April 1989

Demons and Daemons: Personal Reflections on CAID

Del Coates
San Jose State University, dcoates@delcoates.com

Recommended Citation
Del Coates. "Demons and Daemons: Personal Reflections on CAID" *Innovation* (1989): 4-9.

Demons and Daemons

Personal Reflections on CAID

Del Coates, IDSA

Guest editor of this special Spring/Summer issue of **innovation,** ***Del Coates, IDSA, teaches industrial design at San Jose State University. He is also the academic director of the university's CADRE Institute, which is devoted to the application of computers to art and design. Del is a widely published expert on the role of computers in design and has spoken at numerous design and computer graphics conferences. Most recently, he organized the session on CAID at WORLDESIGN88/NEW YORK.***

Looking Back

How time flies! Nearly 25 years have gone by since computer graphics possessed me. It was that long ago that I saw Ivan Sutherland (then a graduate student at MIT and since called "the grandfather of computer graphics") demonstrate his *Sketchpad* software to a group of engineers, scientists and designers at Ford Motor Company. Sutherland projected simple, stick-figure polyhedrons rotating in space, but I saw full-color, realistic images of automobiles rotating on turntables. He showed me the future of industrial design that day and I haven't been able to stop thinking about it since.

In the fall of 1965, Ford granted me a leave of absence to study computer graphics at the University of Michigan. Equipment was primitive by today's standards. The notion of **interactive** graphics was so new that we had no **CRT** terminals at first. We **encoded** our designs on **punch cards**. Fortunately, we had relatively advanced software for creating **bicubic splines** as part of a 3D **wireframe** modeler that was powerful enough to define complex automobile surfaces and display them in perspective with relatively little **input data**. I was able to define a car body with a patchwork quilt of only 12 patches (with 24 numbers per patch) that resulted in a card stack less than six inches thick. More traditional **polygonal modeling** methods would have required many more numbers, and the stack could have reached several feet.

Aesthetics and Computers: The First Seeds

By today's standards of instant feedback, it took an eternity to turnaround drawings—one day! And virtually every plot brought unpleasant aesthetic surprises. Some malformations sprang from simple typos, holes punched in the wrong place. But most occurred because I was designing "in the dark." Numbers locating the patch corners were no problem to find since I could take the three-dimensional coordinates from profiles and cross-sections I'd drawn on grid paper; however, there was no way of knowing at the time I punched the cards exactly what curvalinear path the edges connecting the corners would take, because I had no interactive display to preview them on. I had to choose the six numbers that shaped each curve with a roll of the dice.

Somehow, I had to increase the odds of producing attractive results as well as increase efficiency. As with many designers then (and now), fear of the "demon computer" nagged me. Could I actually create beautiful things with a computer? Or, would the computer's rational, clammy hands chill anything it touched? Realizing that *rationality* was its strong suit, I looked to the Greeks, with their notions of harmony and symmetry, for answers. I severely restricted the numbers I would use to a small harmonic set. Each number was either a whole-number multiple of another in the set or evenly divisible by some member.

This simple strategy was so effective that I was compelled to continue the search for more quantifiable aesthetic variables. It was exciting to think that computers could be taught good taste. Eventually, my investigation led me to formulate a theory of **objective concinnity** and **subjective concinnity** and algorithms that apply these. (For a more in-depth discussion of these ideas, see my article, "Measuring Product Semantics with a Computer," in the Fall 1988, issue of *innovation.*)

Virtually none of my colleagues shared my vision of the computer as helpmate, especially with respect to aesthetics. Either they thought it was too remote to think about or they were troubled, even frightened, by the prospect. And the oldest computer demons of all lurked just below the surface: lost jobs and dehumanization of those that remained.

By the 1980 IDSA National Conference in San Antonio, things had changed enough to draw an audience of 50 to a panel on CAD, an audience feeling fear, hostility and enthusiasm. Still, even 2D drafting systems were far too expensive for most of us. And, although at least one CAD vendor had introduced shaded images of 256 colors, the majority dismissed rendering, even color, as frivolous.

What a difference a year made! The main hall of the 1981 IDSA National Conference at Los Angeles was s.r.o. for my presentation of images from cheaper, yet more robust, CAD systems and photorealistic, animated movies. The optimists seemed to outnumber the pessimists. CAD systems were not only doing more and doing it faster, they were cheaper—rarely affordable, but still cheaper. While CAD systems typically were still limited to around 256 colors, very simple rendering and screen resolution of only 260,000 **pixels**, animation systems had **true color** palettes of over 16 million colors and 16 million pixels. These high-end systems indicated what would be possible. I then predicted that CAID would arrive within ten years, with the marriage of CAD and animation technologies. As those attending WORLDESIGN88 know, the marriage has been consummated and in less than seven years. CAIDis here.

Looking Around

Although CAID is still not our primary medium, it won't be long before most industrial designers are using some form of it. To demonstrate how far we've come, already

90-95 percent of Hewlett Packard's 100 designers use CAID, even though none used it just a few years ago. Moreover, this issue of *innovation* had to turn away high-quality submissions, even though we made it a double issue! To tighten our belt, we limited this issue to articles about how industrial designers actually use computers. (Future issues will carry more theoretical and philosophical articles.)

Hardware and software are much more robust and much less expensive today. A system for designing 3D objects, rendering them in relatively realistic fashion and plotting dimensioned drawings would have set you back perhaps as much as $200,000. Although no vendor makes a system with the best modeler, the best renderer and the best drafter bundled together, you could probably put together a suitable compromise for around $50,000 today. And $10,000-systems may be just around the corner.

But, in the cold, hard light of day, we still have a long, steep climb ahead. Although vendors like Intergraph and Alias Research now openly address industrial designers' needs, the industry as a whole doesn't. And, as much as I have advocated CAID, I find little I can justify doing with a computer. Computers still put a straightjacket on spontaneity—It's still easier to grab a piece of paper and a pencil to sketch a concept. And, although I cannot render as realistically as the best systems (which I can't afford, anyway), I can still knock out an effective rendering faster and better than the equipment I can afford. The *real* advantages of CAD don't show up until a design has to be modified.

The Conceptualization Problem: Perhaps the shortcoming most often cited by industrial designers is the CAID systems' ineptness for conceptualization. All the "sharp pencil" processes of analysis, refinement and optimization are "back end" design processes. The crucial front end, where concepts are sketched and played with, calls for fuzzy-pencil processes that foster spontaneity.

Fuzzy tools are especially important where aesthetic issues dominate. The moods projected by felt-pen empathic sketches on bleeding newsprint are more important than how accurately they represent the ultimate product. The designer depends to some degree, in fact, on the unpredictable outcome of sketching sessions, and looks for the fortuitous accident that might yield an appropriate new feeling.

The Rendering Problem: I want a system that can render so well it fools the eye into thinking it's seeing photographs of real things. Accurate drawing and rendering is essential to good design process, to solving and assessing aesthetic problems and to accurately communicating concepts to other professions. We've come a long way toward the goal of photorealistic CAD images, but we're still not there.

Most rendering software depicts only the diffuse reflection typical of unglossy objects. It gives an accurate enough impression of a product's overall form, but leaves out vital artifacts that affect aesthetic judgments: shadows, **specular** (mirror-like) **reflections** of the environment on glossy surfaces; the glow of diffuse inter-object reflection; and refractive distortions of transparent materials. Although variable and often subtle, these are real design elements that affect the product's image and its consumer appeal no less so than profiles, edges and colors. The specular reflections of the environment on an automobile, for instance, are so important that some designers actually design the horizon reflection first and let it shape the car.

Specular reflection and **refraction** require a ray-tracing algorithm and diffuse inter-object reflections require a **radiosity algorithm**. The algorithms are not especially complex, but they do require many calculations and iterations which soak up enormous amounts of a computer's time. So programmers have taken short cuts to simulate them. **Phong's algorithm**, a variant of the diffuse reflection algorithm, approximates highlights that really are specular reflections. **Reflectance-mapping** mirrors a hypothetical environment onto a product's surface to heighten the specular effect but does not reproduce inter- or intra-object reflections (a reflection of a car's windshield into its hood, or a reflection of one object into another). We can simulate transparent materials by merely letting the background show through objects, but these images lack the telltale distortions of refraction that give them their essential, aesthetically important, character.

Making Do: By ignoring CAID because it falls short of meeting *all* our needs *perfectly*, we risk missing the benefits it does offer. When adopting a new tool, you need to relax your expectations and, to some extent, let the tool have its way. You end up, not merely exchanging technologies, but improving the way you think and do your thing. CAID will move closer to the ideal (toward imitating traditional media better) but, as John Houlihan discovered at Timex, the design process itself may be changed in the end.

Designers in smaller firms that cannot afford sophisticated CAD and paint systems are finding that even modest PC- or Mac-based systems yield significant, worthwhile results when used to supplement traditional media. They continue

In the early days of CAD, notions of the "demon computer" nagged me. Was the computer capable of creating beauty? Or, would its rational, clammy hands chill anything it touched?

Even though we can today achieve photorealism with CAD, we are forced to compromise. Why? Because it is either too costly or too slow.

to use markers, pastels and colored pencils, but they base their sketches and renderings on computer-produced wireframe perspectives. Because industrial designers are by nature accomplished illustrators, they don't need plots that are complete to the last detail. They can get by with what amounts to a block diagram. Where the computer can't model fillets and other tricky transitions, the designer supplants the computer with drawing skill. Because a concept's geometry is stored three-dimensionally in the computer's memory, underlays for any subsequent views the designer chooses from the screen will represent the object as accurately as the original (virtually impossible to achieve with hand-drawn views) and in a fraction of the time for a conventional drawing.

In This Issue: When are traditional techniques best? What can CAD bring to the product development process? Can CAD generate finished products, such as labels? Have the demons of lost and dehumanized jobs reared their heads? Will increased design productivity translate into extended exploration or cookie-cutter design? How can CAD help in meeting shortening lead times if it takes a long time to learn? Will it help industrial design communicate? Will industrial design lose or gain product control, lose or gain prestige and stature? These are just a few of the questions the articles in this issue will answer.

Looking Ahead

Tomorrow's CAID systems will go far beyond today's drafting, documentation and rendering to help with the full design process from proposal preparation to ergonomics, from structural analysis to vendor contact. **Finite-element analysis** (FEA) and other analytical tools already provide powerful means for optimizing product design. But future systems will go beyond analysis. The next wave will be **expert systems**, "computerized consultants that harness the knowledge of the best in the field and provide advice concerning a specific problem," in David Svet's words.

Special kinds of expert system software, called **daemons**, will labor unseen in the background. They will continuously and automatically analyze critical aspects of designs and, when given permission, will optimize the designs according to predetermined rules and trade-off formulas. The most sophisticated CAID systems will have several "specialist" daemons, overseeing mechanical engineering, ergonomics, aesthetics, cost, quality control, environmental impact and so on. These specialists will cooperate more effectively than any human committee can, keeping steadfastly to the rules set down by the designer. The ergonomics daemon would wave a red flag when the designer violates an ergonomics standard, and the aesthetics daemon would critique designs and help the designer to improve them.

Paving the way for daemons, the next generation of CAID systems will employ **parametric design** technology. To glimpse the power of this concept, let's consider the process of designing a ketchup bottle with a hypothetical system. Any physical or descriptive **attribute** of the bottle can be considered a parameter: dimensions, color, semantic associations, material, cost and so on. You can even go further and define such abstract attributes as materials and components vendors, the energy required to make the glass or the environmental impact of the process—anything, in fact, that seems important enough to take into account when designing the bottle—as long as you can define relationships with other parameters that it would affect or be affected by.

Each parameter occupies a cell in a **relational database management system**. One cell contains the parameter "wall thickness," for instance. The cells are functionally connected by mathematically- or logically-defined relationships: When wall thickness goes up, so does weight, rigidity, cost of materials and shipping, the energy required to produce the glass and environmental pollution. Other parameters, like "transparency," go down. The "semantic associations" cell has **semantic differential** scales like "light-heavy;" this scale probably moves toward "heavy" with a thicker wall. The designer can freeze a parameter at a certain value, let it vary over a defined range or let it vary freely. The daemons sort out the details and decide trade-offs, between cost and aesthetics, say, based on how the designer has apportioned power to them, in accordance with priorities. The designer can query any cell, at any time, to get the value or status of a parameter, and thereby assess any aspect of the design. Or the designer can keep a running tab on the most critical parameters.

Some cells of the database can connect with outside databases. If the designer asked for the current projected cost of the product, a daemon would dial up appropriate vendor databases containing current price schedules, query other relevant cells or databases, and plug in the values. If a cost limit had been set, the daemon would periodically check the situation on its own and call the designer's attention to potential cost overruns.

If all this sounds too restrictive, consider this: In the final analysis, a product fails because its designer has taken too narrow a view of the problem, because some critical constraint has gone unnoticed or disregarded. This happens even to the conscientious designer who tries to be

comprehensive and think of everything. It's often impossible to keep all the balls in the air. Nor can every designer be an expert in everything. Daemon-guided parametric systems promise comprehensive designers opportunities to come closer to their ideals than ever before because lightening-fast daemons do most of the juggling, are as expert as the expert minds tapped to program them and never miss a trick.

Aesthetics Daemons: Quantifying Aesthetics
The notion of quantifying aesthetics bothers some designers. . . . Oh, hell! It bothers *most* of you! Designers willingly let computers solve a myriad of pragmatic problems, but they are loathe to let them mess around with aesthetics, the last refuge of the designer's self-esteem. So I approach my final subject—solving aesthetic problems with computers—with trepidation. Yet, after trodding this hallowed ground for over 20 years, the affirmative evidence is too abundant to ignore: Beauty is quantifiable.

As luck would have it, aesthetics can be quantified in units of **information**, the same **bits** and **bytes** computers use. Like a word processor document, an object designed with a CAID system can be described as a distribution of information over its visible surface. The pattern of distribution matters most, not the total amount. One page of text brings tears to the reader's eyes, while another, with the same number of characters (bytes) but a different distribution, brings laughter. Similarly, one ketchup bottle seems "light," while another, which holds the same quantity of ketchup, seems "heavy" because the distribution of curvatures on their surfaces (also measurable in bytes) differ.

Looking back on those early experiments at The University of Michigan, I realize now that, by restricting the variety of input data, I merely dispersed the information more evenly throughout the designs. That's all **objective concinnity** turns out to be-- a relatively uniform distribution of information potential. In the extreme, the characters on a page of text with maximum objective concinnity are all identical. The result may be elegant visually, but semantically inappropriate for all practical purposes because it says nothing relevant. I could have maximized the objective concinnity of my car designs by assigning, not just harmonious numbers to every patch, but identical numbers. But the results would not have resembled cars. When the objective concinnity of a 3D object is increased, it approaches that old Greek ideal, the sphere. Elegant and timeless, but functionally and aesthetically inappropriate for most applications, including cars and ketchup bottles.

Every manipulation of an object's surface (changes of curvature, proportion, orientation, color or whatever) affects the distribution of information and, thus, the semantic appropriateness of the thoughts and feelings it evokes. One configuration sends the wrong message (the car seems slow, awkward and rough) while another, with greater **subjective concinnity**, suggests the car is fast, nimble and smooth.

The ideal CAID system might need two aesthetics

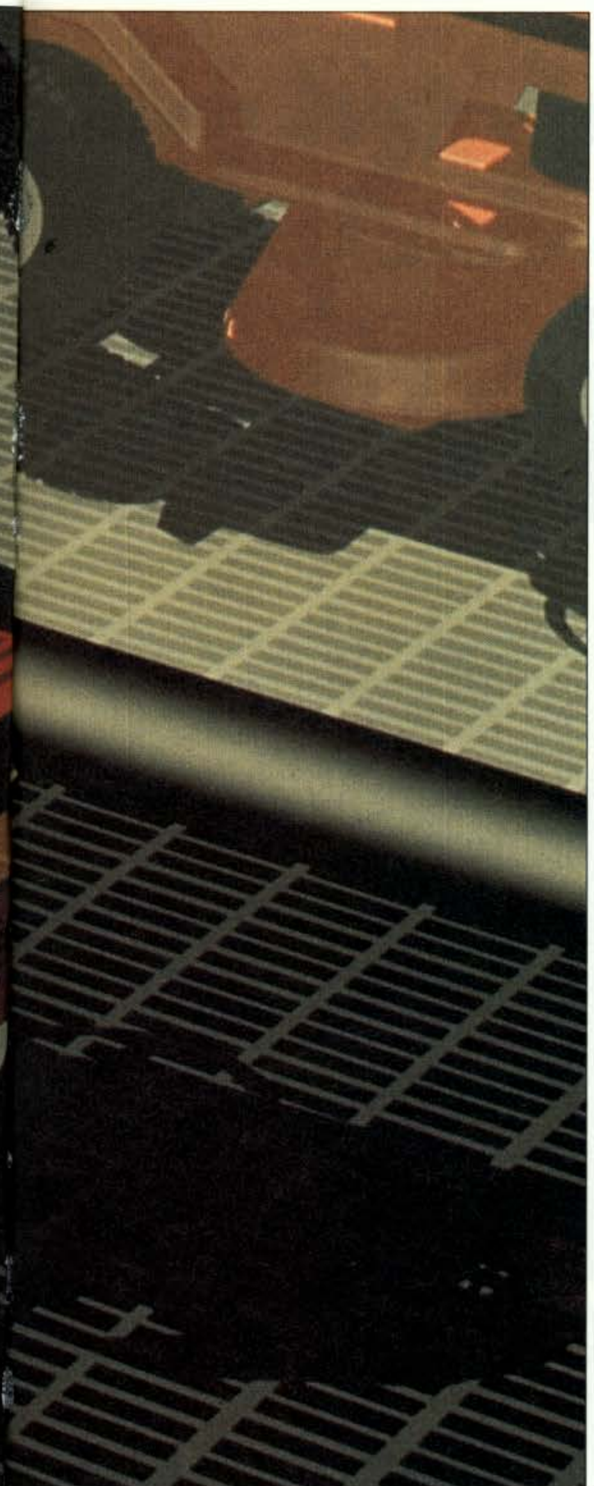
This car is a Volkswagen Passat, modeled on the CDC/ ICEM system. The data was transferred into Alias, via the European data transfer protocol VDAFS, for rendering. The background was scanned on a Howtek scanner and texture mapped onto the car to simulate the reflections on a glossy surface. These simulated reflections tradeoff true photorealism for improvements in speed and cost. (Produced by CDC in West Germany.)

Produced on an Intergraph system, this image of a Murray of Ohio garden tractor demonstrates how ray tracing can produce realistic reflections and shadows. Ray tracing renders the specular reflection that characterizes glossy surfaces producing actual photorealism.

daemons, one responsible for each kind of concinnity. The one responsible for objective concinnity would always try to distribute a design's information as evenly as possible. Rather than engaging in a continuous tug-of-war with the designer, trying to force everything into a sphere-like form (or, at least, as though it came from the Bauhaus), its diligence would be limited by an objective concinnity control at the designer's fingertips. The ketchup bottle designer would set the control high enough to smooth his freehand sketch of the bottle's profile—as though he had used french curves to refine it—and not high enough to wash out the essence of the sketch.

This single profile sketch is all the input the parametric CAID system would need for a photorealistic rendering of a finished product. The designer would have already set the necessary parameters for volume, wall thickness, the color of ketchup and so on. The system would pull an existing parametric design for a cap from a **standard parts library**. It could also pull a finished label design from another library and map it onto the surface.

We'd hardly expect the designer to be satisfied with this one five-minute exercise. He'd want to explore further, perhaps with a few variations on the theme by modifying the original profile drawing. He could have photorealistic renderings of each variant in minutes and display them side by side for careful comparison. Or he could take advantage



of the computer's potential by letting it create the permutations automatically. He'd kick off the process by letting selected parameters vary. Assuming that the initial design has a traditional ketchup bottle shape, one variable parameter might be the approximate point where the bottle bulges out from the neck to the main body. The computer could generate ten new designs by locating the point of transition at ten equally spaced heights between upper and lower limits marked by the designer. The objective concinnity daemon would apply french curves, in effect, to each profile to smooth it suitably. Other daemons would preserve the bottle's fixed volume, overall height and other specified parameters.

The computer has much more potential, of course. It could spew out 100 variants if it also did ten different diameters for the bulge, 1,000 if the bottom diameter varied as well, and 10,000 if the designer instructed the computer to also come up with ten slight variations of his original profile. While this would tap much of the computer's potential, it would vastly exceed the designer's ability to separate the wheat from the chaff. This is where the aesthetics daemons would come to the rescue by doing the initial screening.

The objective concinnity daemon would throw away any designs that didn't fall within, say, 50-60% of maximum objective concinnity (people seem to prefer designs with moderately high, not extreme, levels). The subjective concinnity daemon would narrow the field further by selecting those designs which met certain semantic requirements. Let's suppose, for example, that the client planned to market a spicy version of the ketchup along side the normal flavor. The subjective concinnity daemon would know, on the basis of **semantic differential** surveys of consumers, which surface characteristics seemed relatively "hot" and which seemed "cold." It would then sort the remaining bottles into hot and cold groups (it would further reduce the total by throwing away designs that were too neutral). Finally, the objective concinnity daemon, most talented at comparing forms for their similarity, would match up pairs from the two sets that, although markedly different in semantic temperature, would have some family resemblance to promote product identity. The process might be guided by a general instruction from the designer: "Select 20 pairs with; Condition 1—maximum difference within pairs on hot-cold semantic dimension; Condition 2—maximum similarity within hot-cold pairs." The computer would repeat the sorting procedure until it had narrowed the field to the specified 20 designs. This being a reasonable number to judge, aesthetically, the designer would make the final choice and present it to the client.

This example still represents only a fraction of CAID's potential. Procedures that tapped randomness, fractal geometry, and the newly emerging field of chaos theory could conjure innovative forms that might never occur to a designer, even with years of exploration, and trigger new directions of thought. With the sketchiest of inputs, the designer could easily set a CAID system on a course that would yield hundreds of thousands, even millions, of designs. But all this mind-opening, creative potential would go for naught without some help with the crucial winnowing process. In such a world, aesthetics daemons would be agents of practical necessity, not merely philosophical choice.

The WordMap, a Source of Definitions
This issue closes with the "WordMap to CAID." It defines CAD and CAID terms that you will encounter in this issue and in other literature and discussions on the topic. The ***boldfaced*** *words in each article are defined in the WordMap. They are only boldfaced the first time they appear in that article.*

I want to acknowledge David Svet's help in defining some of the artificial intelligence terms.