a command-line interface (for example, "rm klunkyinterfaces.doc," where "rm" stands for "remove") or even a pull-down menu is not direct manipulation.

**Perception-action loop.** There is a potentially symbiotic relationship between direct perception and direct manipulation. The objects in the interface can be designed to support the powerful human skills of both perception and action simultaneously. When this occurs, the display interface and the control interface are merged into one; the perception-action loop (the coupling between a human and the world) is intact.

**Examples of interface design.** The application of these three general principles will require different design strategies for different work domains. Two examples will be provided.

**Law-driven domain.** The most effective interface design strategy for law-driven domains is to develop spatial analogs (that is, abstract geometrical forms) that directly reflect the underlying domain constraints (Fig. 2). This display combines over 100 individual sensor values into an octagonal form. The octagon is perfectly symmetrical when power-plant conditions are normal (Fig. 2a). Conversely, asymmetrical distortions provide visual evidence that there is a fault (Fig. 2b). The operators learn to associate particular patterns of distortion in the display with particular types of faults. Direct perception has been achieved: the operator can literally see the state of the plant directly, without the need to gather relevant information and perform complicated mental calculations. (Figure 2c illustrates the display in an actual power plant.)

An optimal implementation of direct manipulation in this interface would involve high-level control: the operator would have the capability to point, click, and drag the vertices of the octagon to a desired location, thereby changing the underlying variables. Unfortunately, this is not possible because of the complexity, interconnectedness, and potentially conflicting goals that characterize a power plant. Direct manipulation must therefore be implemented at a lower level. Thus, to change the setting of an individual variable, the user might select a visual indicator and drag it to a new location on a graphical scale (rather than typing it in).

**Intent-driven domain.** The most effective design strategy for intent-driven domains is to develop spatial metaphors (for example, the desktop metaphor on computers) that relate the requirements for interaction to more familiar objects and activities. The Apple iPhone provides an excellent example.

Direct perception in the iPhone's interface is achieved primarily through static spatial metaphors. These metaphors use physical similarity or symbolic convention (a stylized phone icon) to provide a semantic link between the user's intentions (the need to make a phone call) and the underlying functionality of the iPhone (cell phone). Thus, this design strategy uses graphical representations to leverage preexisting concepts and knowledge, thereby
Fig. 2. An abstract geometrical form display used to represent dynamic values of system variables for a process control domain. (a) The octagon form in a perfectly symmetrical shape represents the values of variables under normal power-plant conditions. (b) The nonsymmetrical distortion of the octagon form characterizes a particular abnormality in the plant (that is, a loss-of-coolant accident). (c) The geometrical form display as it appears in an actual power plant. (Line drawings from W. F. Jenseefer et al., Generating an integrated graphic display of the safety status of a complex process plant, United States Patent no. 4,675,147, June 23, 1987; photo courtesy of David D. Woods, Cognitive Systems Engineering Laboratory)

Assisting casual users in understanding and using the iPhone.

It is less obvious that the iPhone uses an overarching spatial metaphor to facilitate navigation among the various applications and modes. This metaphor includes an array of icons (representing applications) in a "home" mode, a row of icons at the bottom of an application interface (representing modes of the application), and a dedicated mechanical button at the bottom of the phone (return to home mode). The spatial dedication of these interface components makes navigation of the iPhone much like navigation in the real world: getting to different applications and modes in the interface is somewhat similar to navigating between rooms in a well-known building.

Direct manipulation in the iPhone's interface represents a true advance in the state of the art. There are no menus, no cursor, no mouse, no stylus; the user's fingers are the only pointing device that is required. The user directly manipulates the objects of interest in the interface in a very natural manner: the touch-sensitive screen recognizes a number of fairly complicated gestures, including taps (single and double), slides, swipes, flicks, and pinches. This gesture-based interaction represents the wave of the future for interface design, even for those devices that are not handheld.

Summary. Designing effective interfaces is a more difficult problem than it initially seems. Cognitive Systems Engineering (CSE) provides a conceptual perspective and principles of design that point to successful solutions. Ultimately, interface design should be driven by the nature of the problems to be solved and by the capabilities and limitations of problem solvers.

For background information see COGNITION; COMPUTER; COMPUTER ARCHITECTURE; COMPUTER PROGRAMMING; ENGINEERING DESIGN; HUMAN-COMPUTER INTERACTION; HUMAN FACTORS ENGINEERING; PERCEPTION; PSYCHOLOGY in the McGraw-Hill Encyclopedia of Science & Technology.

Kevin B. Bennett; Shannon M. Posey