Editor: Frank Bliss

Ten CAD Challenges

David J. Kasik
The Boeing Company

William Buxton
Buxton Design

David R. Ferguson
DRF Associates

The world is enamored with the number 10. Perhaps it's because arithmetic is a lot easier when the number 10 is part of an operation. Or maybe David Letterman and many others have finally achieved a long-term impact on society with their top and bottom 10 lists.

Ivan Sutherland used the number 10 to bring significant consistency to hidden surface algorithms for computer graphics in his classic paper.[1] It's in honor of Sutherland—developer of the original Sketchpad application, the forerunner of today's computer-aided design (CAD) applications—that we describe our 10 CAD challenges. The concept of CAD itself has expanded into computer-aided manufacturing (CAM) and computer-aided engineering (CAE). Basic CAD techniques are reapplied in a number of places, including electrical and mechanical product development, buildings, and entertainment. In this article, we focus on mechanical product development.

CAD, CAM, and CAE generate massive amounts of data that must be clearly organized and placed under strict configuration control. The CAD industry also provides support to these functions through product lifecycle management (PLM) systems. In assessing the state-of-the-art CAD, we're examining a reasonably mature, respected, and accepted form of technology. The industry's multibillion dollar yearly revenues have been reasonably flat for the past few years. For an introduction to CAD's history, see the "Brief CAD History" sidebar (next page).

We could place our 10 challenges in a number of different categories. We've chosen three: computational geometry, interactive techniques, and scale. Given our knowledge of the state of the art, the categories we would have chosen 10 years ago would likely be different. Each of the categories presents an opportunity to expand CAD's penetration to new user communities and increase its long-term impact.

Geometry represents the kernel data form that a designer uses to define a physical product in any CAD application. Correct assembly of the geometric shape, structure, and system components is the basis for figuring out how to produce the complete product (through CAM) and how the product will respond once in service (through CAE). Only a few commonly used libraries implement computational geometry algorithms, an indication of the technology's relative maturity. Our analysis identified three specific geometry challenges:

- geometry shape control;
- interoperability across CAD, CAM, and CAE applications; and
- automatically morphing geometry in a meaningful way during design optimization.

We discuss the significant technical challenges facing the CAD industry and research community and present an approach to address these issues.

The key component of all CAD applications is the end user. Users' success or failure in interacting with CAD products ultimately governs the success of the products themselves. We extend the meaning of *interactive techniques* for this article to include more than low-level input device, display, and human factor issues. We deliberately include the way users work with applications to accomplish a work task. Using this extension lets us define three difficult challenges with interactive techniques:

- the order and flow of the tasks a user performs to accomplish something,
- the relationships among tasks in a complex design environment, and
- the manner in which old designs can effectively seed new ones.

The third category describes the challenge of scale. In many ways, the CAD market has reached a plateau just because it has not yet discovered ways of going beyond its current limits. We introduce scale challenges that slow the progress of overall CAD penetration. Should these challenges be addressed in a meaningful way, larger growth in CAD business will likely occur. The four scale challenges include

0272-1716/05/$20.00 © 2005 IEEE

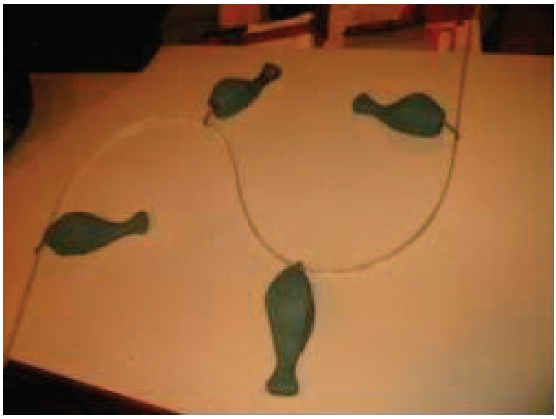
Brief CAD History

CAD was one of the first computer graphics applications in both academia and industry. Ivan Sutherland's Sketchpad at the Massachusetts Institute of Technology and the DAC-1 project at General Motors both started in the early 1960s. Industry developed its own CAD applications, delivered on multiuser mainframes, in the late 1960s and 1970s. The 1980s turnkey systems bundled hardware and software. Current CAD implementations separate hardware and software components. As a result, CAD software most often executes locally on powerful Unix or Wintel workstations with specialized 3D accelerator hardware and stores data on distributed servers.

Even though engineering drawings were the dominant output of now-defunct CAD vendors—for example, Lockheed's CADAM and commercial companies like AD2000, Applicon, Autotrol, ComputerVision, Gerber IDS, Intergraph, Matra Datavision, and VersaCAD—through the 1980s, early stages of design relied on a variety of unique curve and surface forms. Customized systems were used in aerospace for surface lofting (such as TX-95 at Boeing) and surface design (such as CADD at McDonnell-Douglas) and in the automotive industry for surface fitting—such as Gordon surfaces (General Motors), Overhauser surfaces and Coons patches (Ford), and Bezier surfaces (Renault). Current CAD applications rely on more general geometric forms like nonuniform rational b-splines (NURBS).

Solid modeling also started in the late 1960s and early 1970s but from different roots. Larry Roberts worked at MIT to automatically identify solids from photographs. The Mathematics Application Group Inc. used combinatorial solid geometry to define targets for nuclear incident analysis and subsequently developed ray-traced rendering and solid modeling in Synthavision. Other efforts—for example, TIPS from Hokkaido University, Build-2 from Cambridge, Part and Assembly Description Language (PADL) from the University of Rochester—had limited industrial impact.

Solid modeling is the 1990s preferred technique for defining 3D geometry in small through large companies, and good modeling software is readily available. The use of 3D is common in computer animation, aerospace, automotive, (and is becoming acceptable for building architecture), and 2D is also used effectively.



Courtesy of Dave Ferguson

1 **Early meaning of spline weights.**

- finding ways to cope with vastly larger quantities of data,
- communicating key concepts to user communities traditionally outside the CAD community,
- reliably migrating data to new versions of software and hardware as product life spans increase, and
- improving productivity for geographically distributed project teams.

As we built this set of challenges, we realized that the sum of the challenges was greater than we expected. Therefore, we suggest an approach that can help others understand and manage the changes needed. We believe that the technical challenges we've identified will lead to fundamental changes in the way people work.

Geometry

Geometry lies at the core of all CAD/CAM/CAE systems and, while not the only item of interest in product design and development, it's the sine qua non of design. Geometry includes both the algorithms and mathematical forms used to create, combine, manipulate, and analyze the geometric properties of objects: points, lines (curves), surfaces, solids, and collections of objects.

Geometry begets three fundamental questions: What are the objects to be represented? What mathematical forms and approximations will be used to represent them? How will information about the representation be computed and used?

Given the power and generality of current design systems, we can imagine that all issues related to these questions had been adequately answered. However, this is not the case. We discuss three geometry challenges (shape control, interoperability, and geometry in design exploration) that still require extensive research and development.

History

To illustrate the evolution of geometry methods and usage we focus on five time periods: pre- and early 19th century and early, mid, and late 20th century. In each of these periods there was a fundamental shift in the methods and usage of geometry driven by new design requirements.

Prior to the 19th century, the major use of geometry was to define and maintain line drawings for manufacturing and records keeping. The tools and methods were rudimentary: Lines were constructed by ruler and compass methods and curves were traced from a draftsman's or loftsman's spline, a thin wooden or metal strip bent and held to a desired shape using weights (see Figure 1).

In the early 19th century, these methods were augmented with descriptive geometry, primarily using second-degree algebraic equations, to provide more precise mathematical descriptions and increase accuracy and precision in engineering drawings. The objects of interest were still lines, but descriptive geometry required new tools (for example, slide rules and tables) to aid in calculating various curve properties (such as point location). Descriptive geometry changed from curves drawn on paper to precise mathematical formulas from which any point on a curve could be calculated accurately.

The next major advance took place early in the 20th century. Catalogs describing functionality and application of particular families of curves in engineering began appearing. The catalogs were based on careful scientific analysis and engineering experimentation. For example, the National Advisory Committee for Aeronautics (NACA) generated a catalog of curves that classified airfoil shapes according to flow properties (see Figure 2 and http://www.centennialofflight.gov/essay/Evolution_of_Technology/airfoils/Tech5G1.htm and http://www.centennialofflight.gov/essay/Evolution_of_Technology/NACA/Tech1.htm). Objects became more than curves; they also had attached properties that described their appropriate use in various design activities. Design could now begin with curves known a priori to have properties needed for a viable design.

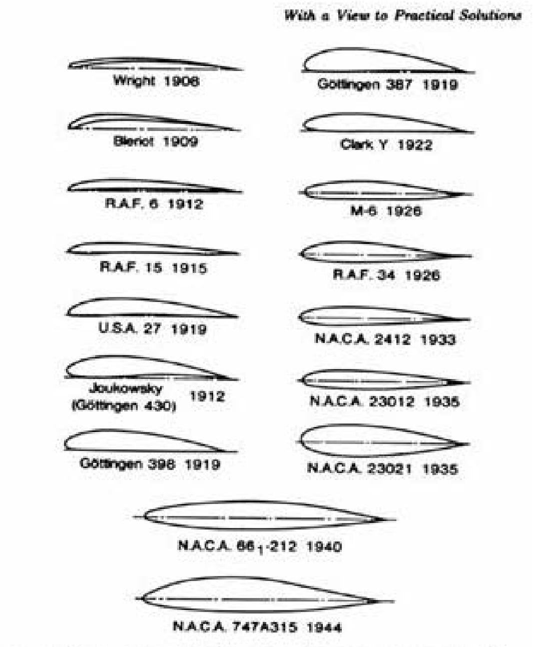
In the 1940s, Roy Liming at North American Aircraft introduced actual surface representations in his conic lofting system.[2] These were not like today's surface representations, but his system did provide for complete surface definitions rather than simply a family of lines. It was now possible to think of mathematically computing mass, aerodynamic, and hydrodynamic properties. At this point, geometry began changing from describing a physical object to becoming the base for calculating engineering characteristics. This lessened the dependence on physical models and began the push toward virtual design. Tools also began changing. With the increased emphasis on analysis, calculators and computers became required tools.

Liming's conic lofting methods proved their worth in the design of the P51 Mustang airplane (see Figure 3). Liming boasted that the Britain-based Mustangs could fly to Berlin and back because their surface contours did not deviate from the mathematical ideal.[3] However, it was not possible to visualize the geometry without a physical prototype or mock-up.

Modern graphics systems made it possible to display 3D geometry. Designers could visually inspect designs to find mismatched or ill-fitting parts and shape defects. The new display technologies required radical change in how geometry was represented: Everything in a design had to have a precise mathematical representation. Gone were the line drawings used by draftsmen. The old algebraic methods of descriptive geometry were displaced by mathematical splines, NURBS, tensor product splines, and other analytically based forms.

Having a complete mathematical description brought another fundamental change. Previously, the preferred method of constructing a curve or a surface was highly intuitive: lay out sequences of points; construct curves that pass through the points; and then construct a surface from the curves by cross-plotting, conic lofting, or some other means. Using purely mathematical algorithms to interpolate or approximate opened up new possibilities: Curves could be defined by approximating a sequence of points, surfaces could be defined by clouds of points, and geometry could be constrained directly to satisfy engineering constraints (for example, clearances and shape).

Geometric methods and objects used to support product design and definition have changed significantly. As design objectives and systems continue to respond to the ever-increasing need to design rapidly, efficiently, and virtually, geometric methods will need to meet the challenges and change accordingly.



The historical evolution of airfoil sections, 1908–1944. The last two shapes (N.A.C.A. 66_1-212 and N.A.C.A. 747A315) are low-drag sections designed to have laminar flow over 60 to 70 percent of chord on both the upper and the lower surface. Note that the laminar flow sections are thickest near the center of their chords.

Courtesy US Centennial of Flight Commission

2 NACA Catalog of Airfoil Shapes.

3 P51 Mustangs.

Challenge 1: Shape control

The success that graphics had in forcing everything to have a precise mathematical representation actually increased concern over inflections. The hand-drawn and hand-constructed methods, including Liming, had implicit control of shape whereas the new, polynomial and piecewise polynomial methods (splines, B-splines, and so on) did not. Inflections in a curve passing through a sequence of data points are possible using polynomials or piecewise polynomials even though the data points do not suggest inflections. Therefore, it became important to devise algorithms that not only fit points but also allowed shape control. Shape control defines the occurrence and position of inflection points (points at which the signed curvature of a planar curve changes). The control is needed for both engineering and manufacturing optimization.

Researchers have proposed various schemes for shape control. Most failed because there were always cases for which the methods failed to properly preserve shape. Many methods exist for detecting shape anomalies after the fact, and a person must fix the anomalies by hand. Removing the person in the loop lets geometry pass directly to other applications for optimization. The challenge becomes finding algorithms that avoid anomalies in the first place.

Attempts to modify the methods used previously to account for shape control did not work in general, and it wasn't until the 1980s that we realized the basic principles of those methods were wrong.[4] Although nonlinear methods[4] have proven themselves in a variety of shape control situations, most of these methods have not made their way into commercial CAD systems.

Challenge 2: Interoperability

Current CAD systems do not integrate well with CAE analysis (such as structural mechanics, fluid dynamics, and electromagnetics).[5] For example, computational fluid dynamics (CFD) must interrogate geometry quickly and reliably. Most CFD codes construct a computational grid from the geometry. Building the grid reliably means that there should be no unintended holes in the geometry—that is, the geometry should be what CFD practitioners refer to as watertight. Real geometry from CAD systems is rarely watertight.

Geometry from one CAD system is difficult to translate reliably into another. Estimates peg the cost of interoperability in the US auto industry at $1 billion per year.[6]

Holes, translation errors, and other problems arise from two major sources: floating-point arithmetic and tolerances. Floating-point arithmetic, which forces approximations in numerical calculation, is addressable theoretically but not practically. We could impose higher precision (double, triple, and so on) to drive down the resulting errors. Or we could change to a rational arithmetic system and eliminate the need for floating point. Digital floating-point arithmetic is a research area by itself.[7]

Tolerances control the accuracy of computed solutions and are a fact of life in today's CAD systems. A simple example involves calculating the curve of intersection between two surfaces. When the algebraic equations representing the geometry are simple (for example, a plane or a circle), a closed-form solution for the intersection exists. However, closed-form solutions generally do not exist for operations on equations of a sufficiently high degree (for example, intersecting two bicubic surfaces). Computing the intersection curve uses approximation, a problem independent of precision. Some CAD systems will recompute intersection curves if more accuracy is needed. This doesn't solve the problem, especially if the intersection curve is used to generate other geometry.

Tolerances are needed to control the approximation.[8] Too loose a tolerance can give results that are fast but incorrect. Too tight a tolerance can result in poor performance or failure to converge. Even seemingly simple surface-to-surface intersections become difficult because of choosing tolerances.

Tolerances determine the success of downstream engineering (CFD, finite element) and manufacturing (numerical control programming, quality assurance) analyses. Selecting a tolerance that guarantees a high probability of success requires that the geometry generator understand the kinds of analyses to be employed, the environment of the analyses, and even the specific software to be used a priori.

In summary, digital arithmetic and current math theory are insufficient to perform reliably for complex geometry operations and to interoperate well with downstream analysis software. The geometry must be as watertight as possible for downstream use, and algorithms cannot result in topological inconsistencies (for example, self-intersections and overlaps). The challenge is to find ways to deal with poor results. Perhaps a new math theory that has closed-form solutions for complex surface operations and supports watertight representations for downstream analysis is the way to address this challenge.

Challenge 3: Design exploration

Automated design exploration through multidisciplinary optimization presents the third challenge. Design optimization requires that geometry remains topologically valid as parameters are perturbed while preserving the designer's intent. There are two aspects to consider: how to parameterize the geometry for downstream analysis and how to structure geometry algorithms to support continuous morphing, a key to any optimization process. The former is primarily an engineering function, which we do not discuss here.

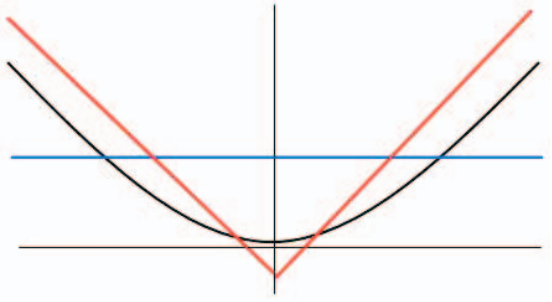
Morphing is a requirement that CAD systems do not currently support. Morphing algorithms today allow the hole in the upper block to flip into the lower block when the edges of the two blocks align. This is fine geometrically. However, this is a disaster for optimization, because the geometry does not morph continuously with the parameters.[9]

The challenge is to design and build geometry systems that ensure the continuity of morphing operations. Morphing continuity differs from geometric continuity. Geometry often has discontinuities (for example, tangents) that must be preserved during morphing. Morphing continuity means that the geometry doesn't change suddenly as parameters change.

Parameter values must be simultaneously set to reasonable values to ensure valid geometry for analysis and optimization. Automating morphing is a challenge because CAD systems have evolved as interactive systems that let users fix poor results. Design optimization needs a geometry system that automatically varies parameters without user guidance and yet maintains design integrity and intent.

Multidisciplinary design causes us to rethink the geometric design process as well as the algorithms. For example, many CAD systems use a piecewise quadratic or cubic algorithm for defining a curve through a sequence of points. These algorithms will not reproduce an embedded straight line exactly. Preserving embedded line segments forces the curve fit algorithm to be modified whenever three successive points lie on or are near (determined by some system tolerance) a straight line.

Figure 4 contains a second example of geometry that seems reasonable but causes problems during morphing. Consider the surface $H(x, y, z) = y - ax^2 = 0$ so its intersection with the $z = 0$ plane is described by the function $f(x) = ax^2$. Parameter a is a shape parameter that varies according to an optimization process. Suppose the algorithm for approximating intersection curves uses piecewise straight lines and the fewest join points possible to achieve a certain tolerance. Now suppose the tolerance is 0.5. For values of a approximately equal to 1, the approximation will alternate between a function with constant value 0.5—that is, a spline with no knots—and a piecewise linear function—that is, a spline with one interior knot. In other words, varying the parameter a from slightly less than 1 to slightly greater than 1 causes the model to change discontinuously. As a passes through the value 1, the intersection curve not only changes shape (the blue line morphs to the red line in Figure 4), but its properties (for example, arc length) change discontinuously. This algorithm design produces good static approximations but fails during morphing.



Courtesy Dave Ferguson

4 **Blue curve morphs discontinuously into red curve.**

Summary

Addressing the geometry challenges outlined here will not be just a simple task of going through and improving or deleting offending algorithms. It also requires a fundamental rethinking of how geometry design systems should work.

Interactive techniques

Significant improvements in interaction are not going to be achieved by making more efficient menus or a better mouse. Rather, they are going to depend on rethinking the nature of the process. Three specific challenges result:

- changing the order of workflow,
- understanding the concept of *place* in the workplace, and
- intentional design.

History

Interaction technology has evolved slowly over the past 40 years. Input devices have not changed significantly. Physical buttons, such as keyboards, function keys, voice, fingers, and so on, let people talk to the machine. Graphical pointers come in numerous shapes and sizes and let us move in two or three dimensions, with similar resolution to early devices. The graphical user interface has remained essentially unchanged since 1984.

Displays have gotten a lot smaller and a lot larger. Smaller displays are ubiquitous because of the phenomenal growth in the cell phone industry; larger display penetration is steadily growing. The basic resolution of display devices (number of units per square inch) is about the same as Sutherland's Sketchpad was in the early 1960s.

Improvements have been achieved largely by adding new functionality, enhanced graphic design, and better flow of control. However, systems are not easier to use, and the demands on the user today might be even higher than 20 years ago. The complexity of designs, data, and geometry are growing as fast, or faster, than the power of the tools to handle it.

Nevertheless, some things are changing, and these changes will afford the potential to break out of this situation. For example, more than 10 years ago, Nicholas Negroponte challenged people to imagine what would be possible if bandwidth was essentially free. The only thing that matched how amazing and unlikely that concept was at the time was its prescience.

The parallel challenge today would be, "Imagine what would be possible if screen real-estate were essentially free." Already, paper movie posters are being replaced by $10,000 plasma panels. Imagine the potential impact when the cost of a comparable display drops two orders of magnitude and it is cheaper to mount a 100-dpi display on your wall than it is to mount a conventional whiteboard today.

Large displays will be embedded in the architecture of our workspace. We will stroll through and interact with such spaces with agile small, portable, wireless devices. And many of the changes that are going to affect the future of CAD will emerge from the evolving behavior of both people and devices as they function within them.

Within this context, a divide-and-conquer approach will be used to address the complexity posed by today's CAD systems. While power will come in numbers, most will be relatively inexpensive and target a particular function. The isolated gadgets that first emerge as add-ons to existing systems will morph into the keystones of a new mosaic of integrated technologies that will transform the process.

To realize the potential of this, the following three challenges involve thinking about evolution in a human-, not technology-, centric way.

Challenge 4: Reversing engineering

Important changes do not result from doing the same things faster or for less money. They come when you flip approaches and methods on their head, that is, we must constantly ask ourselves, Do we do things the way that we do today because it is the right way, or because it's the only way that we knew how when we started?

The inertia of the status quo blinds us to the recognition that major changes, such as those due to Moore's law, open up different and desirable approaches. If we overcome that inertia, we can explore other approaches.

In the field of aerodynamics, for example, it's truly important to get designs that are efficient and safe. Government agencies such as NASA and companies such as Boeing spend millions of dollars on tools, such as wind tunnels and CFD analysis, that help them test such systems. However, these tests typically come late in the design process. Now combine this fact with one of the most basic rules of design: The later in the process that a mistake is detected, the more expensive it is to fix.[10]

Replacing this rigorous testing at the back end is not the answer, but it would be more efficient if technology existed that allowed preliminary tests on the desktop or even on PDAs. Designers could then catch bad designs much earlier in the process. This would also help them discover, explore, refine, and understand the most promising designs as early in the process as possible.

This approach to CFD could be applied to a number of other parts of the CAD workflow. Consider the ability to introduce elements such as stress testing, volume or strength calculations, and so on, earlier in the workflow. This would allow designers to consider the interconnection and interoperability of parts much earlier in the design cycle than is currently the case.

Adding simulation earlier into the design process will enable consideration of behavior, rather than just form, to play into the process. The fundamental change in workflow is the essential interactive technique that will make the geometry design exploration challenge of the previous section a useful and usable capability.

Challenge 5: Everything in its place

There was a time when someone's location in a CAD environment indicated their current job. For example, in an automotive design studio, there is often one location where people deal with interiors, another contains the full-sized clay models, and some other location is set up for exploring colors.

However, the typical modern CAD environment contains a uniform sea of anonymous cubicles or desks. In general, anyone walking through such a space will not likely be able to tell if they are in the accounting or the engineering department. Because people don't move around anyhow and are essentially anchored to their desk, this organization isn't important in some ways. Yet, in the midst of all of this, we hear an ever-louder call for collaboration. We also hear of the emergence of ubiquitous computing. How do these two points relate, if at all, in terms of transforming the CAD workplace?

To begin with, ubiquitous computing will break the chain that anchors the engineer to a general-purpose workstation. This change will not just enable but necessitate designer's moving from one specialized area to another in a manner harkening back to the best of past practice.

Not only will the workspace be broken up into specialized areas with specialized tools, but these tools will also consist of a combination of private and public displays and technologies. Some will be mobile and others embedded in the environment. Examples of the latter would include large format displays that function as digital corkboards, surfaces where you can view large parts on a 1:1 scale, and areas where you can generate physical parts using a 3D printer.

The physical mobility of a person and data can greatly impact agility of thought. Mobility brings increased opportunity for collaboration and increased visibility of a particular activity. Rather than design a system that lets us send more email or documentation to the person at the other side of the studio, the studio should be designed so that we have a greater probability and opportunity to bump into and work with that person face to face. If we are going to have work-across sites, then linking designers' efforts must be linked automatically as a consequence of undertaking a particular activity.

Our notion of space is not about making things more abstract or virtual. Rather, it concerns the recognition and exploitation of the attributes and affordances of movement and location in a technology-augmented conventional architectural space. We discuss the impact of collaboration among the geographically distributed in challenge 10.

The challenge to future systems is as much about human–human interaction as it is about new forms of interaction between human and machine.

Challenge 6: What we do

CAD companies might view themselves as primarily purveyors of tools for the creation of high-quality 3D models. Consequently, if they want to grow their business, they might conclude that they should make a better, more usable modeler.

However, this stream of thought might be as wrong as it is reasonable. Consider the following questions:

- Is there a shortage of trained people to fill the existing demand for creating 3D models?
- Are there things that need to be modeled that cannot be built by existing users with their current skills and tools?
- Is there a huge untapped market for 3D models waiting for an easy-to-use modeling package that takes little training to use?

In general, the basic answer to all of these questions is "No." It's not modeling—at least in the sense that it exists in today's CAD packages—that lies behind any of the fundamental challenges outlined in this article.

In fact, it might well be that all of these questions are poorly posed, since they all assume that the intent of the user is modeling. It is not. Rather, it's getting a product or a part made. Modeling is just one way of doing so, and in many cases, not always the best way.

We are then challenged to answer this better question: What besides modeling from scratch might enable us to achieve our product design? A good answer to that question should lead to a more innovative and effective solution than today's design-from-scratch modeling approach.

Our favorite recourse in such cases is to look to the past. In this case, the clue lies in Figure 2, the NACA catalog of airfoil sections. As we discussed previously, such catalogs let designers build an aircraft by selecting components with known properties rather than working from scratch.

We suggest moving back to this approach—but with appropriate modifications. As product complexity increases, the CAD process will subdivide into those specialists who design and build components (often out of subcomponents) and those who make larger assemblies out of them. However, the components in future CAD systems will not be fixed objects made up of what computer scientists might call declarative data. Rather, they will include a strong procedural component.

The basic problem is that objects, even those as simple as an airfoil, don't easily scale. If you make exactly the same form, but larger, the plane might not fly. A simple example is the hummingbird. It can definitely fly. However, if you scaled it up in size by a factor of 10, it could not.

In this object-oriented process, designers generally do more than just select components and plug them into a larger assembly. Rather, they take components and transform them into what is needed. The components act as a starting point, something designers work from, rather than start from scratch.

Given that physical products don't scale, one key aspect of component design is the component's embedded capacity to be transformed along meaningful and desired dimensions, while maintaining specific tolerance, performance characteristics, manufacturability, maintainability, and so on.

Embedded in the component catalog, therefore, are just not the parts, but the knowledge of how parts can be transformed while maintaining essential properties. Consequently, the tools of the CAD engineer not only help the designer find the appropriate components but also support transforming them in such a way as to maintain the desired properties.

Finding, cloning, modifying, morphing, and adapting will largely replace constructing from scratch. In so doing, the assumption is that methods such as aerodynamic testing, stress analysis, and others, can occur in advance for each component and carry over into an assembly. This approach transforms the current practices of who does what, where, when, why, and how.

Summary

Emerging user-interface technology will allow us to transform how we work. Furthermore, the new technology can help us implement the transformation without any major discontinuity with respect to current practice.

Our sense and analysis of interaction needs to switch from how we interact with a specific computer or package to how we interact as people, how we interact with change, and how we interact with our materials—at what level, in what way, and to what objective.

Scale

The challenge of scale is one that scientists and engineers continue to encounter as they push the limits of macro- and nanotechnology. Moore's law governs the expansion of computing hardware's limits. Advanced hardware lets us produce larger quantities of data. Software advances tend to lag behind more powerful hardware, yet we seem to be able to consume computing hardware resources at a rate that always leaves CAD/CAM users begging for larger storage capacity, greater network bandwidth, and better performance.

Managing scale results in a delicate balancing act of knowing when "good enough" occurs. While there are numerous dimensions of scale in CAD/CAM/CAE, the primary areas of challenge today include

- Sheer data quantity that can exceed a project team's software, hardware, and cognitive capacity.
- Making critical data comprehensible to other users.
- Keeping data meaningful as product life spans become longer.
- Collaboration with a geographically distributed workforce.

The challenges of scale in this section derive from experience at Boeing, a large and varied aerospace manufacturer. While Boeing does not represent the norm, similar problems exist in automobile production, shipbuilding, and other manufacturing companies. More importantly, solving a Boeing-sized problem has often resulted in breakthroughs that have profoundly affected smaller efforts.

History

Scale has grown as the overall process of designing, building, and maintaining complex products has changed.

Humans have designed and built extraordinarily complex artifacts throughout history. The seven wonders of the ancient world are clear examples: Roman aqueducts still deliver water service to Italian cities, and the Great Wall of China still stands, acting as a tourist destination and is clearly visible from the ground and Earth orbit.

Documentation on the ancient design process is sketchy, but became more formal during the Renaissance. Leonardo da Vinci's sketches of various mechanical possibilities captured a more formal depiction and allowed designers to analyze possible configuration strengths and weaknesses before construction started. These practices continued to evolve into highly accurate renderings documented as engineering drawings in the first half of the 20th century. The evolution of computational geometry described earlier has taken us to the point we are now in the 21st century.

The challenges we discuss in the rest of this section are the direct result of change in the fundamental design-analyze-build-maintain process. Until the latter part of the 20th century, highly complex projects featured integrated design-build teams. Slave labor was the dominant force in the earliest build teams, and on-site designers oversaw every detail of construction. Large efforts took a long time and involved thousands of workers. There has been a desire to decrease construction time and the number of people ever since.

As increased specialization in individual aspects became more prevalent, principal design and assembly continued to exist within close geographic proximity. Consider the evolution of design-build at

Courtesy Boeing

5 **Boeing Red Barn.**

Boeing. Boeing's first airplanes were designed and built in the Red Barn (see Figure 5). As build problems occurred, engineers could walk to the factory to make on-site corrections.

As automation and the complexity of the product increased, later airplane generations continued to be designed and built in the relatively close proximity of the Puget Sound region in Washington. Boeing located the engineers designing the 737 and 757 within minutes of the Renton assembly plant; Everett housed the 747, 767, and 777. These engineers were responsible for the design of all of the major sections of all airplanes.

With the 787, its next-generation airplane, Boeing is distributing the design, manufacture, and assembly of major subsections across the world. Each major supplier will deliver preassembled sections of the airplane to final assembly, rather than delivering smaller components as occurred with previous models. This approach will dramatically reduce final assembly times. Computing and communications systems are mandatory for this to occur, and huge amounts of data must be integrated and managed. Because the design is completely digital and 3D, a larger variety of people will look at images of the data during the 787's life cycle. Time and distance complete the notion of scale: Mechanical products built today have longer and longer lifetimes, and geographic distance separates people.

Challenge 7: Understanding vast quantities of data

Design, engineering, manufacturing, and maintenance processes have routinely existed in virtual isolation from one another. Part of the scope challenge relates to data, especially as products become complex. Another part relates to people because the graphics vocabulary used in the design process is dramatically different from the one used when the product is being manufactured and assembled.

Early efforts in product design used the waterfall model. In this approach, designers handed drawings to engineers for analysis leading to design improvements. The drawings then made their way to manufacturing planners, people making and assembling parts, and others responsible for production maintenance. The late 1980s saw a change in the designer-engineering analysis interface because of the advent of 3D modeling systems with enough capacity to generate and manage an entire commercial airplane. Downstream users continued with master definitions represented as 2D drawings.

Initial efforts to deal with issues like manufacturability and maintainability focused on integrated design-build teams. The team members used different applications and kept data in segregated areas with different configuration management schemes.

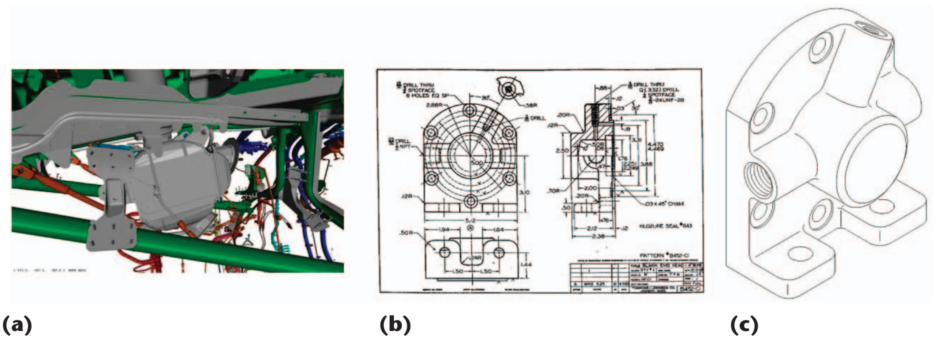
CAD/CAM software vendors are starting to address this problem with product lifecycle management (PLM) systems that extend previous product data management (PDM) systems. PLM systems let companies store customer requirements, design geometry, engineering analysis results, manufacturing plans, factory process designs, maintenance designs, and so on, in a single repository that infuses configuration management throughout.

PLM systems are difficult to sell. The essential benefit (managing data across key aspects of a manufacturing company) generates more cost in each individual area but decreases overall product costs through forced pre-facto integration and better configuration management and visibility. PLM systems compete with other large corporate systems for managing personnel, enterprise resource planning (ERP), supply chain management (SCM), and so on. Personnel management, PLM, ERP, and SCM systems all replicate significant portions of the same data. Finally, getting new technology sold, especially when the technology works behind the scenes, is always a difficult problem.

The real problems occur after product integration becomes institutionalized. Integrated products must be examined from a number of different views. Each group has a different purpose and wants to believe that their view is the master. For example, a commercial airplane has the following views:

- *As-designed view* (engineering bill of materials). The relationships among individual components reflect a logical organization of the data (primarily geometry) as modified to satisfy engineering improvements for aerodynamics, structural strength, weight, and so on.
- *As-ordered view* (manufacturing bill-of-materials). The relationships among individual components reflect the parts that must be ordered and the specific attributes (including geometric definition) needed to successfully acquire the parts. Concepts like alternate suppliers are important in this view.
- *As-delivered view*. This view contains information about the physical configuration turned over to a customer after the assembly process is complete. Some components might have been replaced (for example, a supplier changed) or modified during assembly.
- *As-owned view*. The actual components in a final product change over time. Components get routinely replaced or repaired as maintenance occurs.

Each of these views is essential during a product's life cycle and each wants to be master. Providing each view without negatively impacting the integrity of the underlying data poses a difficult challenge.



Courtesy Boeing

6 **Communicating data: (a) solid shaded image, (b) engineering drawing, and (c) stylized maintenance image.**

In terms of sheer storage space, the amount of data needed to faithfully represent a product and to show how decisions were reached in its design is expanding by decimal orders of magnitude. The data ranges from marketing intelligence about customers to versions of complex geometry to multiple data sets from engineering analysis runs to manufacturing planning scripts to sensors monitoring manufacturing processes, physical testing, and real-time product health.

In addition to managing multiple configurations and views, people face a challenge when they have to interpret this vast quantity of information. For example, technology is on the horizon that will allow real-time display[11,12] of an entire commercial airplane model. Humans have a finite ability to comprehend complexity, and understanding how to best display the data in a meaningful way requires extensive work.

Challenge 8: Appropriate designs for other uses and users

This challenge focuses on the large number of people who are removed from the data creation and analysis process and rely on the digital product definition for their own job tasks. These people participate in sales, fabrication, assembly, certification, maintenance, and product operation. In the case of a commercial airplane, the user community extends far beyond company boundaries. For example, airline personnel buy Boeing products, perform maintenance, order spares, train pilots and attendants, and so on. Partners design and analyze significant airplane components. Suppliers formulate proposals to fabricate parts and assemble components. Government agencies certify that the final product performs according to published regulations and guidelines.

Many of these groups have their own graphic vocabulary because different information must be communicated to different audiences. Consider the following examples.

Engineers and designers commonly use images like Figure 6a. The colors are meaningful in terms of the engineering and design functions of individual components, but not for manufacturing or maintenance. This is a significant change from practice prior to 1990, where the master communication mechanism was the engineering drawing (see Figure 6b). The authority image for certification, manufacturing, and maintenance is still the engineering drawing. Communicating via the engineering drawing to a maintenance engineer causes a radical transformation, as Figure 6c shows.

Figures 6b and 6c are clear illustrations of this challenge. The two represent exactly the same 3D geometry. However, the graphic styles are dramatically different. Tools are commonly available to work with the base 3D model—for example, make everything in a scene translucent but objects of interest; show and hide; rotate, scale, and translate; and measure. We can generate the basic 3D hidden line image for Figure 6c, do some early guesses to explode parts, and so on.

These tools fundamentally change the basic style in which the graphic image is rendered, and some might even produce engineering drawings. None contains an ability to adapt an image for specific uses or users. Therefore, the challenge lies in knowing how to draw the image to communicate to a specific user community because the communication techniques and standards vary significantly from community to community and from company to company.

As a result, companies spend significant amounts of time and labor retouching or redrawing images. Automation also offers the possibility of providing new types of images that might be more task appropriate.

Generating the initial image itself presents a significant challenge. The taxonomy of effective graphic images (Figure 6 represents a small subset) is much larger. Massironi[13] has developed an excellent taxonomy that delineates the types of images and the fundamental graphic techniques people have used to communicate visually.

Challenge 9: Retrieving data years later

Today's product specifications for tolerance, fit, reliability, and so on, are greatly different than they were 40 years ago. For example, the Boeing 707 successfully introduced commercial aviation to the jet age. Yet the 707's part fit was loose enough that it received the nickname "the flying shim." On the other hand, the first Boeing 777 fit together so precisely (largely due to the use of CAD/CAM techniques from 10 years ago) that the number of discrepancies needing redesign was substantially less than what had appeared to be an extremely optimistic early prediction. Rather than the multiple mock-ups needed for previous models, the 777 manufacturing mock-up flew as part of the flight certification process. Similar stories exist in the automotive and other industries.

One constant remains: The engineering drawing serves as the design, manufacturing, and certification authority. While we realize that the engineering drawing has its limits, it has another important attribute: It can be archived for long periods of time and still be understood. The Mylar film that Boeing uses for its permanent records lasts longer than the 50-year life span of commercial airplanes.

In contrast, consider the challenge if the archives were stored digitally. The media itself is not the problem because it can be routinely copied to new media. The real challenge is changing software versions. Users routinely expect that some percentage of their data will not migrate successfully from one version to the next. Some algorithms will be tweaked, and what worked in the last version doesn't in the next (or vice versa). A substantial version change or choosing a different software supplier means massive amounts of data conversion and rework. The net result is that companies cannot afford to change to a new system version from the same vendor until years after initial release, let alone change to a new vendor entirely. As the amount of configuration-managed data increases, the data migration challenge becomes even more pronounced.

Challenge 10: Limited by the speed of light

Challenge 5 discussed the impact of users being released from the constraints of single-user desktop and laptop computers into an environment where workspaces can be shared. This challenge addresses the users working in a geographically distributed environment, where walking across the hall becomes impossible.

A continued debate rages about the internationalization of large and small businesses throughout the world. In spite of specific national interests and boundaries, companies (many of which are small and mid-sized) are acquiring design, manufacturing, assembly, maintenance, and customer support service help throughout the world. The incentives range from specific technical skills to trade offsets to cheaper labor.

Doing effective distance collaboration for 3D design, which lacks the immediacy and extensive cues of a face-to-face session, is a long-term research and cultural challenge. The key productivity aspect for improved design cycle time is collaboration across a number of different stakeholders. There are a wide variety of collaboration models and a reasonable amount of research done in computer-supported collaborative work. Researchers are starting to pay attention to collaborative task analysis.[14] However, research on how to make such efforts more effective on a global scale is in its infancy, and the cognitive effects of distance on a collaborative 3D work environment have not been addressed.

Summary

This final set of challenges indicates that the way groups of people work on design problems and the longevity of the electronic versions of their products is undergoing a dramatic change. The magnitude of the challenges these changes are causing is at least as great as getting people to adopt CAD in the first place because the challenges impact the fundamental way people work.

How to proceed: Spurn the incremental

We conclude by suggesting an approach that can address our 10 challenges. The basic CAD business model is incremental refinement, and it can only take us so far.

Incremental refinement is based on bringing out the $n + 1$st version, where the new version release has improvements. For this model to work, the incremental improvement for release$_n$, ΔI_n, must be greater than some threshold value, VT, which represents the minimum improvement that will still motivate a company to purchase the new version.

The cost of achieving that degree of improvement increases with each release and can be approximated by

$$\Delta I_n \geq VT \Rightarrow \$\Delta I_n = \$\Delta I_2^{On}$$

In other words, to keep the incremental value of release$_n$ high enough to motivate purchases, the cost of those improvements is on the order of the cost of those of the second release (that is, the first upgrade), raised to the order's nth power.

Two factors contribute to this cost picture. First, as systems go through successive releases, they grow in complexity. This negatively affects their malleability and increases the cost of adding value or refactoring the software or basic functionality. It's not just the number of lines of code or additional features that cause this. As a system approaches maturity, the legacy of the initial underlying architecture, technologies, and paradigms creates a straightjacket that severely affects the cost of change.

Next, as products mature, the software more or less works as intended and markets approach saturation. For most users, the current version of the software is good enough. Changes increasingly tend toward tweaks and tuning rather than major improvements. Or they are directed at more specialized functions needed by a smaller segment of the market. Consequently, there is less to motivate most customers to upgrade.

As the product reaches late maturity, the accumulated impact is that development costs increase while the size of the addressable market decreases. Software sales are no longer sufficient to cover the growing development costs. At this point, companies increasingly rely on annual support contracts, a model difficult to sustain. At best, the switch to support revenue delays the collision of technology costs and economic viability. Even when a company totally reimplements its product for a new version, the tasks the user performs are generally the same as in the previous version and often give less functionality than the last version of the old system.

The CAD industry is reaching this state. How do we get to the next level, the one that should address these challenges?

Order-of-magnitude approach

If incremental refinement won't work, perhaps it will come from some new breakthrough. Perhaps some new invention will magically appear and save the day. The National Academy of Science recently released a report that studied the genesis of a large number of technologies, including graphical user interfaces, portable communication, relational databases, and so on. For each, it looked at how long that technology took to get from discovery to a \$1 billion industry. The average was about 20 years.[15] This suggests that any technologies that are going to have impact over the next 10 years are already known and have probably been known for at least 10 years.

We believe that the answer is not new technologies but new insights into technologies that are already known, whose potential is perhaps not appreciated, or which have so far not been technically or economically feasible. The engineering of significantly better products should come from fresh insights on what is already known or knowable.

By analogy, think about the marks on the door frame of the kitchen closet where parents record the growth of their children. Without that documentation, especially because you live with them day-to-day, you generally don't notice the changes. Long after it has actually happened, by measuring them again, you notice that they have passed on to the next stage of growth. This also happens in living with technology.

Our observation leads to the order-of-magnitude (OOM) rule, which says: If anything changes by an order of magnitude along any dimension, it is no longer the same thing.

OOM is a way to measure significance of the types of changes needed to meet the technical challenges we've described. Exploiting this rule forces us to notice OOM changes and understand their implications. However, it also lies in teasing out less obvious, but meaningful dimensions, along which to test for such OOM changes.

Because we have all been so intimately involved with these challenges and CAD technology, we have not taken the time to create the measures that allow us to see where profound changes have already occurred and where they need to occur. We believe that the approaches needed to address the 10 challenges are most likely to result from rethinking things from a human-centric rather than a technology-centric perspective.

Conclusion

The next wave in CAD will come about largely through the cumulative effect of the introduction of a number of small, lightweight technologies that collectively form a synergistic mosaic, rather than due to the introduction of some monolithic new technology. The value of this approach is that it can be introduced incrementally, without requiring some disruptive discontinuity in skills and production. Thus, change will occur through radical evolution. That is, incremental evolutionary change will happen in a way that leads us to a radically new approach to working.

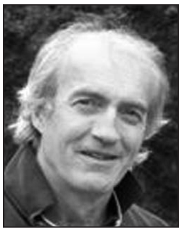
There will be changes in how we do things, and these will significantly affect the nature of CAD systems and how they function. We suggest that using OOM offers a strategy to cause revolution in an evolutionary manner. Ultimate success will happen if and only if all of the technical areas in which OOM changes occur are kept in balance. We conclude that institutionalizing radical evolution is the top CAD challenge we face in 2005, 2015, and beyond. ■

References

1. I. Sutherland, R. Sproull, and R. Schumacker, "A Characterization of Ten Hidden-surface Algorithms," *ACM Computing Surveys*, vol. 6, no. 1, Mar. 1974, pp. 1-55.
2. R.A. Liming, *Practical Analytic Geometry with Applications to Aircraft*, Macmillan, 1944.
3. P. Garrison, "How the P-51 Mustang Was Born," *Air & Space Magazine*, Aug./Sept. 1996.
4. D.R. Ferguson, P.D. Frank, and A.K. Jones, "Surface Shape Control Using Constrained Optimization on the B-spline Representation," *Computer Aided Geometric Design*, vol. 5, no. 2, 1988, pp. 87-103.
5. R.T. Farouki,"Closing the Gap between CAD Model and Downstream Application," *SIAM News*, vol. 32, no. 5, 1999, pp. 303-319.
6. Research Triangle Inst., *Interoperability Cost Analysis of the US Automotive Supply Chain*, Nat'l Inst. of Science & Technology, March 1999; http://www.mel.nist.gov/msid/sima/interop_costs.pdf.
7. C.M. Hoffmann, *Geometric and Solid Modeling: An Introduction*, Morgan Kaufmann, 1989.
8. D.R. Ferguson et al., "Algorithmic Tolerances and Semantics in Data Exchange," *Proc. 13th ACM Symp. Computational Geometry*, ACM Press, 1997, pp. 403-405.
9. V. Shapiro, *Limitations in Current CAD Systems*; http://sal-cnc.me.wisc.edu/Research/parametric/limits.html.
10. F. Brooks, *The Mythical Man-Month: Essays on Software Engineering*, Addison-Wesley, 1975.
11. W. Baxter et al., "GigaWalk: Interactive Walkthrough of Complex Environments," *Proc. Eurographics Workshop on Rendering*, ACM Int'l Conf. Proceeding Series, 2002, pp. 203-214.
12. I. Wald, C. Benthin, and P. Slusallek, "Distributed Interactive Ray Tracing of Dynamic Scenes," *Proc. IEEE Symp. Parallel and Large-Data Visualization and Graphics* (PVG), IEEE Press, 2003.
13. M. Massironi, *The Psychology of Graphic Images: Seeing, Drawing, Communicating*, Lawrence Erlbaum Assoc., 2002.
14. D. Pinelle, C. Gutwin, and S. Greenberg, "Task Analysis for Groupware Usability Evaluation: Modeling Shared-Workspace Tasks with Mechanics of Collaboration," *ACM Trans. Computer–Human Interaction*, vol. 10, no. 4, 2003, pp. 281-311.
15. Computer Science and Telecommunications Board, "Computer Science and Telecommunications Board of the National Research Council," *Innovation in Information Technology*, National Academies Press, 2003.

***David J. Kasik** is an architect for software engineering, geometry and visualization, and user-interface technology at the Boeing Commercial Airplanes Group Information Systems. His research interests include an innovative combination of basic technologies and increasing awareness of the impact of computing innovation outside the computing community. Kasik has a BA in quantitative studies from the Johns Hopkins University and an MS in computer science from the University of Colorado. He is a member of*

the ACM, ACM Siggraph, and ACM SIGCHI and received a Siggraph Special Recognition Award in 1992 for acting as Liaison to the Exhibitor Advisory Committee. Contact him at David.j.kasik@boeing.com.



William Buxton *is principal of his own boutique design and consulting firm, Buxton Design, where his time is split between working for clients, writing, and lecturing. He is also a chief scientist for Bruce Mau Design of Toronto, and he is an associate professor in the Department of Computer Science at the University of Toronto. Buxton has a B.Mus. from Queens University and an MSc in computer science from the University of Toronto. He received the 1995 Canadian Human–Computer Communications Society Award, the 2000 New Media Visionary of the Year Award, and is a member of the CHI Academy. Contact him at bill@billbuxton.com.*

David R. Ferguson *was Boeing's leader in geometry research and development until his retirement. His research interests include the mathematics of curves and surfaces with special emphasis on nonlinear methods for geometric construction. Ferguson has a BS in mathematics from Seattle University, and an MS and PhD in mathematics from the University of Wisconsin. He is an editor of* Computer Aided Geometric Design. *Contact him at DRF.ASSOC@comcast.net.*

For further information on this or any other computing topic, please visit our Digital Library at http://www.computer.org/publications/dlib.

2005 Editorial Calendar

January/February: Emerging Technologies*

This issue covers the Siggraph 2004 Emerging Technologies exhibit, where the graphics community demonstrates innovative approaches to interactivity in robotics, graphics, music, audio, displays, haptics, sensors, gaming, the Web, AI, visualization, collaborative environments, and entertainment.

*Bonus CD-ROM of demos included with this issue.

March/April: State-of-the-Art Graphics

This issue covers an array state-of-the-art computer graphics, including new developments in VR, visualization, and novel applications. The broad range of topics highlights the usefulness of computer graphics.

May/June: Smart Depiction for Visual Communication

Smart depiction systems are computer algorithms and interfaces that embody principles and techniques from graphic design, visual art, perceptual psychology, and cognitive science. This special issue presents such systems that hold the potential for significantly reducing the time and effort required to generate rich and effective visual content.

July/August: Applications of Large Displays

The emergence of large displays holds the promise of basking us in rich and dynamic visual landscapes of information, art, and entertainment. How will our viewing and interaction experiences change when large displays are introduced in our workplace, home, and commercial settings? This special issue will serve to collect and focus the efforts of researchers and practitioners on the frontier of designing large displays.

September/October: Computer Graphics in Education

Graphics educators are cultivating the next generation of developers. However, hardware and software barriers to entry have shrunk, and people from nongraphics areas have begun adopting the technology. This special issue will highlight approaches from inside computer graphics education and uses from outside the field in domain-specific education.

November/December: Moving Mixed Reality into the Real World

As computing and sensing technologies become faster, smaller, and less expensive, researchers and designers are applying mixed reality technology to real problems in real environments, This special issue will present a broad range of issues faced by designers as they move state-of-the-art technology beyond the laboratory.

IEEE Computer Graphics and Applications