Mismatched metaphor: user vs system model in computer-aided drafting

Ulrich Flemming, Department of Architecture and Engineering Design Research Center (EDRC), Carnegie Mellon University, Pittsburgh, PA 15213, USA
Suresh K. Bhavnani, Department of Architecture, Carnegie Mellon University, Pittsburgh, PA 15213, USA
Bonnie E. John, Department of Computer Science and Department of Psychology, HCI Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA

We report findings from an extensive study of the users of a Computer-Aided Drafting (CAD) system. Our observations suggest that the CAD system is used inefficiently, because users approach computer-aided drafting from a T-square metaphor reflecting their past experience with traditional drawing media. This prevents them from discovering and using effectively powerful system commands that have no equivalent in manual techniques. These findings suggest that we should rethink the ways in which CAD users are trained and manuals are written, and that we introduce CAD users to a more strategic use of CAD, particularly to a Detail/Aggregate/Manipulate (DAM) strategy that takes advantage of the compositional logic underlying a design. © 1997 Elsevier Science Ltd.

Keywords: Architectural design, computer-aided drafting, user behaviour, case study, modelling

It has been observed at numerous occasions that 'phase-one' computers, designed to automate tasks requiring massive mathematical calculations or routine activities, e.g. accounting, have had impressive successes, whereas 'phase-two' computers, designed to augment human problem-solving capabilities, typically have shown disappointing results [1]. This phenomenon is commonly referred to as the productivity puzzle. Computer-Aided Drafting (CAD) does not differ much from this general picture[2]. Our own observations, described briefly in Section 1, confirm these findings and actually expand them.

1 **Landauer, T K** *The trouble with computers: usefulness, usability, and productivity* MIT Press, Cambridge, MA (1995)
2 **Bhavnani, S K, Flemming, U, Forsythe, D E, Garrett, J H, Shaw, D S and Tsai, A** 'CAD usage in an architectural office: from observations to active assistance' *Automation in Construction* Vol 5 (1996) pp 243–255

ELSEVIER
0142-694X/97 $17.00 *Design Studies* **18** (1997) 349–368 PII: S0142-694X(97)00006-9

We must clarify at the outset what we mean by CAD in the context of this paper because the 'D' in CAD has variously been used to denote 'drawing,' 'drafting' or 'design.' A computer-aided drawing system, or drawing system for short, is any program that allows users to produce interactively configurations of graphical shapes composed of lines and curves that may, or may not be filled by two-dimensional patterns. These systems are distinct from paint systems, because the shapes, after being created, are not immediately broken up into bitmaps from where they cannot be reconstructed; rather, they remain accessible and therefore editable as such. A drafting system is a drawing system designed to support the production of —possibly complex — drawings that are part of the professional services provided by architects and other design professionals to clients; they typically contain all of the functionalities of a normal drawing program and augment them significantly with powerful features addressing the more demanding use made of them; e.g. MicroStation or extensions written on top of it, offer — in addition to the typical commands to create lines and curves — a mechanism to manage uniform borders of drawings through reference files; a database to store non-geometrical attributes of building elements; and based on this, the capability to generate schedules, etc. A design system supports design decision making more actively than drawing or drafting systems, which typically record a designer's decisions after they have been made, but are unable to make suggestions of their own; evaluate choices and thus give feedback; manage constraints etc. Some commercial drafting systems have started to incorporate features of design systems, but true computer-aided design systems, if they exist at all, do this only in the form of research prototypes [3].

CAD in the following means computer-aided drafting in the above sense. An objection which is sometimes voiced against the type of studies we are reporting in this paper is that drafting as we know it is rapidly becoming obsolete as electronic forms of communication become more wide-spread, actually universal. Our response is that the evidence for this claim is far from convincing. Physical drawings are the contractual basis for the services rendered by design professionals to clients all over the world and will remain so for some time; this assures a continued role for drawings in the foreseeable future. In building design at least, we are, furthermore, far from a situation where designs are generated with the help of computer-aided design systems that can provide input for special applications and generate traditional drawings automatically without the type of user involvement needed in present-day drafting systems. Drafting systems will, therefore, remain practically useful for the foreseeable future in building design. However, even when these systems become obsolete, the strategies we introduce below will remain useful for the generation of computational

3 **Flemming, U and Woodbury, R** 'Software environment to support early phases in building design (SEED): overview' *J. Architectural Engng* Vol 1, No 4 (1995) pp 147–152

design, because these strategies represent responses to genuine design features that are independent of the way in which they are depicted.

In addition, we consider the issues raised of theoretical interest. To date, cognitive analysis that aims at increasing productivity and decreasing learning costs has dealt mainly with 'linguistic' tasks, where the information is presented textually or orally, like text-editing, programming, or call-handling by telephone operators. A domain like CAD, which is dominated by visual processing of spatial or geometrical attributes and relationships, has received much less attention; i.e. the applicability of cognitive analysis techniques to these types of tasks is still an open issue and our study makes a contribution to this emerging field.

We have specific evidence that users approach a CAD system and are indeed encouraged to do this by the training they receive, as a mere substitute for manual drafting tasks. Our observations suggest (Section 2) that they employ a 'T-square metaphor' that enables them to find methods for executing certain drafting tasks very effectively, but at the same time prevents them from executing other tasks more efficiently and from taking advantage of the full power of the system. The T-square metaphor can, in fact, lead to outright errors in the drawings produced.

We argue that the efficient use of CAD depends on the use of strategies that involve not only planning ahead, but a different type of planning, which differs significantly from planning as used in manual drafting. These strategies are not discovered through metaphors like the T-square metaphor: drawing with a CAD system is fundamentally different from drawing by hand. These differences are not discovered spontaneously, but have to be taught. This appears as a critical disadvantage for CAD, because it violates the premise that the interaction between a user and a computer should be as close to the user's prior experience with the task as possible (hence the 'desktop metaphor' employed in office automation). However, we show in Section 4 that the kind of preplanning required in effective CAD can take advantage of genuinely architectural concepts: the composition of a drawing can reflect the formal composition of the design being drawn in terms of an underlying vocabulary of parts and systematic relations between parts.

We conclude with the expectation that an approach to CAD training and use as advocated here will establish the production of CAD drawings as a genuine craft based on a more than superficial familiarity with the tools used, tools that are not just substitutes for traditional tools, and must be appreciated and understood on their own terms.

1 Inefficient CAD use observed

One of us (Bhavnani) observed professional architects using a CAD system in their daily work environment and recorded his observations. The observations covered 14 CAD users (mostly architects), lasted two weeks, and employed a variety of observational and recording techniques ranging from informal, open-ended ethnographic methods to videotapes and keystroke records, which we are currently subjecting to a detailed, formal analysis. As a first result of this study, we generated a list of preliminary recommendations to vendors, users, managers and academics that spell out, for each group, what contributions it could make to overcome some of the observed inefficiencies in CAD use [4]. The present paper presents observations and recommendations that emerge from the more formal analysis, a keystroke-level investigation of how the users in the study approach and execute drafting tasks.

We start by describing two examples that demonstrate inefficient strategies employed by the observed users while performing CAD tasks. The CAD system used was Intergraph's MicroStation, one of the most sophisticated commercially available drafting systems today.

1.1 Example 1

One of the users in our study (referred to as B1 below) modified a design file from a marked-up hard copy. His task was to draw (plan views of) fire protection enclosures around columns in a floor plan. The fire protections were to be drawn as L-shaped polygons patterned with dots and triangles symbolizing concrete.

The enclosures for the first column consisted of two identically patterned, L-shaped polygons that were mirror images of each other [Figure 1(a)]. To construct these objects, B1 first drew the top shape (1), and then mirror-copied the shape to create the bottom shape (2) as shown in Figure 1(a); he then pochéd (patterned) each of the two shapes (3). It would have been more efficient to detail the first shape completely by drawing and patterning it, and then mirror-copy the complete, patterned shape to create the second shape.

B1 also used an inefficient method to complete the subtask of drawing the L-shape itself [Figure 1(b)]. He drew the top horizontal line (1) and then the left-most vertical line (2). Next he used the COPY PARALLEL command to make copies of the two lines drawn (3) and used the MODIFY TO INTERSECTION command to cleanup the intersection of the two lines (4). Finally, he drew one endcap of the shape (5) and used the COPY PARALLEL command to make a copy of the inner elbow of the shape to create the lower endcap of the shape (6).

4 **Bhavnani, S K, Flemming, U, Forsythe, D E, Garrett, J H, Shaw, D S and Tsai, A** 'CAD Usage in an architectural office: from observations to active assistance' *Automation in Construction* Vol 5 (1996) pp 243–255

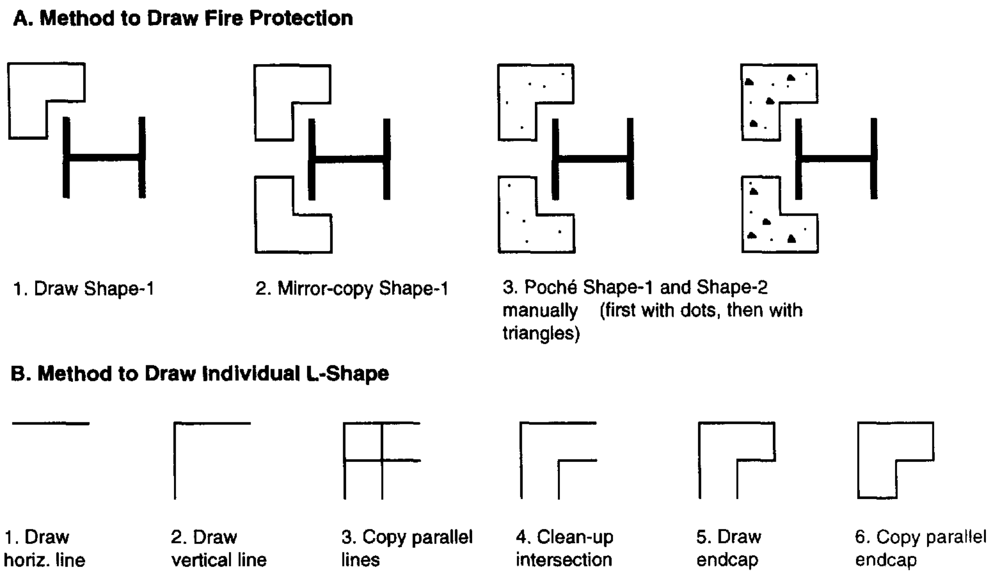

Figure 1 Methods used by B1 to draw the fire protection (c) 1996 ACM, Inc. Reprinted by permission

This way of drafting had two repercussions. Firstly, as the automatic PATTERN command in MicroStation is designed to pattern only closed shapes, B1 had to pattern each shape by copying individual dots and triangles from a nearby concrete shape. Secondly, when he decided to mirror-copy the shape, he had to group the individual line segments temporarily together using the FENCE command before he could mirror-copy the shape. In the version of MicroStation used in our study, this procedure required the user to select individual pixels accurately several times. This difficult perceptual/motor task turned out to be quite error-prone. If he had used the command PLACE ORTHOGONAL SHAPE instead to create the L-shape as a closed polygon, he could have used the automatic PATTERN command in MicroStation, as well as the regular MIRROR COPY command. This would have avoided the error-prone attempts at precise line drawing, manually patterning and creating an appropriate fence to achieve the grouping.

1.2 Example 2

Once the L-shapes were drawn, B1 moved the endcaps that coincided with a wall ending at the column a short distance away from the wall (Figure 2). There are two possible explanations for this. It is a common technique in manual drafting to draw a gap between two objects that share a boundary to indicate that the two objects are actually different physical objects and not parts of the same object. Alternatively, the user may have wanted to

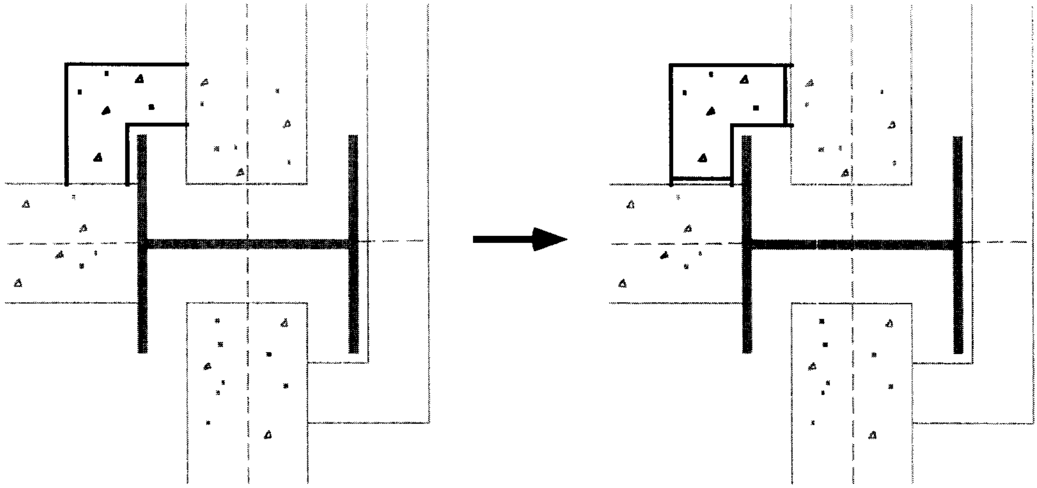

Figure 2 Modifications of end caps

reserve a gap in which he could draw a wall cover like sheet rock at a later time. The result is the same in either case: the actual dimensions of the L-shapes are wrong, because moving the caps reduces the sides of the shapes by a small amount. This poses a problem if, e.g. the CAD drawing is used as the basis for deriving a three-dimensional geometric model or for automatic dimensioning, both of which would be wrong in this case. As an alternative, the user could have moved the L-shapes a small distance away from the walls; this would have preserved their dimensions, but if the gaps have no physical meaning as place-holders, the positions of the L-shapes would be now wrong.

Note that this action is harmless in manual drafting, because the convention of separating parts that are physically connected, but separated graphically by a small gap, is well-understood and accurate measures are normally not taken directly off a drawing, e.g. with a ruler, but read from measuring lines giving quantities explicitly.

2 *The T-square metaphor at work*

Our data, which are still being analysed at the time of this writing, provide evidence that the examples presented in the preceding section are by no means unique. We suggest that the inefficiencies and errors observed can be consistently explained by what we would like to call a 'T-square metaphor' motivating B1 and other users in the way in which they structure a drafting task and select commands to execute it. The term metaphor goes ultimately back to Aristotle, who defines it as a transference of meaning between words, i.e. as a rhetorical device [5]. One of the most famous metaphors in this sense, in the history of modern architecture is LeCorbusier's dictum that 'a house is a machine for living.'

In the context of human–computer interaction, especially interface design, the term metaphor has come to denote the transference of a part of a user's prior, often task-related experience (called the source domain) to features of the interface (the target domain) so that they are better understood, learned and remembered [6]. At the same time, it may become more than a rhetorical device; e.g. the interface widgets called 'radio buttons' are not known by another name; their behaviour is modelled after the buttons on a car radio, and they are given the same name by the interface developers; in this case, the transference happens not so much between words, but between objects (based on a behavioural analogy). The metaphor thus becomes part of the conceptualization of an interface, not just an interpretation imposed on it after its creation.

The best known of the interface metaphors is the 'desktop metaphor' used by the inventors of the Xerox Star interface, a then very novel and innovative office information system. Its creators

> ...decided to create electronic counterparts to the physical objects in an office; paper, folders, file cabinets, mail boxes, and so on - an electronic metaphor for the office. We hoped this would make the electronic 'world' seem more familiar, less alien, and require less training [7].

The same metaphor was used in the Apple Lisa and later the Macintosh interface [8], whose success made it spread to other platforms. Note that the metaphor applies not only to the objects manipulated by the user, but the manipulation commands themselves; Figure 3 illustrates this by showing the way in which the cut-and-paste operation is introduced in the MicrosoftWord manual. The accompanying text reads: 'Moving text is a two-part process: you cut and then paste, just as if you were using scissors and glue' [9].

Carroll et al. [10] introduce metaphors as 'the most widely used design technique for controlling interface complexity:'

> Metaphoric comparisons and interface presentations do more than render static denotative correspondences. They have motivational and affective consequences for users. Perhaps more importantly, they interact with and frame users' problem-solving efforts in learning about the target domain.... The ultimate problem that a user must solve is to develop an understanding of the target domain itself — a mental model [10].

5 Ricoeur, P *The rule of metaphor* **R Czerny** (translation) Toronto University Press, Toronto, Canada (1977)
6 Target and source domain are also called tenor and vehicle or primary and secondary subject, respectively, in the metaphor literature; see **Way, E C** *Knowledge representation and metaphor* Kluwer Academic, Boston, MA (1991) p 28
7 Smith, D C, Irby, C, Kimball, R and Verplank, B *Designing the Star User Interface Byte* Vol 4 (1982) p 252.
8 Seybold, J and Seybold, A Apple's Macintosh and Lisa 2: The alternative to IBM, The Seybold Report on Professional Computing, Vol 2 No 6 (1984)
9 *Microsoft Word. Getting Started* Microsoft, Document OB-23344-1091 (1991) p 42
10 Carroll, J M, Mack, R L and Kellogg, W A 'Interface Metaphors and User Interface Design' in **M Helander** (ed) *Handbook of human–computer interaction* Elsevier/North Holland, New York (1988) p 81; **van Norman, M A** 'Digital model shop: the role of metaphor in a CAD User Interface' *Design Computing* Vol 1, No 2, (1986) pp 95–122

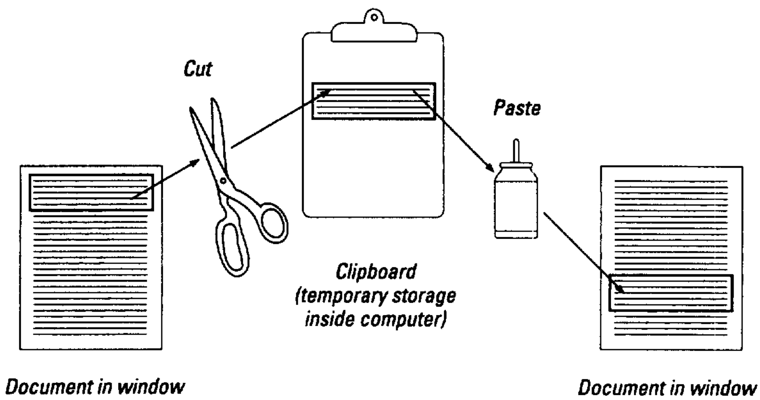

Figure 3 Cut-and-paste illustration in the Microsoft Word manual (c) 1991 Microsoft Corporation. Portions reprinted with permission from Microsoft Corporation

It has been observed in this connection that users develop metaphors spontaneously when they try to understand or explain how a program works, and that often several metaphors are at work [11]. Indeed, if one accepts that 'learning by analogy is the only way that humans actually learn... it is likely that people will generate metaphoric comparisons on their own.' Thus, 'metaphors, including mismatches and composites, may be inevitable in human–computer interaction' [12].

The T-square metaphor for CAD systems as suggested by us is analogous to the desktop metaphor for office information in that it directly translates parts of a user's prior experience with the task at hand to the computer tool. In the present case, it suggests an approach to drafting that mirrors closely the sequence of steps a draftsperson would employ when using manual techniques. Consider example 1(b): The task is to determine the location of a point at a precise distance from more than one reference point or line. In manual drafting, the draftsperson draws parallels at the appropriate distances from the reference lines, which is easily done with T-square or parallel rule, triangle, and ruler and finds the desired point as the intersection of the lines. The sequence of steps B1 uses to execute this task closely resembles this strategy: he uses the COPY PARALLEL command twice to create what are essentially construction lines and follows this with two applications of the MODIFY TO INTERSECTION command, which find the desired intersection and remove the unwanted portions of the construction lines.

Example 2 illustrates how conventions that are common in manual drafting are transferred to CAD, where they may cause problems. This approach cannot be explained through the characteristics of the CAD system at all and we have to look for an external explanation of the kind provided by

11 **Madsen, K H**, 'A Guide to Metaphorical Design' *Communications of the ACM* Vol 37 No 12 (1994) pp 57–62.

12 **Carroll, J M, Mack, R L and Kellogg, W A** 'Interface Metaphors and User Interface Design' in **M Helander** (ed) *Handbook of human–computer interaction* Elsevier/North Holland, New York (1988) p 70

the T-square metaphor. This explains, at the same time, why the user did not find a more appropriate way to show that two objects that are joined are nevertheless physically distinct in the CAD drawing without falsifying their shape or location: they can be pochéd with different patterns indicating their different materials, which is extremely tedious in manual drafting, but almost instantaneous in CAD, provided the objects have been defined as closed shapes.

It is true that unlike the desktop metaphor mentioned above, the T-square metaphor has not been used consciously in the design of the CAD system whose users we observed. However, the metaphor is strongly suggested by training manuals that contain statements like 'CAD is an expansion of the way you draw' [13] and go on to describe the use of commands like DRAW LINE. The metaphor is further reinforced by CAD concepts such as 'drawings' and 'layers' that have direct analogs in traditional drawing sheets and overlay drafting (the use of overlapping tracing sheets).

The overt similarities between drafting by hand and by computer may give rise to the T-square metaphor even in the absence of reinforcements as we find them in manuals and training material. After we had formed the T-square hypothesis from our analysis of B1's videotaped interaction with the CAD system, we asked B1 in a non-leading way how he found the specific method to draw L-shapes that he used consistently over the entire drafting session. He volunteered, on his own, an explanation in terms of his prior drafting experience. We do not want to use this as direct corroborative evidence. However, we want to record at least that the T-square metaphor does not contradict how the CAD users themselves reflect upon their work.

Our strongest argument for the T-square metaphor at work in the observed behaviours is that it explains consistently the ways in which the users of the CAD system structure their drafting tasks, especially where this leads to manifest inefficiencies. To be fair, we have to admit that the T-square metaphor works at times in the expected sense. Example 1(b) demonstrates how it allowed the user to solve a problem that is easily handled in manual drafting, but somewhat more difficult for CAD systems: to determine the location of a point at a precise distance from more than one reference point or line. As the example shows, B1 found an analogous way using the CAD system, which included the correct use of the non-trivial MODIFY TO INTERSECTION command.

In example 1(a), however, postponing the copying of a group of elements until all the details have been completed is important if one wants to take full advantage of the MIRROR COPY command; B1 missed this opport-

13 **Obermeyer, T L** *AutoCAD architectural lab manual* McGraw-Hill, New York (1987) p v

unity and he missed it consistently in further drafting tasks. Example 1(b) demonstrates a less than optimal approach at the lower level of drawing an individual closed shape: before patterning the interior, the user should have transformed the individual lines into a closed shape, using the CREATE COMPLEX SHAPE command. Alternatively, he could have drawn the shape as a closed polygon from the start using the PLACE ORTHOGONAL SHAPE command; this, however, would pose a problem the user avoided: having to create accurate corner points repeatedly in the correct sequence. None of these operations, of course, has meaning in manual drafting: there is only one way to draw a shape — with individual lines, which are combined into shapes only in the user's perception. The error demonstrated in Figure 2 can be explained only by reference to traditional techniques: nothing in the CAD system would compel a user to do this.

The resulting inefficiencies have been analysed in greater detail elsewhere[14]. The important point for the present discussion is not so much that they occurred at all, but that the T-square metaphor proved unable to make users aware of the shortcomings of the methods used, let alone lead them to an 'exploratory' mode of learning new and better ways of using the system, and thus build a more accurate mental model of the system, as is expected from the role of metaphors in human–computer interaction. Although the observed users had many years of experience using the CAD system, they consistently used inefficient methods to complete their tasks and never progressed beyond a certain level of proficiency, where they could 'get the job done,' albeit inefficiently and — at times — with errors.

Possible reasons for this become apparent if we take a closer look at the drawing strategies appropriate for manual vs CAD drafting [15]. Figure 4 illustrates this for the task to draw three identical arched windows in a row. The manual method [Figure 4(a)] starts with drawing the three arcs that cap the windows, followed by the vertical lines and then the horizontal (bottom) lines. This sequence reflects various established principles for accurate and quick drafting, which take into account that it is easier to match a compass-drawn arc with a line than vice versa; minimize the movement of drafting utensils over the paper; prevent smudging by not moving utensils over figures already drawn, etc. [16]. In CAD however, as shown in Figure 4(b), it is better to first draw all the lines that make up the repetitive shape (1 and 2), group them (3) and then make two copies (4), or translate the shape twice.

Manual drafting is governed by the mechanics and logistics of moving drafting utensils over paper into various configurations that aid the draftsperson in drawing accurate graphical elements, and this is completely inde-

14 Bhavnani, S K and John, B 'Exploring the unrealized potential of computer-aided drafting' *Proceedings CHI-96*, Vancouver, Canada, 14–18 April 1996. ACM, New York (1996) pp 332–339. **Bhavnani, S K and John, B** 'From sufficient to efficient usage: an analysis of strategic knowledge' *Proceedings CHI-97* Atlanta, March 1997, ACM, New York pp 91–98.
15 Bhavnani, S K and John, B 'Exploring the unrealized potential of computer-aided drafting' *Proceedings CHI-96*, Vancouver, Canada, 14–18 April 1996. ACM, New York (1996)
16 Ching, F *Architectural graphics*, Van Nostrand Reinhold, New York (1975)

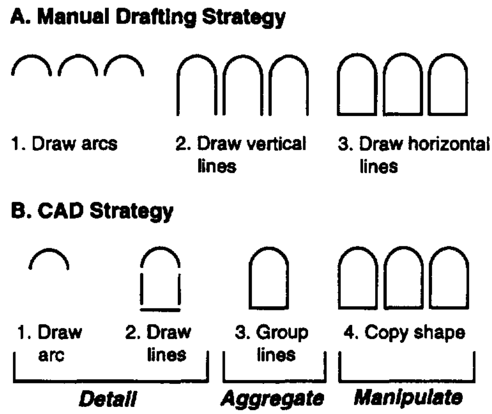

Figure 4 Efficient manual and CAD strategies for the same drafting task. [15] *(c) 1996 ACM. Reproduced by permission*

pendent of the way in which the graphical elements combine to represent physical objects; i.e. the draftsperson decomposes the task at hand in terms of the graphical primitives to be drawn and the way in which they are placed in relation to the general drafting direction (from upper left to lower right for right-handers and from upper right to lower left for left-handers). The more powerful commands offered by CAD systems, on the other hand, come into play only if the drafting task is decomposed in terms of the objects of which the design is composed and their relative spatial relations, and they can be used more efficiently the sooner these objects have been created.

We have suggested elsewhere that the strategy appropriate for CAD drafting is a Detail-Aggregate-Manipulate (DAM) strategy as illustrated in Figure 4(b) [15]. In the same paper, we have given reasons why we believe that this strategy cannot be discovered on its own, e.g. through exploratory learning guided by the T-square metaphor, but has to be taught explicitly and incorporated into documents and training manuals, which may have to be completely restructured, because DAM has no equivalent in traditional practices. In the remainder of this paper, we deal with a major objection that is frequently raised against this suggestion: that we fall into the old trap of blaming the user when we should blame (i.e. redesign) the system.

We are actually doing neither of the above. However, we must stress that in light of the very real differences between manual and CAD drafting, one cannot expect that users who have been educated in traditional techniques can master the new tool without changing their basic approach. We are not the first to recognize the differences between manual and CAD drafting, and the importance of reorganizing drafting tasks when using a CAD system. Crosley, e.g. emphasizes the importance of 'thinking CAD:'

It's possible to use computer-aided drawing without really taking advantage of its capabilities. Even some experienced CAD users have simply transferred all their manual-drawing habits over to the computer [17].

Later he adds that '...the advantages of CAD are not free; they come at the expense of having to actually design the drawing' [18]. Very much in the spirit of the lessons we learned from example 1 and similar incidences, he advises users to 'never draw anything twice!' when describing the COPY command [19]. He and the few other authors of manuals that address this issue at all leave the burden of finding the right approach in a given situation entirely to the user.

Our evidence suggests that users do not discover the appropriate drafting strategies by themselves and do not get the kind of assistance that would remedy this. The system model presented by manuals and the instruction received is that of a vast collection of commands. There is not much explicit instruction in terms of how to put these commands efficiently together in a drafting task. The T-square metaphor is clearly insufficient to allow users to fill this gap on their own.

We conclude from this that users have to be explicitly instructed in how to structure a CAD drafting task appropriately and in the strategic planning this requires. Thus, we put the burden primarily on the instructors, training manuals, text books, documentation, etc. We will see later that certain features of the CAD system interface may also have to undergo changes. What we have to expect from users is a readiness to learn new ways of doing familiar tasks; we will explain in the succeeding sections why this may not be an unrealistic expectation.

The DAM strategy is the best strategy that we have found so far that is both able to lead to a more efficient drafting process and at the same time explicit enough to be taught. We show in the next section that this strategy can be linked to genuine interests and concerns shared by architectural users.

3 From transformational geometry to transformational architecture

Our own observations, as well as those of authors like Crosley, show that one of the most important strategies for efficient CAD use is to avoid drawing the same object twice. This involves detailing and aggregating the repetitive part sufficiently at the outset or, if parts are repeated across drawings, recalling predrawn 'cells' or 'blocks' from some library or file. The rest can be done by a combination of copy and move commands.

17 **Crosley, L M** *The architect's guide to computer-aided design* John Wiley, New York (1988) p 6
18 **Crosley, L M** *The architect's guide to computer-aided design* John Wiley, New York (1988) p 11
19 **Crosley, L M** *The architect's guide to computer-aided design* John Wiley, New York (1988) p 41

The move commands offered by CAD systems, in turn, include the basic rigid motions or isometries translation, reflection and rotation. One of the most important theorems in geometry, the Isometry Classification Theorem, states that these three isometries, together with a fourth one called glide-reflection, are the only transformations that preserve distances and, consequently, all properties dependent directly or indirectly on distance; these include the property of being a line, angles, parallelism, etc. That is, the isometries are sufficient to move any figure to any conceivable position in the plane without destroying its shape [20]. We can, in fact, dispense with the glide reflection, because it can be implemented by a translation followed by a reflection. Note in general that any sequential application of isometric transformations is in itself an isometry, i.e. leaves the shape unchanged. This is the theoretical foundation for the inclusion of translations, reflection (or mirror) and rotation commands in any modern CAD system.

The practical motivation for including the isometries in a CAD system is that they allow for the efficient handling of standardized or repetitive elements with the same shapes. When it comes to building design, the frequency of repetitive shapes can be traced to the nature of architecture itself. It has been observed that architecture is characterized by an 'intentional regularity' [21]. It has also been shown that a very basic strategy employed by architects through the centuries to achieve this regularity is to base the physical components of a building on a 'vocabulary' of a few, repetitive parts arranged in patterns that are easily recognized; formally, they can often be understood in terms of the isometric transformations [22].

We illustrate this with a few examples intended mainly for readers with a non-architectural background. Patterns that can be understood in terms of translations and reflections are ubiquitous in architecture, particularly in styles and movements in the classical tradition, from ancient Greece and Rome, through the Renaissance and the Baroque to Classicism, and the numerous revivals of that style up to the present day. Figure 5 gives a modern example; it shows the front elevation and longitudinal section of S. Ungers' T-house. The elevation is characterized by 'bilateral' symmetry or reflection about a central axis, and elevation and section show the repetition of a window and wall panel module, respectively, arranged in a linear sequence that can be understood in terms of translations.

Figure 6 shows the elevation of a 'Queen Anne house' that is irregular overall, but contains various parts that are bilaterally symmetric by themselves (indicated in the figure by the respective axes of symmetry; such symmetries are usually called local and found in many architectural traditions).

20 Gans, D *Transformations and geometries* Appleton-Century-Crofts, New York (1969)
21 Scruton, R *The Aesthetics of architecture* Princeton University Press, Princeton, NJ (1979) p 165
22 March, L and Steadman, P *The geometry of environment* MIT Press, Cambridge, MA (1971)

Figure 5 T-House, Wilton, NY (1992), S. Ungers, architect (c) Ungers. Reproduced by permission of S. Ungers.

Figure 6 Elevation of a Queen Anne house, Pittsburgh, PA (1887) (c) Ulrich Flemming.

Rotations at work in a plan are illustrated in Figure 7, which shows the plan of a small apartment building by F.L. Wright that consists of four parts derived from each other by 90° rotations about a centre point where all four parts meet.

The patterns achieved by applying transformations can become more intricate if they are applied recursively. We illustrate this using Palladio's Villa Rotunda as an example [23] (Figure 8) in Figure 9. This building has two perpendicular axes of symmetry that centre identical porticoes at each of

23 **Palladio, A** *The four books of architecture*, reprint of the English edition of 1738 by **I Ware** Dover, New York (1965) book II, plate XIII

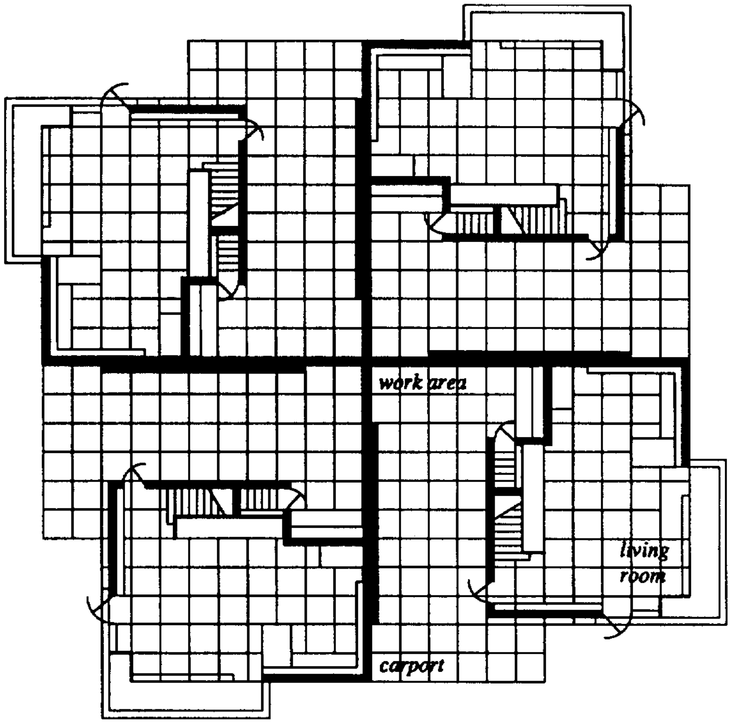

Figure 7 Suntop Homes, Ardmore, PA (1939-42), first floor plan. F.L. Wright, arch. (c) Ulrich Flemming.

the four sides of the building. The porticoes themselves are bilaterally symmetric and can be drawn quickly by drawing one half (1) and then reflecting it about its axis (2). Two successive reflections about the diagonals generate the entire exterior (3a–4a). Alternatively, two successive rotations about the centre point achieve the same result (3b–4b). Other possibilities exist to achieve this, e.g. with a combination of rotations and reflections. The interior can be drawn with similar strategies that capitalize on the strong symmetries inherent in the design.

Translations of repetitive shapes may also be applied recursively; e.g. a facade may consist of a row of identical windows, each of which may in turn consist of a row of identical panes, possibly with some mirrored patterns thrown in. Much of the formal intricacy of architecture is based on these nested or recursive patterns and arrangements. Note that a system supporting a way of drafting that takes advantage of these regularities needs general transformations like rotations about arbitrary points, reflections about and translations along arbitrary lines. A system like MicroStation provides all of these, albeit often with a very unintuitive interface.

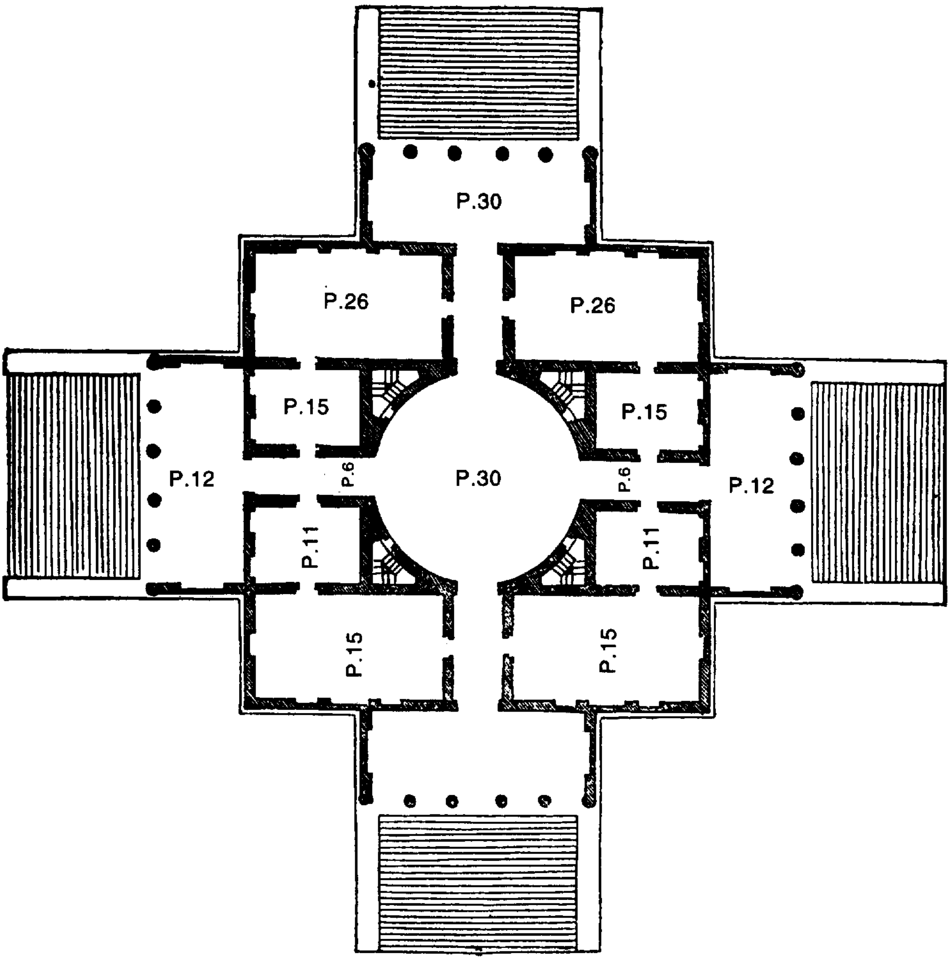

Figure 8 Villa Rotunda, plan[23]. Reprinted with permission by Dover

Mitchell at al. have observed the same translational and recursive patterns in architectural compositions and use them as motivation to teach basic constructs of graphics programming [24]. They further liken the efficient use of CAD systems to programming when they state 'You must first analyse the drawing in terms of the vocabulary to be used' and then 'think carefully about the structure of the drawing in terms of repetition, conditionals, the hierarchy of parts and the use of transformations' [25]. We suggest that a structured approach to CAD education is needed parallel to Mitchell et al.'s approach to graphics programming.

24 **Mitchell, W J, Liggett, R S and Kevan, T** *The art of computer graphics programming* Van Nostrand Reinhold, New York (1987)

25 **Mitchell, W J, Liggett, R S and Kevan, T** *The art of computer graphics programming* Van Nostrand Reinhold, New York (1987) p 515

The DAM strategy, recursively applied, can provide this structured approach, where the compositional logic inherent in the design to be drawn

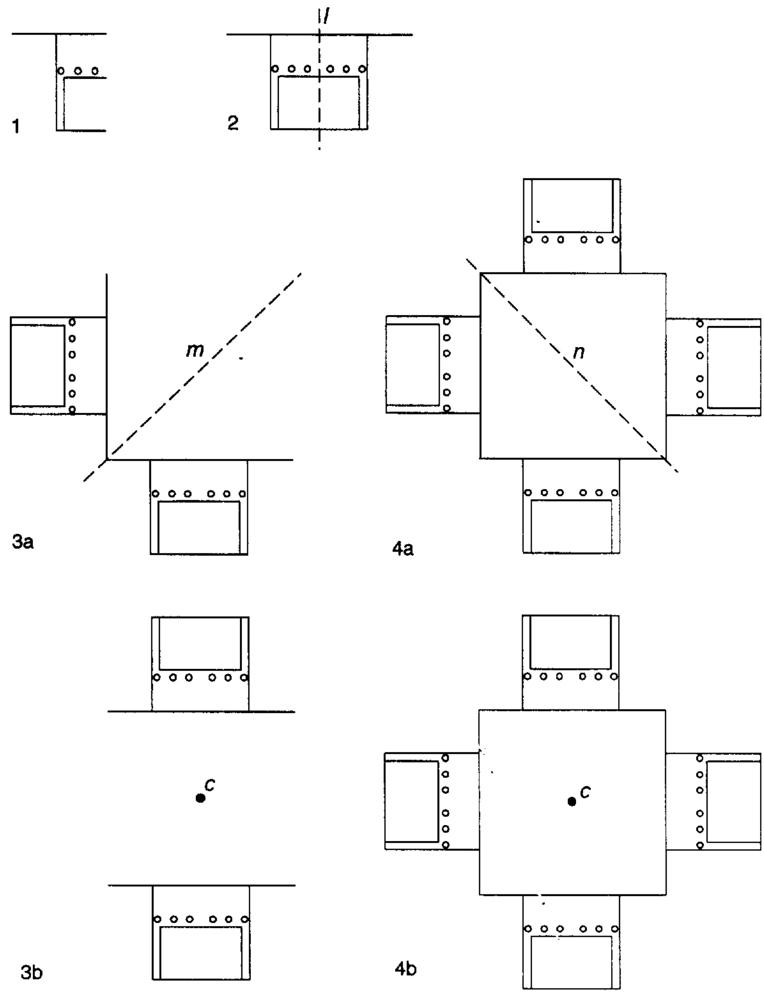

Figure 9 Drawing the exterior of the Villa Rotunda by recursive application of isometric transformations

determines the levels and objects to be aggregated at each level. Clearly, such a drafting strategy needs to be carefully planned ahead. We suggest that it need not be alien to users, at least not if they have a sufficient architectural background. On the contrary, it may be based on their genuine interest in creating intriguing architectural compositions of parts.

The T-square metaphor may remain useful at the detail stage when individual graphical elements that make up a shape have to be drawn. However, it should be embedded into a DAM strategy that is able to take fuller advantage of the power offered by an advanced CAD system while remaining, at the same time, true to the background and interest of its users.

At this time, of course, we have no evidence that this will have the desired effect, although we are interested in finding this out. As a first step, some are planning to design and teach an introductory CAD course in the upcoming academic year that approaches the task not from the T-square metaphor nor from the level of individual commands, but from the strategic perspective implied by the DAM strategy as envisaged here; i.e. we try to find an answer to the question posed at the beginning of this section: what does it mean to 'think CAD'? The expectation is that this will allow users to develop a more complete and accurate mental model of the CAD system.

Along the way, we may discover desired features of the CAD system that are currently missing; e.g. reflections about an axis are most natural when users can draw or select the axis explicitly; the MIRROR COPY command in MicroStation does not provide for this; it uses an implied axis that is very difficult to visualize and manipulate explicitly by a user. Similar problems exist for rotations about arbitrary points.

Finally, we would like to suggest that the approach put forward here can be extended to the entire set of Euclidean transformations, which include, in addition to the isometric transformations, the similarity transformation or similarities for short. Similarities do not preserve distances, but proportions between distances and properties dependent on these, e.g. lines, angles, parallelism, etc. That is, they preserve most of the properties preserved by isometries. Through the centuries, architects have used similarities between shapes, or 'proportional systems', to establish part/part or part/whole relations, examples of which are shown in Figure 10. Similarities between rectangles are easily established by showing that they have parallel or perpendicular diagonals. When used in this function, the diagonals are known as regulating lines in architectural design (see Figure 10; the regulating lines show both part/whole and part/part similarities in the elevation of a residential structure).

Similarities are offered in CAD systems through scaling commands, which can be used to shape similar objects. Support for the manipulation and creation of similar shapes could be provided by a CAD system that offers not only scaling, but allows users also to define regulating lines explicitly, similarly to the way in which they should be able to define symmetry axes or points of rotation explicitly.

A legitimate question that may be asked is this: what happens to a piece of architecture possessing none of the regularities that can be exploited by an efficient drafting strategy? An answer that is consistent with our view that CAD is not a substitute for manual drafting is this: there may indeed

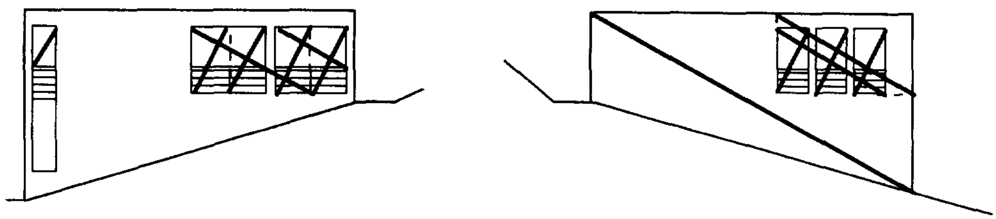

Figure 10 Regulating lines in an elevation establishing similarities between whole and parts [Kress House, Massagno, Switzerland (1986) Campi/Pessina arch.]

be designs that can be as easily drawn by hand as by a CAD system, if not easier. In this case, one may not use a CAD system, unless of course one is interested in more than a drawing, e.g. in a computable representation that can support additional applications like the generation of schedules; i.e. CAD does not always improve the productivity of drafting (and unless more efficient strategies are employed, this may indeed be the norm), but it also offers functionalities that go beyond drafting, because it generates a computable design representation that can be used for other purposes. Consequently, it has to be understood and approached on its own terms, which remain mostly hidden if it is approached, implicitly or explicitly, through the T-square metaphor.

4 *Drafting as craft*

Productivity in the broad sense is the motivation behind much of the discussion of human–computer interaction and interface design, including the role of metaphors. Dominant issues are ease of learning a system; retention of what has been learned; possibilities for exploratory learning, and reduced need for training and manuals; frequency of slips or errors, etc. Given that a modern CAD system is an expensive investment, these are indeed legitimate concerns. However, when we present parts of our analysis that go down to fine details, e.g. the execution of individual commands, we are sometimes accused of going too far or of resurrecting Taylorism in a modern form.

To this, we have to respond that first of all, records of command usage down to the keystroke level provide the empirical evidence without which a hypothesis like the existence of a T-square metaphor would remain pure speculation. It was only after we had a close look at what happened during a drafting task that we formulated the hypothesis, which we then tried to confirm by both looking at more data and by seeking corroborative evidence.

Furthermore, analysis at the level of command use and keystrokes allows us to model the behaviour of users quite precisely. Such models have been shown to have excellent predictive power for evaluating task-completion and learning time resulting from differences in procedures and interfaces[26]. These models allow us to posit and explore rigorously the implications of different strategies like DAM, prior to extensive training of actual users and to evaluate design suggestions for CAD interfaces without building them.

Analysis at this level, especially when rigorously defined in a model, is a representation of what people need to know if they are to use a system productively. This knowledge has been shown to be a good basis for documentation, training, help systems and even automated tutoring systems [27]. Thus, our analyses provide an important step toward realizing our intention to teach the knowledge and skills necessary for effective CAD use.

Aside from a practical interest in productivity, learning and training, there is a complementary aspect to our work that is as important to us. We want to establish CAD drafting as a craft requiring skills and knowledge that the users can be proud of, in the way that traditional craftspeople are proud of their skills and knowledge, which includes intimate knowledge of and respect for the tools of the craft. In order for this to happen for CAD drafting, it has to move away from the T-square metaphor.

Acknowledgments

The empirical studies on which the ideas presented here are based were supported in part by the US Army Corps of Engineers Construction Engineering Research Laboratory (USA-CERL) under contract #DACA88-94-K-0006 and by the National Science Foundation (NSF), award #IRI-9457628. The views and conclusions contained in this article are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of USA-CERL, NSF or the US Government. The Engineering Design Research Center at Carnegie Mellon University is an NSF-supported Engineering Research Center.

26 **Card, S K, Moran, T P and Newell, A** 'Computer text-editing: an information-processing analysis of a routine cognitive skill' *Cognitive Psychology* Vol 12 (1980) pp 32–74; **Gong, R and Kieras, D A** 'Validation of the GOMS model methodology in the development of a specialized, commercial software application' *Proceedings CHI-94*, Boston, USA, 24–28 April. ACM, New York (1994) pp 351–357; **Gray, W D, John, B E and Atwood, M E**, 'Project Ernestine: validating a GOMS analysis for predicting and explaining real-world task performance' *Human–Computer Interaction* Vol 8 (1993) pp 237–309

27 **Anderson, J R, Corbett, A T, Koedinger, K R and Pelletier, R** 'Cognitive tutors: lessons learned' *J Learning Sci* Vol 4 (1995) pp 167–207; **Elkerton, J and Palmiter, S L** 'Designing help using a GOMS model: an information retrieval evaluation' *Human Factors* Vol 33 No 2 (1991) pp 185–204; **Gong, R and Kieras, D A** 'Validation of the GOMS model methodology in the development of a specialized, commercial software application' *Proceedings CHI-94*, Boston, USA, 24–28 April. ACM, New York (1994) pp 351–357