Artificial Intelligence for Engineering Design, Analysis and Manufacturing (1999), *13*, 303–320. Printed in the USA.

Interactive generative systems for conceptual design: An empirical perspective

CLAUDIA ECKERT,[1] IAN KELLY,[2] AND MARTIN STACEY[3]

Departments of [1]Design and Innovation and [2]Computing, The Open University, Milton Keynes, United Kingdom
[3]Department of Computer and Information Sciences, De Montfort University, Milton Keynes, United Kingdom

(RECEIVED June 3, 1998; REVISED March 9, 1999; REVISED March 18, 1999; ACCEPTED April 1, 1999)

Abstract

This paper argues from extensive research findings in design psychology and industrial design processes, as well as our own observations, that interactive generative systems can be powerful tools for human designers. Moreover, interactive generative systems can fit naturally into human design thinking and industrial design practice. This discussion is focused on aesthetic design fields like knitwear and graphic design, but is largely applicable to major branches of engineering. Human designers and generative systems have complementary abilities. Humans are extremely good at perceptual evaluation of designs, according to criteria that are extremely hard to program. As a result, they can provide fitness evaluations for evolutionary generative systems. They can also tailor the biases that generation systems use to reach useful solutions quickly. We discuss an application of these approaches: Kelly's evolutionary systems for color scheme design. Automatic design systems can work interactively with human designers by generating complete designs from partial specifications, that can then be used as starting points for designing by modification. We discuss an application of this approach: Eckert's garment shape design system.

Keywords: Generative Systems; Automatic Design; Design Psychology; Aesthetic Design; Conceptual Design

1. INTRODUCTION: ACHIEVING HUMAN–COMPUTER SYNERGY

The purpose of intelligent systems for supporting human designers is to achieve human–computer synergy, to achieve greater creativity and effectiveness than either humans or artificial intelligence (AI) systems can manage on their own. This entails embedding intelligent systems into human design activities, not only to take over subtasks that humans find difficult or tedious, but also to exploit the power of human design thinking.

The argument of this paper is that generative systems for automatic design can be powerful tools for human designers, but need to be grounded in an understanding of design. While the intrinsic structure of the design problem is the most profound influence on what designers do, their strategies and actions are powerfully constrained by their cognitive capacities, and by the representations and operations afforded by the tools they use. Effective tools must be engineered to fit (1) the task, (2) the cognitive characteristics of their users, (3) their users' skills, and (4) the organization of the design process within its industrial context. This requires an awareness of design psychology and a thorough study of the design processes in which a tool will be used.

Effective interactive AI systems should enable human designers to exploit the strengths of AI systems, to perform complex computations, handle multiple constraints, and explore alternative solutions. As interactive tools, generative systems can exploit the strengths of human designers, to evaluate the characteristics and qualities of designs perceptually, and to use visual stimuli as triggers to imagine novel designs. Automatic design systems can work interactively in different roles (which can be combined): evolving designs iteratively with humans performing selection and fitness evaluation; completing designs from partial specifications; and generating initial candidate designs for humans to modify.

1.1. The power of bias

For most interesting classes of artifacts, the space of possible designs is immense. At any stage in the construction of

---

Reprint requests to: Claudia Eckert, Engineering Design Centre, Department of Engineering, University of Cambridge, Trumpington Street, Cambridge CB2 1PZ, United Kingdom. Phone: +44-1223-332673; Fax: +44-1223-332662; E-mail: cme26@eng.cam.ac.uk

a design, the vast majority of possible changes are either nonsensical or foolish. So to create a design that meets its designers' objectives, the generation process must be strongly directed. In the generation of successive partial designs, this direction can come from the expressive power of the representation in which the design is expressed, from the range of design creation actions available, and from the ways in which these actions are selected. In the evaluation of successive partial designs, it can come from the constraints the design must meet and the qualities it must have.

Human designers combine all these sources of guidance. Automatic design systems need strong direction to produce appropriate rather than inappropriate artifacts: they need to be *biased* toward producing some designs rather than others. But when their users want to explore alternative designs for reasons that cannot easily be programmed, *bias* is harmful: Automatic design systems should cover the whole space of appropriate designs, and not just a small subset of it. In this paper, we argue that stand-alone generative systems have biases that are far too strong for many applications. A more fruitful approach is to build in constant domain constraints (for instance in tailoring, that the sleeve crown curve is the same length as the armhole curve), and allow users either to program constraints and biases, or to provide the biasing themselves interactively.

1.2. Generative systems for visuospatial conceptual design

In this paper we concentrate on the use of generative systems for design, in fields where design involves visuospatial reasoning about shape and appearance, and especially in fields where design is partly concerned with aesthetics. The view of designing we present is grounded in the first author's extensive study of the knitwear design process (Eckert, 1997*a*, see Section 5.1), as well as the research literature on designing in engineering and architecture (see for instance Schön, 1983; Akin, 1987; Cross, 1989). Most of this analysis is applicable to many branches of engineering, though some require optimization and reuse of standard solutions rather than support for variety, and electronics and control engineering are distinct arts.

We are primarily concerned with what engineers term conceptual design: the stage in which engineers make the major decisions about what a machine does and how it works, as opposed to embodiment design, in which these decisions are fleshed out in exact detail. Other fields have a different division of labor, and different terminology for the same distinction. For instance, in knitwear design, aesthetic design by knitwear designers is followed by technical design by technicians.

In Section 2 we consider the strengths and limitations of generative systems as tools for conceptual design, to set a context for Section 3, in which we discuss aspects of how designers design that are vitally important for understanding how to embed generative systems into human design processes. In Section 4 we point out some problems in conceptual design that interactive generative systems can alleviate. Sections 5 and 6 present two examples of interactive automatic design systems based on psychological research and design process analysis, for garment shape design and for color scheme creation.

2. ROLE OF AUTOMATIC DESIGN IN CONCEPTUAL DESIGN

The strength of generative systems as tools for conceptual design is their ability to explore the whole of a space of possible designs. A system's design representation formalism and its set of operators for constructing designs define this space; they define the aspects of the final product that are included in the design, and the level of abstraction at which they are described. Generative methods have (or can have) many characteristics that make them ideal tools to support human designers, who can control them by tuning the characteristics of the search space or by guiding the search itself. In the following sections we will see how they fit into patterns of human cognition and work practice.

2.1. Strength of generative systems

For the purpose of this paper we use the term generative systems in a broad sense, to cover methods that generate designs based on a set of input specifications. These include evolutionary methods including genetic algorithms; rewrite rule methods such as shape grammars (see Stiny, 1980; Knight, 1994); and heuristic rule methods including case-based reasoning (see Kolodner, 1993).

Generative systems can be powerful tools to create new designs fast, but require careful and elaborate research and development by the programmer. Mistakes in the design of a generative system are costly and difficult to change. In most systems, however, the difficulty does not lie in generating new designs, but selecting those that are worth considering by a human user or the system itself for further development. A system can generate all the alternative designs that are consistent with (1) the inputs describing the design task, (2) the generative rules and algorithms, and (3) the constraints built into the representation formalism, to map the entire space of designs. If this space is large, further constraints are necessary to keep the number of designs within manageable bounds. Restrictions on the space of permitted designs can be built into the design representation formalism or the generative rules and algorithms, or built into separate evaluation rules. These can ensure that generative systems discard, or never generate, designs that do not meet basic constraints and quality criteria. This approach can be used to generate designs using complex formal or mathematical methods, or conforming to complex or computationally difficult sets of constraints. Such designs can be difficult or impossible for human designers to create, or so effort-intensive that human designers can only create one or a few alternative designs when generating many would be beneficial.

Generative systems create designs that are complete within the scope of the design representation formalism. Thus, the degree of completeness of the design is well understood. This complete description can be used to create mappings to different notations and visual displays.

2.2. Stand-alone generative systems

Although independent generative systems are extremely valuable for modeling human design thinking, and have achieved spectacular successes, notably the shape grammars for Palladian villas (Stiny & Mitchell, 1978) and Frank Lloyd Wright prairie houses (see Knight, 1994), they are complex and difficult to build. Moreover, each generative system works only in a single style. They exploit strong biases to reach a small part of the space of possible designs. However, to be widely applicable to under-constrained design tasks with large design spaces, such as architecture and knitwear, generative systems need weaker built-in biases, and an external source of guidance: a human user.

2.3. Users specifying biases for generative systems

Generative systems for design follow a cycle through problem specification, design generation, and design evaluation (Fig. 1), that is closely analogous to the cyclic pattern of human design behavior (Figs. 2 and 3). Evolutionary techniques such as genetic algorithms create designs by iterating through this cycle many times. Heuristic rule-based systems might only go through one cycle, while a shape grammar might be used in either mode [see Chase (1998) for a discussion of alternative modes of interaction with shape grammars].

Human users can interactively control the behavior of generative design systems by specifying the features that designs must have. These characteristics may be constraints that must be met, or desirable characteristics the design should have (that can be computed after each design is created), or partial designs that the system should keep and extend. These different types of specifications have different implications for how a generative system must work. However, all serve to direct the generative system to a small part of the space of designs made possible by its representation formalism and operators. Kelly's suite of evolutionary color design systems, described in Section 6, allows the users to program constraints that generated designs must conform to, as an indirect but computationally feasible way of specifying desirable emergent properties. The user controls Eckert's garment-shape design system, described in Section 5, by supplying partial designs.

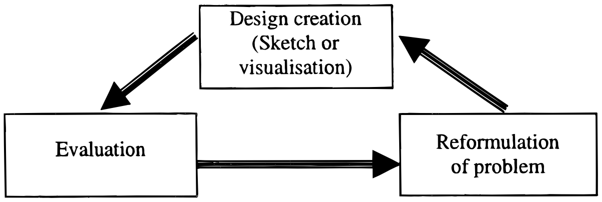


Fig. 2. Human design cycle.

Generative systems for design can work interactively in different ways that depend on how much the user constrains the problem initially, and on what role the human takes in the creation of designs. A potentially important role for generative systems is extending designs when the human designer has already made some important decisions, to explore and illustrate the implications of those decisions. Given a tight but partial specification (perhaps expressed in terms that require further effort to turn into a structural description, such as a garment shape described as a set of measurements), the system generates one or several complete designs, taking over difficult or tedious algorithmic subtasks. Such a system should ensure technical correctness, perhaps interactively by interrogating the user, and might use aesthetic heuristics. This is the primary function of the garment shape design system we describe in Section 5.

Design by computer and design by human are not mutually exclusive. A partial or complete design produced by an automatic design system may serve as a starting point for humans to design by modification. If the design editor used for this purpose does not maintain completeness and correctness, the automatic design system can propose further completions and corrections of inconsistencies. This is how we envisage the garment shape design system we describe in Section 5 being used; we argue in Section 3 that this fits naturally into human design thinking and current industrial practice.

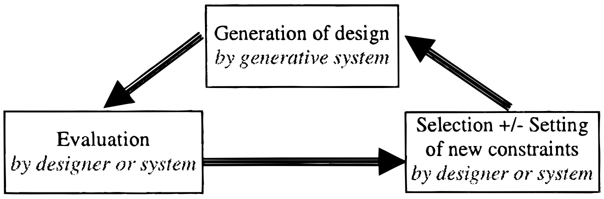


Fig. 1. Generative design cycle.

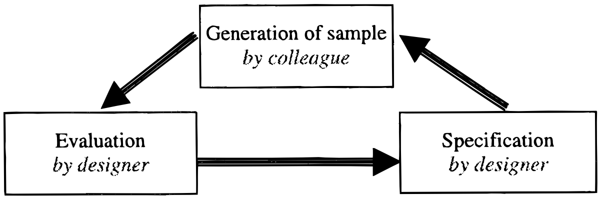


Fig. 3. Design evaluation cycle in industry.

2.4. Users in the generative loop

The great challenge of building generative systems is evaluating the generated designs for further development or final presentation. In an interactive generative system this task can be largely taken over by the user, as illustrated by Kelly's evolutionary systems for color scheme design, discussed in Section 6. Given a loose specification (or set of constraints) defining a large space of possible designs, an evolutionary system creates a sequence of designs with the human user selecting good designs for further development.

Generative systems can include evaluations of properties of the design that can be determined directly from its structural features; computer-generated critiques of designs and other decision-making can be extremely useful, and critiquing systems are a major area of AI research (Silverman, 1992; Fischer et al., 1993). Often, however, the users require evaluations of emergent features of designs, for technical and aesthetic reasons. These evaluations are likely to be extremely difficult to compute from the system's representation of the design. They are also likely to depend on subtle details of the design task and the context, which are hard or impossible to model computationally, and which certainly cannot be modeled for every individual design task. As we describe in Section 3.3, humans are remarkably good at making fast perceptual evaluations of complex and subtle properties of designs by looking at pictures and diagrams, and professional designers' talents and training make them especially good at this. This ability enables skilled users to provide generative systems with quality evaluations quickly and efficiently.

2.5. Generative systems as tools for designers

The creation and evaluation of complete designs has several strong advantages for interactive systems to support human designers:

- A great number of designs can be produced, spanning a large search space.
- The creation of new designs is relatively fast.
- New designs can be created using computational methods or conforming to computable constraints that are difficult or impossible for humans to use.
- All designs are specified at a predictable and well-understood level of completeness, abstraction, and detail.
- All designs can be displayed in ways that suit the user, for example, in pictures or schematic diagrams, and if appropriate in a variety of different forms for different purposes.
- All designs are specified precisely and unambiguously at the built-in level of description.

3. CHARACTERISTICS OF DESIGNERS AND DESIGNING

The term design covers a wide range of tasks, activities, and products, but in all cases it entails solving what psychologists call an ill-structured problem, to create a description of an artifact. An ill-structured problem (Simon, 1973) is one for which a solution method cannot be derived from the problem statement, so it cannot be solved by any linear sequence of correct reasoning steps. Nor does it have a single correct answer, but may have a range of different good answers. The intrinsic structure of design problems dictates that they are solved by making reasoning jumps that may not be sound and so must be evaluated when they have been made. (When we can perform a design task by using a sound algorithmic method, we no longer think of what we do as designing. Sometimes people treat problems that have algorithmic solutions as design problems because the algorithmic methods require too much computational effort or too much mathematics.)

3.1. Design as a style of thinking: The design synthesis loop

Designing is characterized by a distinct thinking style. Talented and successful designers are those who have an aptitude for it. Designers proceed by repeating the cycle shown in Figure 2: analyze and reformulate the problem, imagine a design, evaluate the design [Asimow (1962); see for instance Cross (1989)]. Of course, designing is more complex than this. Whenever possible, design problems are decomposed into manageable chunks with relatively simple interactions; many of these chunks require linear problem solving rather than designing. In engineering and other industries producing complex products, design often comprises a set of nested synthesize–evaluate–reformulate loops, varying in duration from seconds to days. Rapid perceptual evaluations are an integral part of idea generation in architecture and other fields [see Sections 3.4 and 3.5, and Goldschmidt (1991), Purcell et al. (1994), and Suwa et al. (1998)]. Evaluations of other aspects of the design may be planned tasks rather than alternations of mental activities, involving significant reasoning, and requiring significant design effort before they are possible. Some complex design processes use specialist personnel to perform particular evaluations. In some industries the outer synthesis–evaluation–reformulation loops may involve building and evaluating prototypes (as in Fig. 3). In the knitwear industry designers get feedback in the form of manufactured sample garments. The generative design cycle of a generative system (Fig. 1) closely matches human thought processes in design (Fig. 2) and the organization of work in some design industries (Fig. 3).

What is produced in each design synthesis step depends on the designer's mental context, primarily on what the designer is thinking about (what is in consciousness), but also on the designer's recent experiences (the elements of long-term memory that have recently been created or activated). It is also dependent on the designer's knowledge and on the mechanisms of human perception (see Section 3.4). The context includes the formulation of the problem, and the previous version of the design that the designer is working from. The designer's search for a good design is typically a jagged, spontaneous path, in which each step triggers new ideas in a way that can only partially be controlled. In some aesthetic fields, such as knitwear design, designers actively enrich their context, by searching for sources of inspiration that trigger the sorts of design ideas they want (Eckert, 1998; Eckert & Stacey, 1998).

Designs are typically evolved, by a sequence of modifications and extensions. They are seldom created *ab initio*; instead designers alter previous designs and reuse components and solutions to problems. This is true in fields demanding novelty, such as knitwear design (Eckert, 1997*a*, *b*), as well as in fields where reusing standard components and methods is desirable, such as engineering. At the same time the designs that are generated by reuse and modification, by humans or generative systems, can spark off human imagination [see Eckert (1998) for an analysis of the mechanisms of inspiration]. Hence, we regard providing starting points for humans to design by modification as an important role for generative systems, and facilities for manual editing as a valuable feature in an evolutionary generative system.

3.2. Mental representations of designs

Imagining a new design, even as a small modification of an old one, is a pattern synthesis operation of exceptional complexity. Although imagining designs is a skill that develops with increasing knowledge and experience, and can, to some extent, be taught, successful professional designers are usually people who have a high degree of natural ability to visualize and imagine complex objects and patterns. In visual domains like knitwear design, the ability to visualize designs is the key talent, which is trained throughout the designer's working life; good designers can visualize and mentally manipulate products often in considerable detail. For example, some knitwear designers comment that they see design ideas as realistic garments, which they can alter, re-color and rotate mentally, and so find simulation software useful only for marketing.

Many designers imagine designs visually with a lot of detail even when it is not needed, and in consequence are much more comfortable thinking in terms of concrete objects instead of abstractions. They frequently think about relatively detailed concrete designs even when they are merely placeholders for categories. This is especially true in fields like knitwear design. For example, knitwear designers use specific garments that they have seen or visualized to represent and describe garments of a certain mood and style that they wish to include in their collection [see Eckert and Stacey (1999)]. Many engineers find it difficult to use abstract formal methods for conceptual design, for example, bond graphs describing functional relationships [see for instance Karnopp et al. (1990) and Bracewell and Sharpe (1993, 1996) for AI applications], partly because they automatically include rich visuospatial detail in their mental representations of designs, even when it gets in their way. The concepts engineers use in conceptual design usually cut across conceptual categories at a middle level of abstraction between category and particular product and include physical principles and mechanisms, and often provisional assumptions about size, shape, and orientation.

On the other hand, design ideas can often be *vague*—designers only have a rough overall idea for the design; or *incomplete*—only embodying decisions about parts of the design; or *inconsistent*—embodying unresolved contradictions. (Inconsistency is a frequent problem when knitwear designers specify garment shapes—see Section 5.)

Designers often think about designs visually in terms of *emergent properties* that they want the design to have, which are often not closely related to the structural terms in which a design must be specified before it can be realized. For instance, there is a complex and subtle relationship between the aesthetic and technical characteristics of complex knitted structures. For instance, designers might want a color scheme to look "autumnal," or a knitted fabric to look like crochet-work. Many notations for describing designs structurally obscure emergent properties. Designers may have a clear and detailed view of the emergent effects they want, but no idea how to construct a design to achieve them. (This is a significant problem for knitwear designers without a solid grounding in the technicalities of knitting.) Conversely, if designers think in terms of the structural characteristics of a design, especially when using a formal notation, they can lose track of the emergent characteristics of the whole. It can be difficult to keep structure and appearance in mind. Alternative ways of formulating objectives and design ideas can have a powerful effect on how a design is created and on the eventual result.

Generative systems can make emergent properties salient by creating visual representations by applying technical rules to a structural representation of the design. Each visual representation displays information about some structural aspects of the design, and may conceal other information; similarly, different visual representations can reveal some emergent properties of the design and hide others. Using computational representations and visual displays of designs allows designers to work with structural and emergent characteristics as they wish without losing track of information, or losing the connection between appearance and structure.

3.3. Problem formulation by collecting constraints

An important part of designing is reformulating the problem by collecting constraints. Experienced designers prune the design space as much as possible by collecting all the available constraints and identifying the most important [see for instance, Katz (1994)]. In knitwear design, for instance, they look at customer requirements, materials, styles, and contexts to zoom in quickly on one part of the design space.

By discarding options, designers make fundamental decisions about the product, which may later have costly consequences, without being conscious that they are designing. For example, when knitwear designers select yarns for their entire collection they discard or choose certain yarns by looking at one thread for a few seconds, when detailed technical properties of the material later have great effects on the prototyping time for the entire collection.

A generative system can be guided by the initial constraints that are set up by the user. On the other hand, it can also overcome the problems caused when designers overconstrain their designs too early and zoom in on a design solution without exploring the design space fully, by generating a larger number of alternatives in more detail than is possible for a human designer in the available time.

3.4. Perceptual evaluation of design quality

Experienced designers develop powerful skills for *perceiving* the characteristics of a design or partial design that they see or imagine. They can perceptually recognize its features and properties [Schön (1983) terms this *appreciating* its characteristics]. Moreover, they can perceptually evaluate its quality, along technical and aesthetic dimensions. Designers' powers of perceptual evaluation are precisely tuned to the needs of any particular task. In other words, expert designers recognize good designs when they see them, even when they cannot imagine them or construct them. Moreover, they can recognize weak or partial resemblances to what they want, and recognize which aspects of a design are right or wrong; thus, they can recognize steps in the right direction toward a successful design. The knowledge designers use for perceptual evaluation is tacit: Designers know when something looks right, even though they might not always be able to articulate why. Training tacit perceptual skills for design evaluation is a major feature of much design education, notably in fashion and knitwear design.

Most of the research on how designers use external visual representations [see Purcell and Gero (1998) for a thorough review] has been on how and why architects sketch. Notable contributions have been made by Schön (1983), Schön and Wiggins (1992), Goldschmidt (1991, 1992, 1994), and Goel (1995), who also studied mechanical engineers and designers of instructional materials. Their research shows that architects and others make a move in design space, evaluate what they have produced (typically by examining what they have done with their latest sketch), and reformulate their problem by adding information to their understanding of it. Goldschmidt (1991) reports that the architects in her experiments alternated between *seeing as* (perceiving the design their sketch depicts or suggests) and *seeing that* (perceiving that particular characteristics are true of the design).

As Todd and Latham (1992) have argued in the context of computer art, the remarkable human ability to recognize subtle perceptual characteristics of designs can be exploited to achieve synergy between humans and generative systems, by using the human in the generation—evaluation—selection loop to evaluate and select new designs generated by the system.

3.5. Visual displays as a stimulus for creativity

Visual displays of designs and sources of design ideas play an important role in most designers' creative processes. Research on sketching (see the references cited above) has shown that for designers in many fields, drawing sketches to externalize their design ideas is a vital part of their creative thinking. Sketches are not only used to depict objects, individual components, and spatial relationships in a form of external memory, but also to overcome limitations in memory capacity. They also enable perceptual evaluations, activate knowledge held in long-term memory, and trigger design ideas. Schön (1983) [see also Schön and Wiggins (1992)] views designing as an interactive conversation with the sketches, and points out that architects' and other designers' ideas are changed by the feedback they get from their sketches, even though they might not be conscious of the influence the external representation has on the design. In complex design fields, like much of engineering, designers can switch quickly between different mental representations of designs when they think about different aspects of them, and their sketches can represent several different types of information using different notations (Stacey et al., 1997). Research on sketching [for instance Goel (1995); see Purcell and Gero (1998)] stresses the importance of restructuring in the development of new ideas—making qualitatively different reinterpretations of potentially ambiguous images; restructuring is prompted by dissatisfaction with the existing form of the design (McFadzean et al., 1999). The ambiguity and vagueness that make sketches a problematic means of communication (Stacey et al., 1999) are a benefit for triggering alternative perceptions.

Not all designers externalize their designs during idea generation; some designers who have strong and vivid mental imagery use sketches only to communicate their ideas to others, but use their mental representations of designs to trigger evaluations and new ideas. [Nonsketchers are common in knitwear design. Buildings are too complex for many architects to work without externalization, though Frank Lloyd Wright produced few sketches and claimed to design in his head (Goldschmidt, 1995).] Perceiving implicit and emergent properties in mental images and restructuring them is not impossible for strong visualizers [for instance, Peterson (1993); see Purcell and Gero (1998)], but recognizing such properties is enormously facilitated by external visual displays.

Previous designs and other visuospatial forms play an essential role as sources of inspiration for designs that are produced by adapting and combining preexisting design elements, and generating new design elements by analogical mapping and translation. Seeing and remembering sources of visual ideas frequently triggers design invention. [See Eckert (1997*b*, 1998) and Eckert and Stacey (1998, 1999) for our study of the use of sources of inspiration in knitwear design, a process in which they are consciously and explicitly used. See Goldschmidt (1995) for a discussion of precedents and analogical thinking in architecture.]

The potential of preexisting visuospatial forms for stimulating creativity is highlighted by Finke's (1990) experiments on imagined *preinventive forms*. Subjects were asked to imagine combinations of simple geometric forms, and then interpret the combined forms as practically usable objects of a particular type, which they were able to do; moreover, their success increased with the tightness of the constraints placed on them.

The accumulated evidence about the use of images of different sorts in design thinking and in commercial design processes indicates that displays of visuospatial forms related in different ways to the results designers want to achieve are a valuable source of triggers for designers' own visuospatial creation. Generative systems offer a variety of ways for designers to tailor such displays interactively to their own requirements. Design generators can provide starting points for adaptation. Abstract form generators may also be useful as sources of interesting forms requiring further translation into design elements. Computer-generated images typically lack the vagueness and ambiguity produced by the imprecision of sketches, which may channel designers toward literal interpretations of the images and limit their ability to restructure and modify them when imagining new ideas. (But the appearance of sketchiness can be artificially added; the value of doing this should be investigated experimentally.) On the other hand, computer-generated images are superior to sketches in their ability to reveal the emergent global effects of complex forms.

3.6. Satisficing

Designers are usually looking for satisfactory solutions to their problems, rather than the best solutions; they often have no way to judge how easily they can obtain better solutions. So they stop as soon as a design is judged good enough. Simon (1981, for instance) terms this strategy *satisficing*. In industrial practice designers often have to settle for suboptimal designs because they are running out of time. For instance, when a prototype of a knitted garment does not look like what the designer intended, but it works, it is often accepted. Designers have strongly pragmatic attitudes that increase their willingness to use computer systems if they can speed up the design process, thus leading to more or better designs.

4. SOME PROBLEMS IN CONCEPTUAL DESIGN

Conceptual design in engineering and other industries is problematic: Many conceptual design processes are inefficient and ineffective, and mistaken decisions at this stage can be difficult or impossible to correct later, with expensive consequences. In engineering, the need to make conceptual design more effective is widely appreciated in industry and academia, with the widespread introduction of concurrent engineering techniques, and many research projects on computer support for conceptual design. By contrast, the inefficiency of the design process in our example field of knitwear design has seldom been recognized (Eckert, 1997*a*; Eckert & Demaid, 1997). In this section, we argue that interactive generative systems can offer partial technological solutions to some of the common difficulties encountered in conceptual design.

4.1. Ensuring systematic exploration

Failure to explore the space of possible designs adequately is sometimes a major reason for unsatisfactory conceptual design in engineering. Human designers frequently make small modifications to existing designs, when a more radical approach would lead to better products. When they are seeking innovative designs they often focus on the first promising ideas they think of, instead of systematically exploring alternatives. A primary goal of prescriptive engineering design methodologies is to compel designers to consider a wider range of alternatives at a high level of abstraction. Generative systems can support systematic exploration by covering the complete design space defined by their representation formalisms and operators. They can suggest concepts that human experts have not thought of [for instance, Chakrabarti and Tang's (1996) mechanism generation system].

4.2. Order of decisions

Design processes can be biased, often harmfully, when the order in which particular decisions are made is constrained. This can have catastrophic consequences when the design process becomes committed to poor early decisions. One motivation for concurrent engineering is to eliminate decision-ordering biases that are due to the social organization of the design process. Computer tools for design can also impose an ordering of subtasks and decisions. Green (1989, 1991; Green & Petre, 1996) discusses this within his *cognitive dimensions* framework for analyzing the characteristics of computer interfaces. [See Stacey and Eckert (1999) for a discussion of ways CAD systems can bias designs.] Generative systems offer a way to limit decision-ordering biases, by allowing designers to explore the consequences of adopting different decision orderings and constraint priorities, and comparing the results.

4.3. Notations

Design can be influenced by the notations designers use to represent their designs; they affect which aspects of the design are considered, and when, and the order in which decisions are made. In some situations they can determine the terms in which designers think. Any notation for describing a complex manufactured product can only represent some aspects of the design. For instance, pattern books and knitting machine CAD systems offer a variety of notations for knitted structures, but knitwear designers and technicians have no adequate way of describing significant aspects of a knitted garment other than the real thing. Designers are handicapped if the representations they use fail to highlight emergent properties that they need to consider. Generative systems can alleviate this problem by generating candidate designs that are complete within the limitations of the systems' representation formalisms, and so can be displayed in a variety of different visual representations that show different aspects of the design and reveal important emergent properties.

4.4. Communication and interpretation of design ideas

Failures of communication between the members of design teams can be a major problem in conceptual design. As we have found for knitwear design (see Section 5.2) breakdowns in communication can be caused by a combination of: inadequate or poorly used notations; cognitive and cultural differences between team members; differing understanding of the task and the design depending on different expertise and different concerns; as well as by the intrinsic difficulty of communicating design ideas. On the other hand, communication works well when designers make good use of a complete notation. Unsuccessful interaction can also be caused by designers trying to communicate ideas that they have not worked out in sufficient detail: because they do not know what they want; because they are required to produce seemingly precise specifications of imprecise ideas; because they have only developed some aspects of the design; or because they have not recognized inconsistencies between different parts of the design.

Using CAD systems for designing and for communicating conceptual designs, rather than drafting, can force designers to make provisional commitments to exact parameter values when they want to express qualitative decisions and approximate values. They make it difficult to combine different levels of abstraction in a single description of a design. Sketching and verbal descriptions provide more flexibility, but sketching involves a tradeoff between the effort involved and the precision and detail of the information a sketch conveys. Because sketches are imprecise, they require interpretation; interpreters may fail to recognize differences from category norms, and may have different standard meanings for the same categories. Sketches are limited as a means of communication by the lack of an agreed meta-notation for communicating different levels of concreteness, precision, and commitment (Stacey et al., 1999).

Interactive generative systems can contribute to reducing communication problems by enabling designers to create complete and consistent representations of the designs, which are either precise or have a well-understood degree of vagueness.

5. OVERCOMING COMMUNICATION PROBLEMS IN KNITWEAR DESIGN

The first author has developed (Eckert et al., in press) a prototype design support system for garment shape design, which enables the designer to specify designs rapidly and incompletely, and see a graphic display immediately. It uses novel mathematical techniques to construct Bézier curve models of garment shapes conforming to the constraints imposed by the technical characteristics of knitted garments.

The system is designed to fit the cognitive characteristics and the working patterns of professional knitwear designers. It draws on an extensive study of the knitwear design process across the industry (Eckert, 1997*a*). It is intended to enable knitwear designers to develop better conceptual designs faster, and to communicate their design ideas more effectively to their technicians. It illustrates several of the themes we have discussed earlier in this paper.

5.1. Knitwear design

The design of knitted garments is a task shared between knitwear designers, who are responsible for deciding the visual and tactile appearance of a garment, and knitting machine technicians, who do a lot of detail design in the course of using CAD systems to program knitting machines to manufacture the garment. Knitwear design involves the creation of fabric and shape together; it is made complex by the subtle interaction between the appearance of a knitted structure and its technical properties, which is hard for a designer to anticipate. A different yarn or a small change to the design can make a garment impossible or much more time-consuming to knit.

Designers begin working on a season with *research*, that is, finding out what everyone else is up to, and collecting sources of inspiration for their designs (Eckert, 1997*b*, 1998; Eckert & Stacey, 1998). Having decided what types of garments they require for the season's collection, they produce conceptual designs for individual garments. They describe these conceptual designs for the knitwear technicians in *technical sketches*, which comprise a brief verbal description, a freehand sketch that may be quite rough, and a set of measurements that may be incomplete and inconsistent. Designers produce large numbers of technical sketches, of which only a few are selected for sampling. The designers' tasks are in three distinct categories. *Selection*: designers select yarns, inspirational material, and old designs. *Modification*: designers translate, adapt, and combine elements of previous designs and sources of inspiration. (Few garments are designed entirely from scratch.) *Mathematics*: designers need to perform computational tasks such as working out pattern repeats and constructing garment shapes.

Knitting machine technicians program industrial knitting machines using highly sophisticated visual programming environments, which contain expert systems that translate designs specified in a graphic symbolic notation into programs in an assembler-like language. The different knitting machine manufacturers produce CAD systems using different graphic representations and machine instruction languages. The programs are tailored to the capabilities of a particular knitting machine. The technicians require a thorough understanding of yarn properties and knitting machines, and considerable technical skill in implementing knitting-time-efficient programs.

5.2. The communication bottleneck

Communication between designers and technicians is a major bottleneck in knitwear design, and failures of communication result in a great deal of inefficiency and the production of unsatisfactory designs. Designers' technical sketches are frequently *inaccurate*, in that they do not correspond to the designers' intentions; *incomplete*, in that parts of the specification are missing; and *inconsistent*, in that interdependent values are contradictory. Moreover, technicians have commented to us that only 30% of conceptual designs can be manufactured at their intended price point. This communication breakdown has many causes including working arrangements that make it difficult for designers and technicians to talk frequently (Eckert, 1997*a, 1999; Eckert & Demaid, 1997; Eckert et al., in press).*

The technicians need to interpret the designers' intentions, to do what the designers have asked for, to fill in the gaps, and to make sensible compromises between what the designers want and what is technically feasible. But designers and technicians use their different knowledge and experience to interpret design information differently. Technicians think primarily about the structural properties of knitted garments in terms of the notation used by their CAD systems, while designers usually visualize designs as complete garments, and so think primarily about their appearance. Designers are much more aware of changes in current fashion than are technicians, so their use of category terms is based on current norms, while that of the technicians is based on designs they have produced previously. This has the consequence that garments often vary less between seasons than their designers intend.

Technicians' interpretations of designs differ from their designers' intentions because the designers have no cost-effective way to make their intentions clear. It is difficult for them to indicate deviations from category norms on technical sketches: It is hard to indicate which bits of freehand sketch are exact and which are merely approximate placeholders for category information; and technicians frequently ignore the sketches altogether (Stacey et al., 1999). Similarly, as the measurements given to technicians are frequently faulty, technicians often treat deviations from standard values as mistakes. Designers sometimes deliberately provide incomplete sets of measurements to force their technicians to talk to them.

Technicians usually produce programs and prototypes from the technical sketches with little interaction with the designers. The only feedback designers usually get on a technical sketch is a finished sample garment several weeks later. As many of these garments are very different from their intentions, the designers feel ignored, while the technicians do not trust the designers' assertions.

The communication problem would be alleviated by a cost-effective means for designers to produce complete, consistent, technically correct conceptual designs, which clearly indicate deviations from standard values.

5.3. Objectives for intelligent computer support for knitwear design

The system is designed to compensate for designers' sometimes inadequate skills in garment shape construction, and quite rational unwillingness to devote time to generating correct sets of measurements when producing many designs quickly under time pressure. But it relies on the designers' well-honed abilities to recognize good designs when they see them, and to recognize similarities and differences between garment designs from pictures (Fig. 4).

Graphical editing of garment shapes requires a tool with the following features:

- Shape curves that always conform to domain constraints.
- Shape curves that can be configured to conform to customs and company standards.
- Shape curves that are constrained to be consistent with other parts of the design.
- The ability to switch between making changes that are consistent with the rest of the design, and making changes that freely violate consistency requirements.

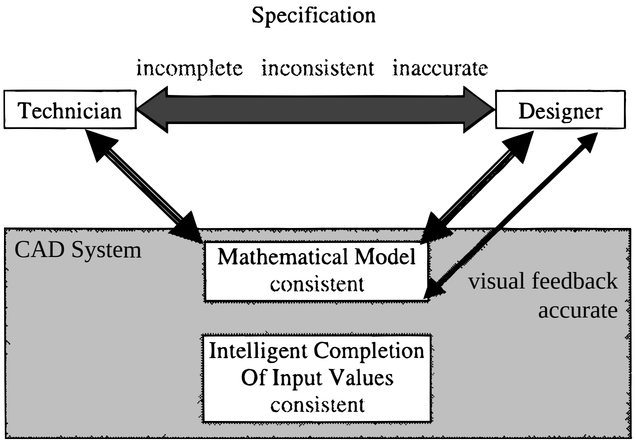


Fig. 4. Overcoming the communication bottleneck.

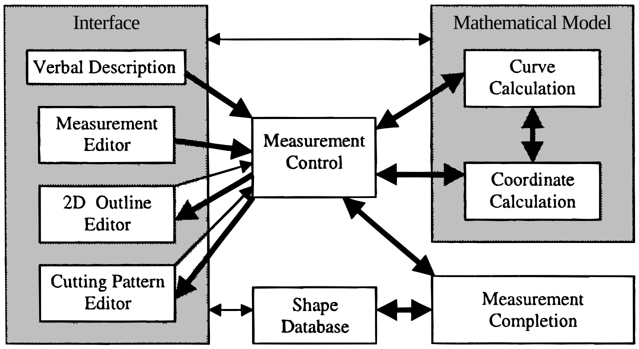


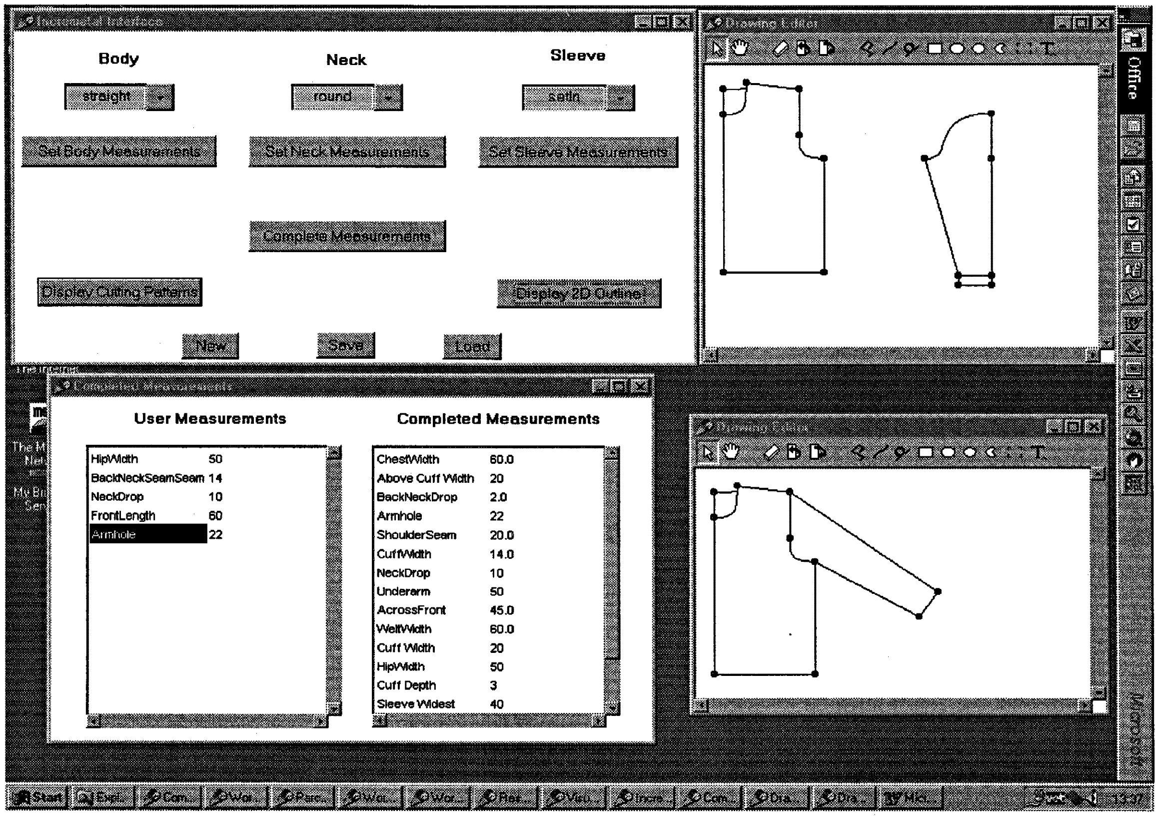
Fig. 5. System overview.

- Visually appealing curves that look right to the user.
- An interface that is reasonably intuitive for a user with no knowledge of the underlying mathematics.
- Curves that are generated using the minimum number of assumptions that are not derived directly from the users' input.

Displaying and editing garment shapes graphically requires a complete and consistent description, which, as we pointed out in Section 5.2, is not to be taken for granted.

The first author has implemented a prototype design support system (described in Fig. 5) in VisualWorks 3.0, a dialect of Smalltalk-80. This is intended to enable knitwear designers to create garment shape designs that are complete, consistent, and technically correct, and provide technicians with designs that leave no scope for misinterpretation. It should eliminate the need to perform the algorithmic subtasks involved in shape construction that knitwear designers find difficult and tedious, while ensuring conformance to current fashion and company standards. Informal evaluation with design experts has been positive.

The system uses the representations, categories, and terms used in the knitwear industry. It displays the garment in three representations that designers are accustomed to: measurements, cutting patterns, and two-dimensional outlines. The system enables designers to identify the category of a garment using the terms they would use to give a short verbal description, for example "straight tunic with set-in sleeve and round neck" (this corresponds to the example shown in Fig. 6). This is done by selecting the categories of the body, sleeves, and neck from pull-down menus. [A complete system would begin with a selection of the general category of garment ("pullover," "cardigan," and so on), and display menus for the corresponding set of shape elements. A system for tailoring would need to include darts.]



User Measurements

HipWidth	50
BackNeckSeamSeam	14
NeckDrop	10
FrontLength	60
Armhole	22

Completed Measurements

ChestWidth	60.0
Above Cuff Width	20
BackNeckDrop	2.0
Armhole	22
ShoulderSeam	20.0
CuffWidth	14.0
NeckDrop	10
Underarm	50
AcrossFront	45.0
WeltWidth	60.0
Cuff Width	20
HipWidth	50
Cuff Depth	3
Sleeve Widest	40

Fig. 6. Example of output generated by the shape design system.

The user can define all or some of the measurements of the garment by filling in numbers on a form. In the example shown in Figure 6 (in the bottom left-hand window) the user has only defined five measurements; the rest have been completed automatically by the system.

5.4. Automatic design: Measurement completion

The basic representation for each design is a set of measurements (left bottom window in Fig. 6). If the set of measurements is incomplete, the system fills in the remaining measurements, by choosing default values according to the verbal description (that is, the category selections). The present implementation of the system uses one set of measurements for each combination of sleeve and side line, and one set of measurements for the neckline.

We intend to develop the system to use more sophisticated rules for adapting default values to fit explicit measurements, a mechanism for creating company-specific defaults, and templates, rules for adapting defaults, and explicit measurements to correspond to body shape measurements, and case-based reasoning for completing sets of measurements “in the style of” another garment.

5.5. Automatic design: Shape construction

The set of measurements is developed into a garment shape description by first translating the measurements into a set of Cartesian coordinates for each garment part, and then joining these coordinates with straight lines and Bézier curves (Bézier, 1968; Faux & Pratt, 1979). Bézier curves have the twin advantages that direct manipulation by moving control points is fairly intuitive, as the curve moves in the same direction as the point; and that mathematical computation of the curves from the control points is fairly simple. Bézier curves give easy control over the end tangent vectors, which are important domain constraints. Moreover, Bézier curves correspond much more closely than spline curves to the curves constructed by professional designers; commercial systems for garment shape modeling using spline curves produce unsatisfactory results.

The internal representation of each garment part is a list of coordinate sets, labeled by the names of the lines they describe, containing the measurement-derived end points and the Bézier control points for each line. The Bézier curves are calculated using a mathematical model of garment shapes that incorporates the constraints that garment shape curves must meet, which are more restrictive for knitwear than for tailored garments (Eckert, 1997*a*). This calculation uses a novel method for generating Bézier curves with a particular overall length (Eckert, 1997*a*), which is used to ensure consistency between different curves that need to have the same length (namely armholes and sleeve crowns).

As in industrial practice, the sleeve is calculated after the body shape. If there is a mathematical inconsistency in the measurements between different garment parts, the sleeve will be altered to match the body because a designer is more likely to specify the overall dimensions of the garment, that is, the body dimensions, correctly. The user can identify a consistent but nonsensical specification (e.g., an extremely long sleeve) by looking at the two-dimensional outline of the suggested garment.

5.6. Interactive shape design using mathematical models

The automatically generated garment shapes are intended to be used as starting points for design by modification. We intend to extend the system to include a graphic interface that will allow the user to modify garment shapes directly by towing around control points in the cutting pattern and two-dimensional outline representations, as well as by tweaking measurements. In this version, the automatic design completion module will be used at the beginning of the shape design process, to create an initial design from partial or inconsistent inputs for further editing, and during the editing process to resolve conflicts when the designer deliberately chooses to make a change that is inconsistent with other aspects of the current design.

5.7. Anticipated influence on the design process

The system enables designers to define garment shapes within a few minutes. A technical sketch can then be generated automatically; this is usually a time-consuming process. As the specified garment shape is technically correct, it will lead to shorter sampling times for each garment, or to more refined designs. So far, all technical innovations in the knitwear industry have led to more complex products rather than to cheaper sampling.

6. DIALOGUE WITH COLORS

Kelly (1999) has implemented a family of evolutionary systems to assist artists and designers in selecting color combinations, when they are chosen for their aesthetic and expressive value. These systems work by allowing the user to choose a succession of randomly generated color schemes with progressively smaller variations. The user does not have to worry about constructing a color scheme incrementally, but only evaluates complete schemes. The different evolutionary systems use the same design generation algorithm, but use different mechanisms for specifying constraints to be imposed on the design generation process. These different types of constraints correspond to different ways of formulating the perceptual properties that a color scheme should have, so that the suite of systems together constitutes a flexible tool for generating color designs with different desired characteristics.

Appropriate color selection is an important consideration whenever psychological responses to the designed artifact are an issue. Where the product is subject to fashion trends, inappropriate color selection many jeopardize the commercial success of the whole project. Color design is demanding even for professional designers who specialize in this field. Humans can discriminate a large number of color sensations, which can be combined into a vast range of color schemes. Further, the expressive effect that they produce is complex and not yet fully understood. Modern color technology and its underlying color science (see Hunt, 1995) are complex and involve ways of specifying colors that may not be easily understood, or may be regarded as counterintuitive by designers. Color CAD software allows the designer to work in a way that is more conducive to human creativity, but still produces a design specification in the notation required by the technology involved.

The aim of Kelly's (1999) research has been to produce generic tools that exploit the findings of color science and psychophysics, as well as practitioner lore in art and design, in helping designers with this color selection task. The systems are concept demonstrators for methods to be used in task-specific CAD systems. However, generic tools of this type may well be useful in situations where a color consultant must design a palette of colors that will visually unify a group of products that have not yet been designed. For instance, this is common practice in the clothing and interior décor industries, where yarn manufacturers decide on their color palettes according to fashion forecasts before their clients start designing. An application to specifying the text and background colors of World Wide Web pages would need to use algorithms modified to ensure that only color combinations containing color differences large enough to afford good text visibility would be presented to the users.

6.1. The color scheme design problem

Color design is not just a matter of preference, pleasantness, and like and dislike. It also involves issues of the appropriateness of the meaning expressed by colors. Should the design be warm or cool, dynamic or tranquil, restrained or extravagant, or any of the other meanings that colors can carry? The space of possible color designs is vast: The number of perceptibly different colors is about 1,350,000 for practical colorants (Pointer & Attridge, 1980). However, the range of available colors may be far smaller in practice because the available technology provides a finite number of alternative colors, and because it simply cannot produce certain colors. Importantly, computer screens are restricted to the gamut of colors that can be produced by mixing red, green, and blue (RGB) light, which is only a subset of the colors humans can see.

Color design is made more problematic by the complex and varying relationship between color stimuli and the sensations that they produce. Color stimuli change their appearance when seen in varying contexts. In art and design this concept is embraced by terms like color interaction and color dynamics, while in science and engineering it is encompassed by the term color appearance modeling. In conventional color design methodology, this means that the designer must make initial color choices while imagining their appearance in the whole that has yet to emerge. Subsequent editing of the colors alters the appearance of the whole design, not just the color that is explicitly altered. In most cases, the later stages of the color design processes are biased by the first selections. In practice, this problem is often circumnavigated by selecting colors from existing combinations found in objects or images (Eckert, 1998).

In color design, designers formulate what they want to achieve in terms of emergent properties of the whole design. But what can be manipulated are the structural properties of a display: For computer monitors, the RGB values of different regions or mathematical transformations of them (though in some cases other structural properties like the sizes and proportions of regions can be altered at the same time). The challenge then is to map emergent properties and constraints onto sets of RGB values.

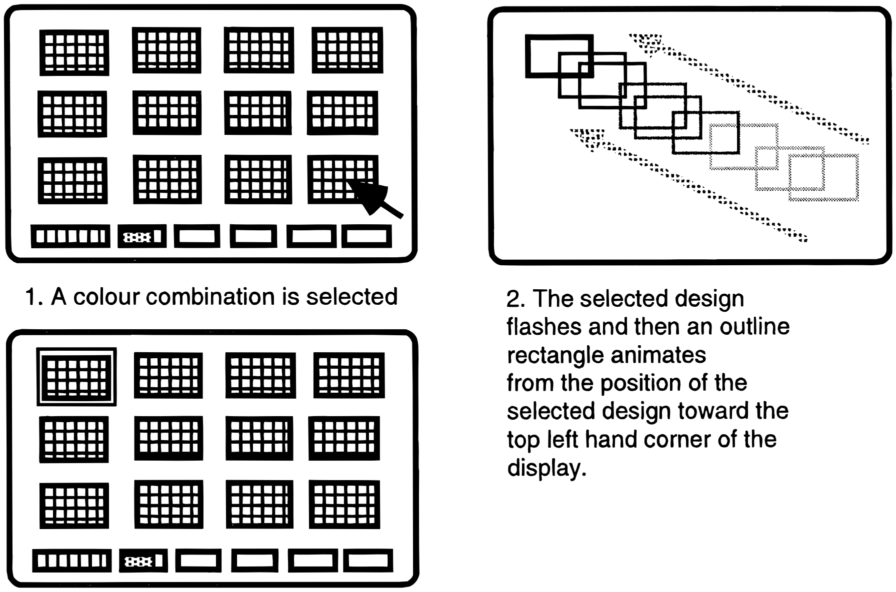
6.2. Guided random generation of color schemes

Kelly's (1999) color scheme generator (illustrated schematically in Fig. 7) represents color schemes as sets of RGB values. These are displayed on the screen as random patterns of colored squares; the user can select one of these for further development. Random vectors are added to the RGB values of the current selected design to create a set of new designs. This cycle is repeated until the user picks a satisfactory color scheme; the size of the random vectors decreases at each stage. From the user's perspective, this process is analogous to Darwinian evolution, with the user determining fitness for survival and reproduction. This evolutionary approach took the work of Dawkins (1986), Sims (1991), and Todd and Latham (1992) as its point of departure. Dawkins developed interactive computer programs to produce a graphic illustration of the power of repeated selection in Darwinian evolution. Sims and Latham and Todd took Dawkins' basic idea and applied it to the design of complex three-dimensional computer graphic scenes. Their work includes, but does not specialize in, color design.

It has not proved possible to find a universally satisfactory rate at which the reduction in variation should take place, and so the user can override the automatic calculation of vector size using an on-screen slider [following Todd and Latham (1992)]. The system could be extended to enable faster convergence on a final design using *evolutionary direction* (Todd & Latham, 1992): Two designs are selected to define a vector in the design space; subsequent designs are constrained to lie on the vector, between the two original designs or beyond the second. A design generation method based on genetic crossover could be used to prolong the search.

6.3. Using an evolutionary system for color design

As the space of possible color designs is vast, and the relationship between colors and aesthetic effects is subtle and



1. A colour combination is selected

2. The selected design flashes and then an outline rectangle animates from the position of the selected design toward the top left hand corner of the display.

3. Previously selected colour combination is displayed in the top left of the display with a box drawn arround it. The remaining locations are filled with new colour combinations created by variations on the previously selected design.

4. Stages 1, 2 & 3 are iterated until a satisfactory design is generated.

Fig. 7. Interactive evolution of color schemes.

can only be predicted to a limited degree, color design is well suited to an evolutionary approach in which designers home in on the scheme they want by repeatedly selecting from a range of alternatives generated by an evolutionary design system. The evaluation and selection of color schemes is initially based on the recognition of faint resemblances to the user's design intentions, and, later on, subtle discrimination between similar strong candidates for a final solution. This is consistent with good practice in art and design. This approach allows rapid navigation of the design space, and only requires the designer to appraise complete color schemes, so avoiding the problems of perceived color changing with visual context and designs being biased by the order in which designers make decisions.

But there is an uneven relationship between human perception of colors and the physical color space describable in RGB values; for instance, the green region of the physical color map is far larger than the yellow region. So using the physical color map without correction for the characteristics of human color perception biases a color generation system strongly toward certain regions of the perceptual color space. However, a color map that is completely unbiased with respect to human perception is unobtainable because the characteristics of the perceptual color space varies with context and task. For instance, the map obtained by measuring small perceptually equal color differences differs from that obtained by measuring large perceptually equal color differences. The field of colorimetry uses several numerical notations for colors, which can be computed from each other by a sequence of mathematical transformations. For instance, CIE LAB is reasonably perceptually even with respect to small color differences. The current implementation of Kelly's system uses raw RGB values, which entail a physical color map that is convenient from the perspective of computer graphics but strongly biased with respect to human perception. Kelly (1999) has considered the use of mathematical transformations of RGB values, such as CIE LAB, to obtain a perceptually less biased color space. The disadvantage of this is that it represent the whole of the color space humans can see with positive values of the dimensions of Lightness, A (roughly redness-greenness) and B (roughly blueness-yellowness), so the color scheme generation system would specify colors that computer monitors cannot display.

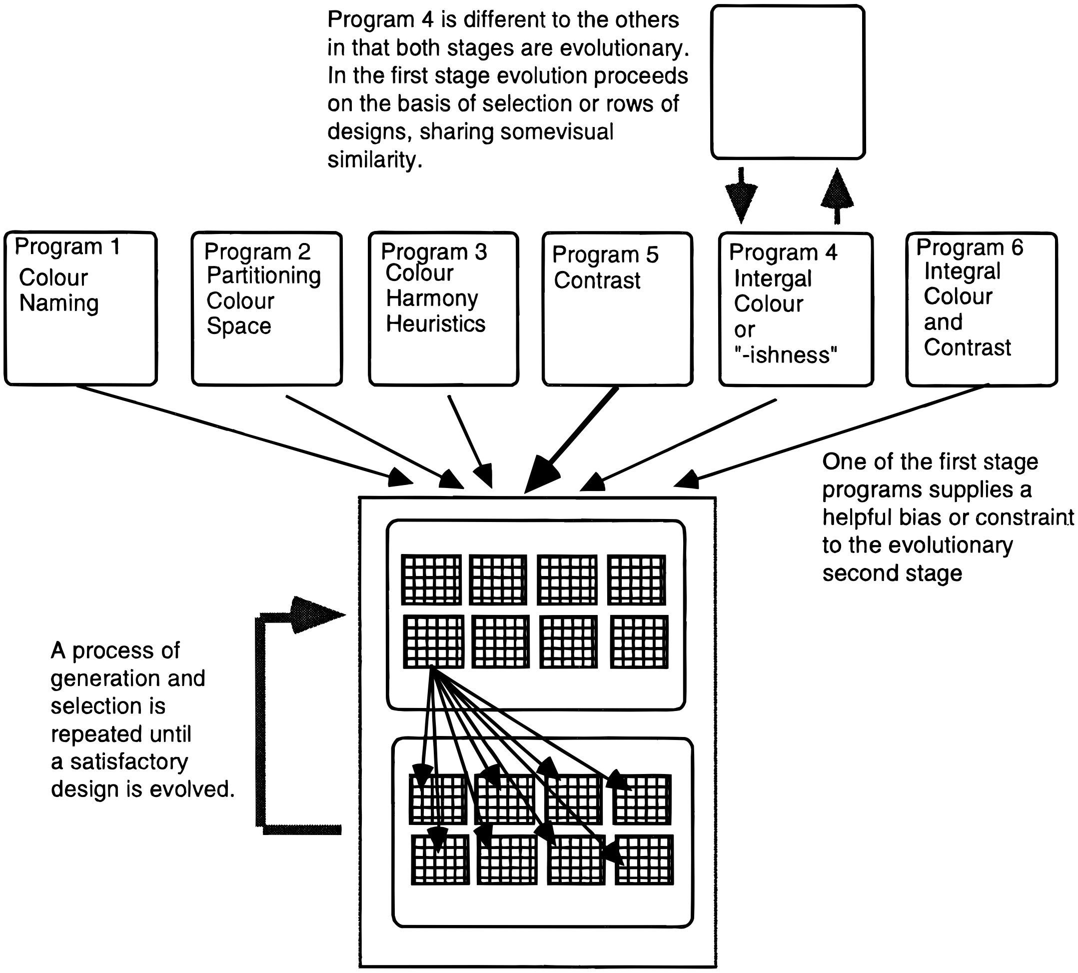
People frequently want color schemes that are extreme in one respect or another, for instance having very high contrast, and so are difficult or slow to reach using a method that searches the color space evenly. So the generation of color schemes by an evolutionary system for consideration by a user should be strongly biased. It should be biased toward generating color schemes that satisfy the constraints and the evaluatable emergent properties that the users require. It should also minimize undesired bias toward particular parts of the perceptual color map. Kelly's first color generation system, described in Section 6.2, was criticized in user trials because it was insufficiently biased: Designers who had a vague idea in mind often found it impossible to identify a promising candidate for refinement from a set of equally hopeless choices. It was also criticized because it was biased in the wrong ways: It proved difficult to obtain color schemes that had notably high or low contrast; or that inclined to one color without excluding others, for instance, designs that were "reddish" or "whiteish." Experimentation and theoretical analysis revealed that it was not possible to make all the desirable extreme designs easy to get at with one algorithm; making one type easy to reach aggravated or introduced other kinds of biases.

6.4. Programming biases

Kelly (1999) has addressed the problems described in Section 6.3, by developing six two-stage evolutionary color scheme generation systems. Each of these systems has a preliminary stage in which the user programs in a bias of a different kind, that the iterative color scheme generator uses to produce color schemes with particular desired characteristics. In the first stage of all six programs, the user is given immediate visual feedback by the display of a number of designs that fit the current specification. The relationship between the bias programming stage and the evolutionary design stage is shown in Figure 8.

6.4.1. Program 1—Color naming

The user selects and deselects regions in color space approximately relating to the eleven basic color naming categories identified by Berlin and Kay (Berlin & Kay, 1969; Hardin & Maffi, 1997). This is done by clicking on rectangles showing typical colors within each category. In doing this the search space is expanded or contracted, according to the number of categories selected. In the iterative stage, color schemes containing colors lying outside these regions are simply discarded. Both the presence of these color categories and the historical order of their appearance (black



Program 4 is different to the others in that both stages are evolutionary. In the first stage evolution proceeds on the basis of selection or rows of designs, sharing somevisual similarity.

One of the first stage programs supplies a helpful bias or constraint to the evolutionary second stage

A process of generation and selection is repeated until a satisfactory design is evolved.

Fig. 8. Actively biased evolutionary design of color schemes.

and white; red, yellow, and green; blue, brown, purple, and pink; and orange and gray) is surprisingly consistent in a large number of languages (Berlin & Kay, 1969; Hardin & Maffi, 1997), and probably reflects the underlying neurophysiology of color vision. In this way the program exploits a function of color cognition that is highly familiar to its users, be they professional designers or nonexperts. The program uses deliberate imprecision. In the first stage, the user makes specifications of the form: "I want red and yellow, but I don't yet know which red or yellow." The second evolutionary stage removes this imprecision. Colors that do not belong to the specified naming categories are simply culled and never presented to the user.

6.4.2. Program 2—Partitioning color space.

This works in the same way as the color-naming program. However, in this case the user selects from 64 divisions of RGB color space. The approach still involves the deliberate imprecision of the color-naming program, but here the scale of the imprecision is reduced by the larger number of categories controlled.

6.4.3. Program 3—Color harmony heuristics

The heuristics this program uses is the 19th and early 20th century idea that the appearance of harmony results from colors sharing particular characteristics. [Todd and Latham (1992) comment on the idea drawn from genetics that if a feature or parameter value does not vary, either it is unimportant or the value is *right*.] The users first select a dimension of color appearance that they wish to control, and then set a value that all the colors in the design will share. The choices are hue, lightness, chroma (which is similar to saturation), and derived dimensions that approximate to perceptual whiteness and perceptual blackness. In the evolutionary second stage, the RGB values for each color are used to compute values for lightness, hue, and chroma in the CIE Lch_{ab} color space (CIE, 1986). The value of one dimension is fixed while the other two are varied. The new color is translated back to the RGB color space and checked to ensure it is within the color gamut of a monitor (i.e., it can be created from red, green, and blue light).

6.4.4. Program 4—Integral color

The user chooses a color that they wish the color scheme, as a whole, to incline toward, without necessarily excluding small amounts of other colors. In effect they choose between redishness, blackishness, greenishness, and so on. This is implemented by specifying the integral color, which is the color that results from averaging the RGB values of the colors in the design. This technique is borrowed from the photographic industry (Kodak, 1947) . Unlike the other programs, this system uses an evolutionary process for bias selection. Evolution proceeds by selection between rows of color schemes that share the family resemblance, which results from having the same integral color. In the second stage new designs are generated by adding small random vectors to the color data, and then correcting the design to restore the specified integral color. This correction is achieved by a process of iterated optimization in which the distance in RGB space between the current and specified integral colors provides the cost function. This program was found to make it difficult to produce highly contrasting color schemes.

6.4.5. Program 5—Contrast

The user adjusts a slider that controls "contrastiness," that is, some measure of the total color variance within the design. The new designs generated by the evolutionary system are adjusted by a process of iterated optimization to have the desired contrastiness. The algorithm used here reflects but does not exactly follow several models to be found in the color science literature (Feldman, 1993; Soen et al., 1987; Luciano & Ronchi, 1989). Contrastiness is derived from the root mean square of the color differences between each of the colors in the design and the integral color of the whole, in an approximately uniform color space known as Hunter LAB (Hunter & Harold, 1987). This program made it difficult to control general inclination to some color.

6.4.6. Program 6—Integral color plus contrast

The user adjusts four sliders to set the bias, three of which control the hue, chroma, and lightness values of the integral color, and one of which controls the contrastiness. New designs are adjusted to the desired integral color and contrastiness using iterated optimization. This program ran noticeably slower than any of the others.

These six programs were compared in a trial in which professional designers and amateurs were observed and then given a questionnaire. Designing color schemes by interactive evolution was quickly and easily learned, and was valued as an effort-effective method of color design. The users preferred the programs that gave the most control, especially those based on color naming and the partitioning of color space. Users reported two conflicting ways in which they found the programs irritating or unhelpful: They were coerced toward a final design too quickly; and they were obstructed from moving quickly toward a strong idea they had in mind. This suggests that this aspect of the interaction should be placed under user control.

6.5. Programming emergent properties

A more ambitious idea that Kelly (1999) has developed theoretically but not so far implemented, is that the first stage of a two-stage system should allow the designer to specify the expressive character of the color scheme. Useful findings from psychology have been identified in Kelly (1999) that indicate that broad categories of adjectival meaning can be identified that are elicited by similar colors or color combinations. These can be summarized by a small number of semantic dimensions, such as warmth, activity, potency (strength), and pleasantness (Sivik, 1974; Kobayashi, 1981). Kelly (1999) has implemented goodness functions that approximately model warmth, activity, and (less well) strength. A functioning program of this kind is a matter for further research and development.

7. CONCLUSION

Analyses of human design thinking and the social organization of design processes, and the capabilities of generative systems for design, indicate that interactive generative systems can be powerful tools that increase the creativity and productivity of human designers. The success of existing interactive generative systems in art and design increases our confidence that there are places for interactive generative systems in a wide variety of design tasks, and that such systems can be easy, intuitive, and enjoyable to use as well as effective.

Computer designers and human designers have the potential to complement each others' strengths and compensate for each others' weaknesses. Automatic design systems are good at performing algorithmic subtasks that humans find difficult or tedious, handling large numbers of constraints, and systematically exploring a design space to generate large numbers of alternative designs. Humans are good at performing perceptual evaluations of the quality of designs using criteria that can be precisely tailored to the demands of each task and that are extremely difficult to program, so they can act as a generative system's fitness function.

Generative systems fit more naturally into human design behavior than one might expect. Much designing (in some fields nearly all) is selecting and adapting promising starting points: Generative systems can provide these while exploring the consequences of designers' initial decisions. Designers are provoked to generate new ideas by stimuli that change and enrich their mental context, especially those relevant and unexpected, such as interesting-looking mistakes.

Interactive generative systems offer ways to alleviate some significant problems encountered by individual designers and design teams.

- By generating many alternatives, they can enable the exploration of a larger fraction of the space of possible designs, so alleviating the tendency of designers to use their first good idea rather than look for a better one.
- By presenting relatively complete designs for human evaluation, they can alleviate biases on the design process due to the order in which decisions get made.
- They can help members of design teams communicate by presenting relatively complete designs that have been checked for technical correctness, so eliminating obvious errors in specifications and overcoming the limitations of inadequate notations.

Generative systems can be lightweight, general-purpose tools, like Kelly's suite of evolutionary systems for color design, or special purpose systems integrated into a single industry's working methods, like Eckert's garment shape design system. But the development of any effective system for commercial use requires a clear understanding of the designers' task, and an understanding of the industrial context in which it will be embedded. This will require empirical study, but the large body of research on design psychology and design processes provides a solid foundation.

ACKNOWLEDGMENTS

Claudia Eckert's research has been supported by SERC ACME grant No. GR/J 40331 for the knitwear project at the University of Loughborough Department of Computer Studies; by ESRC grant No. L12730100173 for the MIND project at the Open University Computing Department; and by Open University Research Development Fund grant No. 717 at the Open University Department of Design and Innovation. Graham Perkins of the Department of Computer and Information Sciences, De Montfort University, Milton Keynes, contributed valuable technical advice and guidance to the implementation of the shape design system. Ian Kelly's research has been supported by an EPSRC studentship at the Open University Computing Department. Three anonymous reviewers made valuable comments on an earlier version of this paper.

REFERENCES

Akin, Ö. (1987). *Psychology of architectural design*. Pion, London, UK.

Asimow, W. (1962). *Introduction to design*. Prentice-Hall, Englewood Cliffs, NJ.

Berlin, B., & Kay, P. (1969). *Basic color terms*. University of California Press, Berkeley, CA.

Bézier, P. (1968). *How Renault uses numerical control for car body design and tooling*. Paper SAE 6800010, Society for Automotive Engineers.

Bracewell, R.H. & Sharpe, J.E.E. (1993). Application of bond graph methodology to concurrent conceptual design of multidisciplinary systems. *Proc. IEEE Conf. Sys., Man and Cybernet. '93*.

Bracewell, R.H., & Sharpe, J.E.E. (1996). Functional descriptions used in computer support for qualitative scheme generation—"Schemebuilder." *Artificial Intelligence in Engineering Design, Analysis and Manufacture 10(4)*, 333–346.

Chakrabarti, A., & Tang, M. (1996). Generating Conceptual Solutions on FuncSION: Evolution of a functional synthesiser. In *Artificial Intelligence in Design '96*, (Gero, J.S. and Sudweeks, F., Eds.), pp. 603–622. Kluwer Academic Publishers, Dordrecht, Holland.

Chase, S.C. (1998). User interaction models for grammar based design systems. Proceedings of Design Computing on the Net '98. *International Journal of Design Computing 1*. http://www.arch.usyd.edu.au/kcdc/journal/.

CIE (1986). *Colorimetry, 2nd. ed.* Document 15.2, Commission Internationale De L'Eclairage.

Cross, N.G. (1989). *Engineering design methods*. John Wiley, Chichester, UK.

Dawkins, R. (1986). *The blind watchmaker*. Longman, Harlow, UK.

Eckert, C.M. (1997*a*). *Intelligent support for knitwear design*. PhD Thesis. Department of Design and Innovation, The Open University, Milton Keynes, UK.

Eckert, C.M. (1997*b*). Design inspiration and design performance. *Proc. 78th World Conference of the Textile Institute*, vol. 1, pp. 369–388. Textile Institute, Manchester, UK.

Eckert, C.M. (1998). *Sources of inspiration in knitwear design*. Report No. 98/10. The Open University, Milton Keynes, UK.

Eckert, C.M. (1999). *The communication bottleneck in knitwear design: Analysis and computing solutions*. Report No. CUED/C-EDC/TR71. Cambridge University Engineering Design Centre (submitted for publication).

Eckert, C.M., & Demaid, A. (1997). Concurrent design. *Proc. 78th World Conf. Textile Inst.*, vol. 3, pp. 101–122. Textile Institute, Manchester, UK.

Eckert, C.M., Cross, N.G., & Johnson, J.H. (in press). Overcoming communication difficulties in the design process by intelligent support for conceptual design. *Design Studies 21*.

Eckert, C.M., & Stacey, M.K. (1998). Fortune favours only the prepared mind: Why sources of inspiration are essential for continuing creativity. *Creativity and Innovation Management 7(1)*, 9–16.

Eckert, C.M., & Stacey, M.K. (1999). Sources of inspiration: A language of design. *Proc. 4th Design Thinking Res. Symposium*, Massachusetts Institute of Technology, Cambridge, MA.

Faux, I., & Pratt, M. (1979). *Computational geometry for design and manufacture*. Ellis Horwood, Chichester, UK.

Feldman, U. (1993). *Quantifying the dimensions of color experience*. PhD Thesis. Massachusetts Institute of Technology, Cambridge, MA.

Finke, R.A. (1990). *Creative imagery: Discoveries and inventions in visualization*. Lawrence Erlbaum, Hillsdale, NJ.

Fischer, G., Nakakoji, K., Ostwald, J., Stahl, G., & Sumner, T. (1993). Embedding critics in design environments. *Knowledge Engineering Review 8(4)*, 285–307.

Goel, V. (1995). *Sketches of thought*. MIT Press, Cambridge, MA.

Goldschmidt, G. (1991). The dialectics of sketching. *Creativity Research Journal 4(2)*, 123–143.

Goldschmidt, G. (1992). Serial sketching: Visual problem solving in designing. *Cybernetics and Systems: An International Journal 2*, 191–219.

Goldschmidt, G. (1994). On visual design thinking: The vis kids of architecture. *Design Studies 15(2)*, 158–174.

Goldschmidt, G. (1995). Visual displays for design: Imagery, analogy and databases of visual images. In *Visual Databases in Architecture: Recent Advances in Design and Decision-Making*, (Koutamanis, A., Timmermans, H. and Vermeulen, I., Eds.), pp. 53–76. Avebury, Aldershot, UK.

Green, T.R.G. (1989). Cognitive dimensions of notations. In *People and Computers IV*, (Sutcliffe, A. and Macaulay, L., Eds.), Cambridge University Press, Cambridge, UK.

Green, T.R.G. (1991). Describing information artefacts with cognitive dimensions and structure maps. In *People and Computers VI*, (Diaper, D. and Hammond, N.V., Eds.), Cambridge University Press, Cambridge, UK.

Green, T.R.G., & Petre, M. (1996). Usability analysis of visual programming environments: A 'cognitive dimensions' framework. *Journal of Visual Languages and Computing 7*, 131–174.

Hardin, C.L., & Maffi, L. (1997). *Color categories in thought and language*. Cambridge University Press, Cambridge, UK.

Hunt, R.W.G. (1995). *The reproduction of colour*. Fountain Press, Kingston-upon-Thames, UK.

Hunter, R.S., & Harold, R.W. (1987). *The measurement of appearance*, 2nd ed. John Wiley, New York.

Karnopp, D.C., Margolis, D.L., & Rosenberg, R.C. (1990). *System dynamics: A unified approach*, 2nd ed. John Wiley, Chichester, UK.

Katz, I. (1994). Coping with the complexity of design: Avoiding conflicts and prioritizing constraints. *Proc. Sixteenth Annu. Meeting Cognitive Sci. Soc.* Lawrence Erlbaum, Hillsdale, NJ.

Kelly, I. (1999). *An evolutionary interaction approach to computer aided colour design*. PhD Thesis. Department of Computing, The Open University, Milton Keynes, UK.

Knight, T.W. (1994). *Transformations in design: A formal approach to stylistic change and innovation in the visual arts*. Cambridge University Press, Cambridge, UK.

Kobayashi, S. (1981). The aim and method of the color image scale. *Color Research & Application 6(2)*, 93–107.

Kodak (1947). *Improvements in or relating to a method and apparatus for making photographic colour prints*. British Patent No. 660,099.

Kolodner, J. (1993). *Case-based reasoning*. Morgan Kaufmann, San Mateo, CA.

Luciano, D.F., & Ronchi, L.R. (1989). On a possible assessment of the perceptual environmental color. *AIC—Color '89*, p. 187.

McFadzean, J., Cross, N.G., & Johnson, J.H. (1999). Notation and cognition in conceptual sketching. *Proc. VR'99, Visual and Spatial Reasoning in Design*. MIT Press, Cambridge, MA.

Peterson, M.A. (1993). The ambiguity of mental images: Insights regarding the structure of shape memory and its function in creativity. In *Imagery, Creativity and Discovery: A Cognitive Approach*, (Roskos-Ewoldsen, B., Intons-Peterson, M.J. and Anderson, R.E., Eds.), pp. 151–186. Elsevier, Amsterdam.

Pointer, M.R., & Attridge, G.G. (1980). The gamut of real surface colors. *Color Research and Application 5(3)*, 145–155.

Purcell, A.T., & Gero, J.S. (1998). Drawings and the design process. *Design Studies 19*, 389–430.

Purcell, A.T., Gero, J.S., Edwards, H., & McNeill, T. (1994). The data in design protocols: The issue of data coding, data analysis in the development of models of the design process. In *Artificial Intelligence in Design '94*, (Gero, J.S. and Sudweeks, F., Eds.), pp. 225–252. Kluwer Academic Publishers, Dordrecht, Netherlands.

Schön, D.A. (1983). *The reflective practitioner: How professionals think in action*. Basic Books, New York.

Schön, D.A., & Wiggins, G.A. (1992). Kind of seeing and their function in designing. *Design Studies 13(2)*, 135–156.

Silverman, B.G. (1992). Survey of expert critiquing systems: Practical and theoretical frontiers. *Communications of the ACM 35(4)*, 106–127.

Simon, H.A. (1973). The structure of ill-structured problems. *Artificial Intelligence 4*, 181–201.

Simon, H.A. (1981). *Sciences of the artificial*. MIT Press, Cambridge, MA.

Sims, K. (1991). Artificial evolution for computer graphics. *ACM Transactions on Computer Graphics 25*, 319–328.

Sivik, L. (1974). *Color meaning and perceptual dimensions: A study of color samples*. Göteborg Psychological Report, Vol. 4, No. 1, University of Göteborg.

Soen, T., Shimada, T., & Akita, T. (1987). Objective evaluation of color design. *Color Research & Application 12(4)*, 187–195.

Stacey, M.K., & Eckert, C.M. (1999). CAD system bias in engineering design. *Proc. 12th Int. Conf. Eng. Design*, Vol. 3, pp. 1413–1418. Technical University of Munich, Munich, Germany.

Stacey, M.K., Eckert, C.M., & McFadzean, J. (1999). Sketch interpretation in design communication. *Proc. 12th Int. Conf. Engin. Design*, Vol. 2, pp. 923–928. Technical University of Munich, Munich, Germany.

Stacey, M.K., Rzevski, G., Sharpe, H.C., Petre, M., & Buckland, R.A. (1997). Intelligent support for conceptual design: A flow modelling approach. *Proceedings of the 11th International Conference on Engineering Design, vol. 3*, pp. 261–266. Heurista, Tampere, Finland.

Stiny, G. (1980). Introduction to shape and shape grammars. *Environment and Planning B: Planning and Design 7*, 343–351.

Stiny, G., & Mitchell, W.J. (1978). The Palladian grammar. *Environment and Planning B: Planning and Design 5*, 5–18.

Suwa, M., Gero, J.S., & Purcell, T. (1998). Analysis of cognitive processes of a designer as the foundation for support tools. In *Artificial Intelligence in Design '98*, (Gero, J.S. and Sudweeks, F., Eds.), pp. 229–247. Kluwer Academic Publishers, Dordrecht, Netherlands.

Todd, S., & Latham, W. (1992). *Evolutionary art and computers*. Academic Press, London.

Claudia Eckert's research on intelligent support systems for knitwear design began with her M.Sc. in Applied Artificial Intelligence at the University of Aberdeen. This work has led to two research council funded projects and a Ph.D. in Design Studies at the Open University. She is now a research associate in the Engineering Design Centre at the University of Cambridge, working on intelligent signposting techniques for guiding engineering design processes. Her research interests include design process modelling and design communication.

Ian Kelly worked as an art teacher in Brighton for 18 years before taking an M.A. in Computing in Design at the University of Middlesex, and going on to do a Ph.D. in Computing at the Open University, on evolutionary techniques for color scheme design.

Martin Stacey, after studying psychology at Oxford and Carnegie Mellon, did a Ph.D. on AI techniques for model-driven scientific discovery at the University of Aberdeen. As a research fellow at the Open University, he developed his current research interests in intelligent design support systems, the human computer interaction aspects of computer aided design, and the psychology of design. In 1996 he moved across the road to become Senior Lecturer in the Department of Computer and Information Sciences at De Montfort University, Milton Keynes.