

WHAT YOU GET IS WHAT YOU SEE

G Crampton Smith

Rapporteur: Steve McGough

WHAT YOU GET IS WHAT YOU SEE

Gillian Crampton Smith
 Computer Related Design
 Royal College of Art
 Kensington Gore, London, SW7 2EU
 g.crampton-smith@rca.ac.uk

Philip Tabor
 Bartlett School of Architecture, University College
 Gordon Street London WC1
 p.tabor@ucl.ac.uk

The word *design* is often associated with the visual, affective qualities of objects. Aesthetics certainly play a central role in fields such as fashion, product, interior, and graphic design, and increasingly in software design as well. But this does not mean that the job of the *artist-designers* (graphic designers and so on, as opposed to *engineer-designers*) is primarily decorative.

Of course artist-designers try to make things look good. Pleasure in the qualities of surface is what probably drew them to design in the first place. But the vague shorthand of approbation they often use (*cool, neat*, and so on) indicates a response that is not purely sensuous. It is mediated through sets of values and meanings currently at large in the world. The artist-designer's skill, developed by training, is to design things that not only fulfill a utilitarian purpose but that also speak to the user through messages that are implicit as well as explicit. Some implicit messages are social: How do we tell the difference between the *Sun* and *The Times*, or between a Harrods bag and one from Tesco? (American readers may wish to substitute the *National Enquirer*, the *New York Times*, Saks Fifth Avenue, and K-Mart, respectively). Others convey function and operation: Which way is up? Where do I open it? What *is* it?

The focus of this chapter is interaction design—designing the way people interact with objects and systems, especially with computer software. The product, therefore, is almost entirely experiential: an alteration in the mind of the user. So the customary (originally Platonic) assumption, that substance—the “real” thing—is separate from, and is privileged over appearance, is misleading. As far as the user is concerned, WYGIWYS: What You Get Is What You See. The interface *is* the product.

The permeation by semiotics of psychology and the social sciences has left its mark on everyday common-sense and conversation. Although sometimes a cigar is just a cigar, as Freud warned, we have become accustomed to looking behind things for their significance—for the role they play in the economy of signs (see Baudrillard, 1981). It is only recently, however, that this aspect of our mentality has become important to software design. One reason is the increasing affordability of richer visual and auditory displays and the computing power to support them—and thus, potentially, of more nuanced representations. A second reason is that users have changed: fifteen years ago computers were the specialized tools of technological professionals, today they are used by people in all (well, many) walks of life for work and entertainment. Information technology is becoming an environment within which people operate, rather than a

device they pick up and use. It is part of everyday culture. People buy products, including software, not only for what those products ostensibly do, but also for what they represent. Consumers buy into a symbolic world which both differentiates them from others *and* reinforces their sense of belonging. A third reason, finally, is that in an information-rich society competition for attention is huge and discriminating. Distinctions of iconography and style are increasingly important to retain the attention of an audience.

This chapter argues that:

- The effectiveness of both the structure of a piece of software design and its interface depends on how they are interpreted by users.
- This interpretation is not context-independent. It depends on the cultural codes, shared sensibilities, emotional responses and habitual prejudices that users bring to it.
- Recent advances in display technology allow richer and more expressive communication at the human-computer interface.
- Decreasingly useful, for these reasons, is the idea that the functionality of a piece of software is separable from, and takes precedence over, its appearance.
- Decreasingly useful too, and for the same reasons, is the idea that the main task of artist-designers is to beautify already-designed products.
- The fundamental training and skills of artist-designers are in detecting, creating, and controlling cultural and emotional meanings.
- To make full use of these skills, the input of artist-designers is needed from an early stage, if not the first, in any design project.

Nowadays, in short, the habitual model of software design—engineering first, visual representation later—makes less and less sense.

Artist-Designers: The Example of Graphic Design

Software design in today's changing environment increasingly requires the input of designers who can predict and control how users will react to a piece of software. Fortunately, suitable skills are already available—in artist-designers whose experience and training is in, say, graphic design, film making, product design, architecture, painting, or illustration. Profile 9, on Teaching Design, describes how the graduate program in Computer Related Design at London's Royal College of Art brings these skills together to exploit developing technologies and media. The training emphasizes:

- *Invention*. Divergent thinking; innovation; new forms for new purposes
- *Empathy*. Putting oneself in the users' shoes; imagining what they are like; how they live, love, and have their being

- *Evaluation.* Juggling and balancing incommensurable factors, such as cost versus performance, or timelessness versus fashionability
- *Visualization/representation.* Imagining objects, ideas, processes, structures, scenarios, and so on; thinking them through in spatial or visual form; vividly communicating them to team members, users and clients.

There is no logical reason why the skills of artist–designer and engineer–designer cannot be combined in one person. A few people are totally ambidextrous in this way, and many partly so. But the time needed to gain education and experience in either domain, not to mention the differences in temperament suited to the two domains, means that most designers specialize in one or the other. It is therefore through teamwork that the domains converge.

The effectiveness of this teamwork is diminished by misunderstandings due to habits of mind that exaggerate the distinctions between: content and form; substance and appearance; function and aesthetics. We will explore these distinctions, first defining more clearly what artist–designers do, using graphic design as our exemplar.

The field of human–computer interaction came of age when the monitor replaced the Teletype printer. Since then, and at least for the foreseeable future, the dissemination of information through a two-dimensional combination of images and alphanumeric characters has been central to software design. For most of the Gutenberg era, however, this task fell to the typographer, and latterly—when the invention of offset lithographic printing allowed cheap reproduction of images as well as of text—to the graphic designer as well. So graphic design, which includes typography, is an apt illustration of more general artist–design theory, practice, and procedure.

Graphic designers find structure in information, order it in ways useful to the intended audience, and give that structure form (in text and pictures) to communicate its meaning. This applies as much to a poster for a rock concert as to a maintenance manual. So graphic design describes the means, not the end. The end of a piece of graphic design is meaning—a rich interplay of implicit and explicit messages, understood by the viewer and transmitted graphically (see Crampton Smith, 1994).

Formally, the graphic designer has remarkably few dimensions of freedom. There is the frame: the perimeter, usually rectangular, of a page for instance (or, in the case of interaction design, of a screen or window). Within the frame may be disposed blocks of text, abstract objects such as lines and geometrical shapes, and iconic or pictorial images. To this scant trio may be added the space left when these items have been disposed. That is the cast of the play.

The stature and dress of these characters may vary within limits. Text is inflected by size, font, weight, and color, for instance. But the number of possible and distinguishable point sizes is usually small; fonts are basically either with or without serifs; a given font comes in roman, italic, plain or bold; and there are only so many basic hues to mix.

Graphic design's relative lack of formal means has three consequences. First, intense attention is focused by both designer and audience on small distinctions, delicate and difficult to control—in the case of type design, difficult for the untrained eye even to distinguish. Second, each combination of distinctions is recognized as

about what they are doing. These kinds of qualities cannot be added on at the end: they are integral to the design and engineering of the product.

The Component Activities of Interaction Design

Of all the activities necessary for the development of any design, five comprise the inventive core of the process:

- *Understanding.* What is going on here? What is the underlying problem to be solved? Photographs, videos, sketches and notes can be used to aid in observing and analyzing the information or problem. Designers talk with people, especially clients and users, and look at the information to be communicated.
- *Abstracting.* What are the main elements? What kind of information is it? What do people want to do with it? What is important, what irrelevant? Lists, sketches and diagrams are the usual tools here.
- *Structuring.* What are the relationships between the elements? What different ways can they be ordered to be useful for users? What are the users interested in? How much can they take in? The designers' assumptions will be checked with the users and the clients.
- *Representing.* How can this structure be represented in visual and auditory form? What representations does the material suggest? What representations might be gleaned from thinking about the users' world? Should the representation be concrete or abstract? Is metaphor appropriate? Here the designer typically uses sketches on paper and interactive sketches in a medium such as Macromind Director™ (see Profile 10), which may be evaluated with colleagues or users.
- *Detailing.* Exactly what color should this be? What style of depiction should be used? Should this be representational or diagrammatic? How is the picture plane handled? How do the elements move? Should an illustrator be hired? Some designers work directly in a paint program like Adobe Photoshop™, others start on paper and move to the computer later.

These five processes are not executed sequentially. Designers will circle between them as activity in one throws light on another. When considering the structure of the information, for instance, a designer might also be thinking how much text can be put on the screen, or how this particular audience might most intuitively interact with the information. So the stage at which interaction designers are admitted to the design process determines the extent to which the full range of their skills can contribute to a software product's success.

Three Projects

To illustrate the process of interaction design, we will describe briefly three occasions at the Royal College of Art in which interaction designers were involved:

- When the basic engineering design was complete, so that only detailed interaction design remained to be done (the accounts project)
- At an earlier stage, when the performance specification of the application had been established, but not the model that was to represent the purpose of the application and how to interact with it (the Suitcase™ project)
- Still earlier, when the users' needs were first discussed, so the purpose and representations could be designed together, each contributing to the other's development (the fashion project).

The accounts project

Imagine that an application is almost finished but performs disappointingly. An experienced engineer is called in to optimize the code but finds that the problems are more fundamental—they lie in the data structures. Optimizing the code would make superficial improvements but the basic flaws remain. In the real world this scenario should be rare because software engineers are involved from the start. An important part of the product, however, is a coherent model that tells users what the product is and how they operate it—its central organizing idea. Yet interaction designers, who have much to contribute to this model's invention and representation, are seldom called in sufficiently early.

Two of the College faculty (Gillian Crampton Smith and Martin Locker) were involved in the attempted rescue of an accounts package, one month before it had to be finished. The original designers, an accountant and software engineer, had modeled the code's underlying structure on traditional paper-based work without sufficiently considering how computerization would change everyday accounting procedures. That the conceptual model, now hard-wired into the structure, made the package very awkward to use became apparent when the screens at last came to be designed in detail. By then the timetable allowed screens to be designed which were neat and presentable but did little more than list the functions provided. An inventive model, relating better to the users' future world and work, would have required the application to be fundamentally restructured.

The Suitcase™ project

A more productive time for interaction designers to enter stage left, then, is when the application's performance specification has been established, but not the way in which it is to be fulfilled. An example is a project for students to redesign the desk accessory Suitcase™, a mini-application that allows Macintosh users to install and uninstall fonts in their system folder. The existing interface allows users to access all the functions but gives little clue about what the application is, what it does, how you use it, or what, when you have used it, has been done. You must either remember or check the manual.

The students started by analyzing the context in which the application would be used and the kinds of tasks users would be doing. They found the basic communication problem was to distinguish easily between available fonts and fonts already installed. Possible solutions used more descriptive phrases such as *available types/loaded types* or *library/in use*, or showed the movement of fonts from a list of those available to a list of those installed.

The most successful solutions (by Robert Girling and by Sally Grisedale) totally recast the users' conceptual model of the application by abandoning the desk-accessory approach and incorporating the tool in the application's font menu. This shows how an interaction designer, thinking about the information in terms of what users do with it and the context in which they use it, may imagine very different approaches to those suggested by the way it is engineered. If interaction design is considered only at the end, software becomes driven by engineering design, of which users are rightly unaware, rather than the representations they actually interact with.

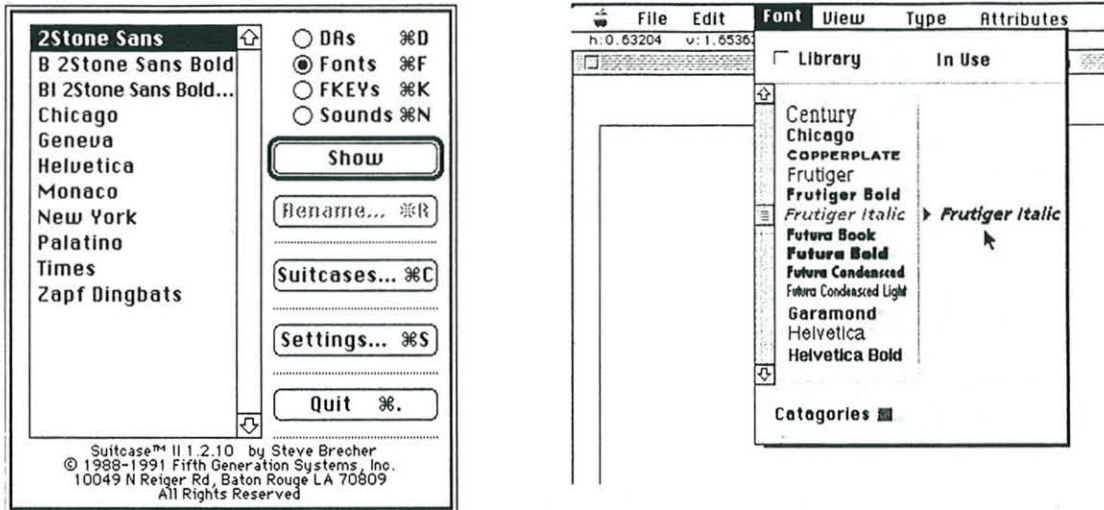


Figure 3.2 *The Suitcase problem.* A font desk accessory for the Macintosh Finder (left) allows users to access all the functions but gives little clue about what the application is, what it does, how you use it, or what, when you have used it, has been done. A more intelligible approach, developed in a short student project at the Royal College of Art (right), was to recast the users' conceptual model of the application by abandoning the desk-accessory approach and incorporating the tool in the application's font menu. Through layout, terminology and animation the purpose of the application—to install and uninstall fonts—was implied, and the state of the system shown explicitly.

The fashion project

In this project interaction designers were involved from the beginning. A consortium of clothes manufacturers asked the College to see how computers could help in the fashion-design process. A research and design team was assembled, comprising interaction designers from engineering, product, and graphic design backgrounds (Gillian Crampton Smith, Charlie Hill and Philip Joe), as well as cognitive scientists (Ellie Curtis and Michael Scaife from Sussex University). The team went through the design stages described above, cycling back through them more than once:

- *Understanding.* One of the cognitive scientists spent time at the factories and design studios, watching what was done and asking people what their work was and how they viewed it. The team began their design investigation as a graphic designer would: by considering the existing paper file compiled for each garment model or *pattern*. This file, containing some 30 different printed forms, had

evolved to enable the manufacturers to keep track of each stage of the pattern's production. The researchers could see from the file the information that already had to be provided, and they hoped to provide additional amenities.

- *Abstracting.* The paper file was used to keep track of every pattern in the production process. It contained information about its geometry, what it was made of, the trimmings to be used and who supplied them, its critical path, and test reports on sample garments. Its fundamental purpose was of course to ensure that all the season's garments got to the shops in time and to avoid costly errors. The file was not active: it did not, for instance, alert the users to emerging problems. It offered no overview of progress—how many patterns were delayed, say, or ahead of schedule.
- *Understanding.* The researchers discovered that people involved in different parts of the manufacturing process held very different views about what the process was and what was important. They concluded that, to make sense to all users, the central organizing idea of the software model should be the garment, rather than the departments, workstages, techniques, or data flow.
- *Structuring.* One of the interaction designers with a graphic-design background looked in detail at the types of information represented and how they might be categorized. The aim was to see if there were a few underlying regularities that could be represented similarly so as to reduce the apparent complexity of the information. One categorization that turned out to be very productive was to divide the information according to how it could be manipulated: pick from a list, choose from a range, tick off something finished, and so on.
- *Representing.* The team brainstormed with lists and sketches, searching for metaphors to represent the information (see Figure 3.3). Could deadlines be shown as milestones or hurdles? Might the different seasons' designs be represented as clothes rails, with garments hanging on them? Though there were useful metaphors for different elements, none seemed satisfactorily all-encompassing.
- *Abstracting.* Surveying the range of different representations they had devised, the researchers were forcibly reminded that each provided a different handle on the process. One of the cognitive scientists mentioned Searle's (1969) concept of speech acts: propositions are not simply true or false but have purpose—are exhortations or supplications, for instance. The team realized that the fashion information was only useful if it promoted action. This helped them escape from the mindset associated with the existing data structure, the paper file, and the assumption that their task was simply to represent the information.
- *Representing.* One demand, clearly, was that every pattern should be ready on time and problem-free. Managers had to be alerted to difficulties when they arose, so that they could minimize the consequences (a stitch in time). The researchers therefore sought ways to communicate a feel for how schedules were progressing

and whether there were problems ahead. The next need was remedial action. So they designed a system whereby faults—a wrinkled zip on a sample, say, or an inaccurate dimension—would be flagged on the image of the pattern itself, and notified directly to the computer desktop of the staff involved (Figure 3.3).

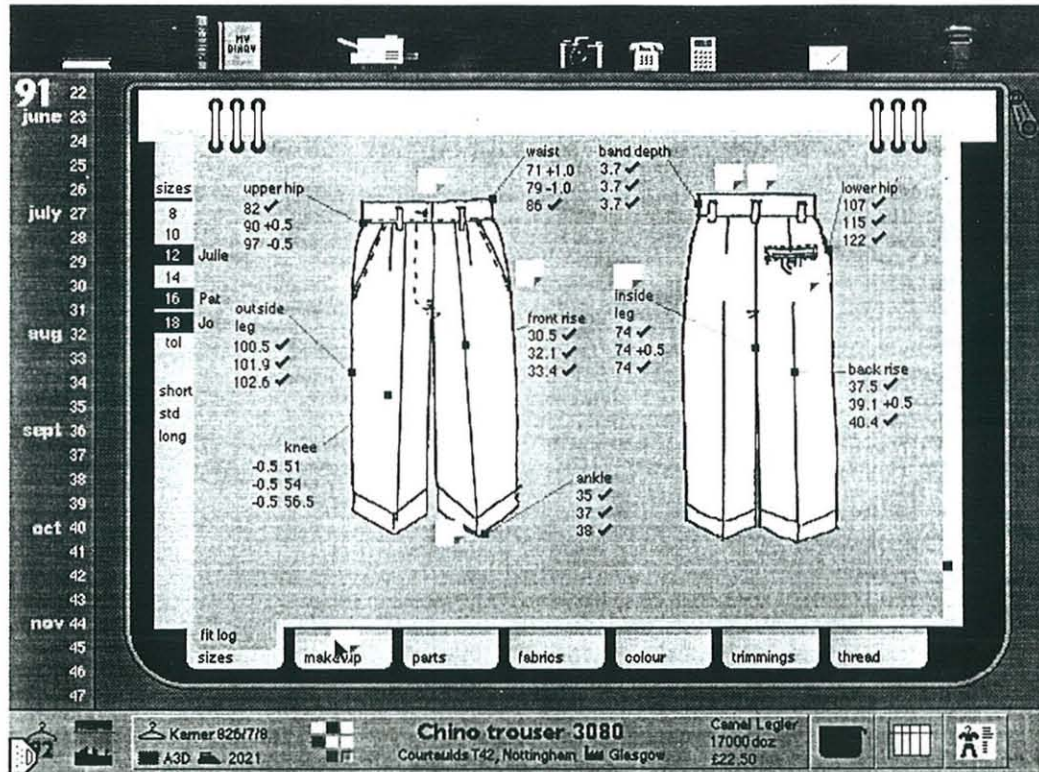


Figure 3.3 Representing action. In the interface designed for the garment manufacturing system, faults were flagged on the image of the pattern itself, and notified directly to the computer desktop of the staff involved.

- *Understanding.* The team returned to the factory to check their assumptions so far, and asked people further about the purpose of the information, what they did with it and what they might like to know that the existing systems did not tell them.
- *Structuring and representing.* To keep in mind all the ideas that had emerged and to discard those they did not think useful, the researchers took pages from all their sketchbooks and photocopied, cut up and assembled them into a large map of all the different approaches. This map prompted them to make suggestions: “If you represented each pattern on the timeline like a bar you could also show how far ahead of or behind schedule it is” ... “You could highlight this flag in color, to make the problems jump out” ... “If you represented problems as post-it notes you could see at a glance where there had been problems and if they were still there; you could use the same representation as a message to the person who had to deal with the problem”.
- *Detailing.* It was only now, well over half-way through the design work, that screen and interaction design began. The work done up until this stage on structure and design strategy is almost invisible to anyone looking at the final result. Indeed,

it only becomes obvious if it hasn't been done well: the better it is done the more transparent it is—which is as it should be.

The detailed design, done by one interaction designer with the help of a second, involved decisions such as the exact screen layout, the colors and type to be used, a consistent logical and visual style for the representations, how transitions between screens should be managed, the feedback to be given, how the user should navigate the information—and the hundred other details that make software effortless and pleasurable to use.

The two designers had decided that each pattern should have two views: its *portfolio* (information about the item itself) and its *timeline* (information about its schedule)—two conceptual models or central organizing ideas to make coherent and easy to grasp the huge range of information. This decision came from considering both how this particular information could be represented within the limitations of the medium, and how it could be made to fit the conceptual world of the users. So using the image of a portfolio with tabs for different sections, for instance, both suggests the original file and refers to the world of fashion designers. In terms of representation it had the advantage of encouraging direct manipulation and allowing problems to be shown clearly on the image of the garment. In this kind of design process few individual design decisions are startlingly innovative; the creative task is to forge them into a coherent and consistent whole—in terms of logic, representation and the users' world.

The process of detailing the representation was the most problematic in terms of tradeoffs between competing requirements. For instance, the designers wanted to represent simultaneously the timeline of each of some 30 garment patterns, the responsibility of one department. Time can be represented as a vertical or a horizontal axis. If they used the best understood—horizontal—however, they could not get enough patterns on screen from top to bottom. They tussled with ways of shrinking, scrolling and so on, but none seemed successful. They particularly resisted turning the representation through 90 degrees because the labels would have to read from top to bottom. But they finally accepted the vertical orientation because users never referred to a pattern by its name, only its number. More important than the legibility of the labels was an overall grasp of which patterns were slipping behind schedule.

There were also problems of pictorial logic and coherence. Generally pictorial icons were used to represent objects; a coat hanger, for instance, represented a season's collection of patterns. Clicking on the hanger, however, revealed not further icons representing each pattern but just a bar, colored to show whether its production was behind or ahead of schedule. Pictorial consistency was sacrificed for relevance and compactness.

Two people worked full-time for four months on the design of this prototype, and three part-time—roughly one person-year in all. Over this period the 30 different representations in the original paper file were compacted into two basic views, showing

the information in a way that allowed users to grasp it much more intuitively. The look and feel of the tool was designed to avoid the character of computer-controlled bureaucracy and resonate with the visual sensibilities of fashion designers. An important part of the system's function was that it should seem enjoyable and stylish to use.

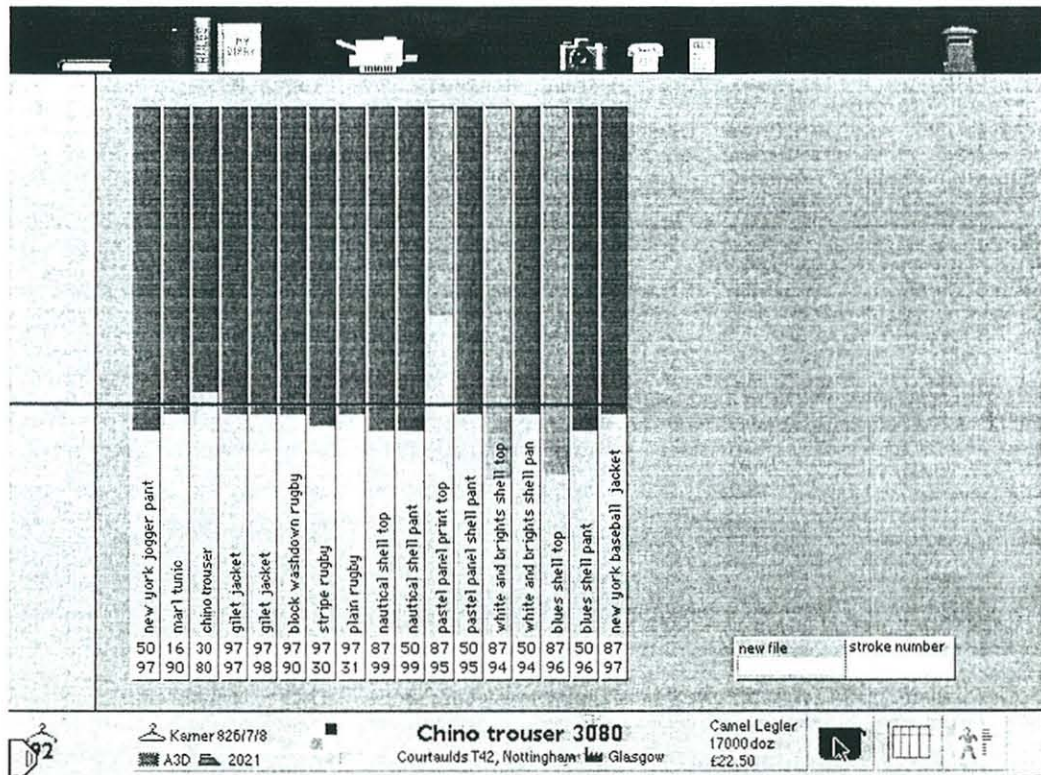


Figure 3.4 Compacting Information. The timeline view acts also as a “finder” for deeper levels of information. The original paper files gave a timeline for individual patterns but departments had no overall view. Here everyone can see the situation at a glance: today’s date is shown by the horizontal line in week 35 so we can see that production is generally in good shape with one serious problem, no 8795, shown on screen in red, which is badly delayed. Two garments (shown by light bars) have been cleared for production. This clear representation is both a spur to action—everyone will be concentrating on the one delayed garment—and an encouragement—seeing the bars creep down to the finish is delightfully satisfying

Interaction Design: More Art than Science

Interaction design involves three interlocking elements to which the skills of artist-designers are particularly germane:

- The design of what the software *does*—its purpose for people
- The design of a model of what the product *is*, and its *representation* to users in the terms of their world, through vision, touch, and sound

- The detailed design of the software's *look and feel*—the exact number of pixels, their color, how fast or smoothly they move, and so on.

Interaction design cannot ignore scientific method and engineering knowledge. Indeed, familiarity with computing technology and psychology is as essential to an interaction designer as familiarity with building technology is to an architect. But because interaction design deals primarily with values, preferences, and meanings—with, if you like, aesthetics and semantics—it can have neither a universal *predictive* theory nor ever-reliable methods to generate solutions. You can say, for instance, that a certain color of type will be legible by most readers, but you cannot say with the same degree of certainty they will or will not like it. So informed instinct (hunch) is as important as principle in interaction-design practice, and osmosis, the largely intuitive absorption of skills and values, is as important in interaction-design education as factual information-transfer.

This level of ambiguity and subjectivity is not a result of interaction design's relative youth. It is intrinsic to all the artist-designer disciplines, and gives them their power, subtlety, and robustness. Interaction design is also evolving fast, so frequently needs to break rules and overturn precedents. Like all artist-designer disciplines in the industrial and electronic ages, it must constantly reinvent itself to respond to changing situations and sensibilities.

In the end, interaction design is more of an art than a science. Its ultimate subject matter—human experience and subjective response—is inherently as changeable and unfathomable as the ocean.

Baudrillard, Jean. *For a Critique of the Political Economy of the Sign*. St Louis, MO: Telos Press, 1981

Crampton Smith, Gillian. The Art of Interaction. In R A Earnshaw and J Vince(eds.) *Interacting with Virtual Worlds*. London: John Wiley, 1994

McLuhan, Marshall. *The Gutenberg Galaxy: The Making of Typographic Man*. Toronto: Toronto University Press, 1962.

Reddy, Michael. The Conduit Metaphor: A Case of Frame Conflict in our Language about Language. In Andrew Ortony (ed.). *Metaphor and Thought*. New York: Cambridge University Press, 1993, 164-201

Searle, John. *Speech Acts*. New York: Cambridge University Press, 1969.

This paper first appeared in Winograd, Terry (ed.). Bringing Design to Software. ACM Press and Addison Wesley: New York, 1996. Reprinted by permission of the publisher.

DISCUSSION

Rapporteur: Steve McGough

Lecture Two

Early into the presentation Professor Crampton Smith put forward the view that the computer is a medium in the same manner as other material. This raised the question from the audience as to the difficulty of dealing with a medium that changes at such a great speed in comparison to other mediums. Professor Crampton Smith responded by saying that they tended to think not about what you can do but more about what it might enable you to do, but that it is often good to think about things in both ways.

Whilst talking about models the question was raised as to whether there was use of designed psychological experiments, careful monitoring and statistical analysis. The speaker responded by saying that it was more important to find out if an item is useful, fun or enjoyable. It is also important to discover if people could use the item. The need for statistical analysis is however also true. It was then asked if this wasn't the point of a focus group. The speaker responded by stating that it was similar but that she felt there were certain snags with focus groups, they tend to come out with statements of what people want.

Whilst discussing the example on clothes designing, where the designers felt computers were slow and restricting in designing, it was asked whether the designers still use paper and pencil methods. The speaker responded by saying that they probably do these days as the example dates back some six years and there has been a great improvement in CAD since then.

Later on in the example the question was raised as to whether the display of garments was representing a single designer's work or the work of a team. The speaker responded by saying it represented the portfolio of one team that would have between five and six designers in it. The question was then raised as to why the table was arranged with the writing vertical. This was explained as being due to the fact that the representation had been chosen, as the screen resolution was better horizontally than vertically.

A participant later asked if the stickers placed over the design were a design issue, with their size reflecting their importance. The speaker responded by saying that this issue had not been considered, but that garments tended to go through the system with various stickers. A question was then asked of the emotional effect of seeing stickers on your work. The speaker responded by saying this was not an intended effect, stickers could build up due to illness as well as poor quality work.

At the end of this example the question was raised as to how this had affected the opinions of the designers. The speaker responded by saying that it gave the data processing department a higher status, also that she felt the designers liked the system but worried about who would maintain it. It was then asked if these were the final results, to which the speaker responded that the work had been only to develop a prototype.

A member of the audience then asked how you could measure use of a product. The speaker responded by stating that such a thing was difficult to assess, as the use would change over time. She then raised two points about the work:

1. The designers thought that they would be willing to use the product.
2. They felt that they had combined many of the existing forms in a useful way, though this was from more of an organizational point and not as related to computers.

Another person asked what difference an artist designer could bring to a situation like this that a software company couldn't. The speaker responded by stating that the advantage was

in representing data graphically, though there was no reason why a software company couldn't do the same. The speaker was then asked if she felt that some people were more willing to talk to designers than software designers. The speaker agreed with this to a certain extent, and qualified it by saying that there was less of a gap between graphic designers and fashion designers than there was between fashion designers and software designers.

The question was asked whether the company was actively involved in the work undertaken. The speaker responded by saying that the project had lasted nearly a year during which time as work was completed it was being shown to them. Thus they were involved but not actively involved in the design.

During the presentation concerning ascertaining the views of the elderly in Amsterdam, a member of the audience asked how old these people were. The reply to the question was that they were over sixty, and after some debate over a modern definition of elderly, this was revised to people who have time to spare.

At the end of the presentation on the surveys of the elderly in Amsterdam, a member of the audience inquired how this work related to software. The reply indicated that the relevance was not known yet and that this was just the first stage of thinking about what to design.

At the end of the session several questions were raised, the first asking for the elderly persons example, did they have any plans yet for what to implement. The speaker responded by saying that there were several practical ideas as to what could be done. One idea that had come forward was the ability to keep in touch with grandchildren, though this may be difficult to achieve without being intrusive. There was some feeling that different regions had different views on what they wanted. An example was given for grandparents in Italy who would like to watch television with their grandchildren in a different city, although this was not considered a good idea by the grandparents in Norway.

A question was then raised based on the example of the squeezable phone, as to how you could determine which ideas were cool and which were not. The response to this was that the only way to tell was to try the idea out.

The next questioner asked if any surveys had been carried out to determine how people used phones. The response to this was that the whole point of the project was to determine what could be done with dexterity. This led to the statement that it should be more a case of looking at the dexterity of the phone rather than the dexterity of the hand, otherwise this would lead to a miss-match in design. Another member of the audience then stated that it all depends on whether you are involved in a design project or just after something that looks flash. The speaker summarized this by saying that she was surprised, as they had been developing cool ideas for some six years and some of them were now available in the shops. The designs did seem far out but they are pleased that people do want them.

An issue was raised by another member of the audience, the fact that designs have both a dimension of space but also dimension of time. Followed by the question that as people are only alive for a certain time, how can you take this into account in designs. The speaker suggested making things from materials that were longer lasting and would develop patterns over time. Also that there was a need to make things that you don't have to throw away.

The last questioner asked how fast could articles be changed since individuals need some level of stability. The response was that more fundamentally since so many things are going virtual, this was another problem.

