FROM SKETCHPAD TO CITY OF BITS

A Story of Shifting Intentions

WILLIAM J. MITCHELL
Massachusetts Institute of Technology
wjm@MIT.EDU

Abstract. By my count this year marks the fiftieth birthday of the field of computer-aided design. It is, therefore, an appropriate moment to look back on how the field has developed in its first half century and then to consider what we might expect in the coming decades – the decades that will see the continued development of what I have called the City of Bits. The story is, as we shall see, one of shifting intentions; at each stage in the evolution of computer-aided design, the technology has found niches in practice determined both by its capabilities at that particular historical moment and by prevailing economic and cultural imperatives.

1. CAD and Modernism

As far as I can tell, the emergence of numerically controlled milling machines, in the 1950s, first motivated interest in the possibility of computer-aided design systems. NC machines required the programming of tool paths for the objects to be milled. Direct programming of these paths was a tedious and error-prone process, so there was interest in efficient systems for specifying and editing three-dimensional shapes. The first CAD software was primarily intended for this task.

By the mid-1950s, a broader view of CAD’s possibilities was developing. A speculative article in Fortune magazine provided a comprehensive and very prescient summary. An elaborate illustration showed an interactive graphics console for input and editing of object shapes, storage facilities for digital models, and a CAD/CAM device for output of physical prototypes.

By the early 1960s, at MIT’s Lincoln Laboratories, Ivan Sutherland had translated the key ideas into a working demonstration – the famous Sketchpad system. This implemented the essentials of a graphic editing system and the
rudiments of parametric design – an idea that was not to come to practical fruition until it was rediscovered decades later. Tim Johnson quickly followed Sketchpad with Sketchpad 3 – a three-dimensional version – and began to explore potential architectural applications.

At Cambridge University – inspired by Sketchpad – William Newman (who was later, with Robert Sproull, to produce the first comprehensive textbook on computer graphics) implemented a pioneering computer-aided architectural design system, and published a path breaking paper on it in the Computer Journal. Newman made the intellectual connection between the pick-and-place operations of Sketchpad and the step-by-step assembly of building designs from the standardized elements of industrialized component building systems. It shifted the focus of CAD from design of parts to design of large, complex assemblies, and it was hugely influential.

Newman’s system also cast CAD as a tool of mid-century modernism. The modernists – particularly in contexts of postwar reconstruction – had established a characteristic strategy of standardization and repetition as a means of building quickly, inexpensively, and at large scale. In other words, they traded off architectural complexity in order to deal with problems of scale. Newman’s system demonstrated that this strategy could be extended back from a rationalized construction process to a computer-supported design process. Many of the early, practical CAD systems of the 1960s and the 1970s were grounded directly upon this idea.

However, industrialized component building was not popular everywhere – and, in particular, it never gained much favor in the United States. But the idea of pick-and-place assembly could be extended to the elements of Euclid’s ancient geometric constructions – points, straight lines, arcs, and circles. Indeed, it was soon realized that CAD editing operations could serve as exact analogues of traditional drafting operations executed with parallel bars, triangles, dividers, and compasses.

CAD drafting systems made a broad and long-lived connection to practice, since architectural classicism, modernism, and historicizing postmodernism all generally operated within the framework of Euclid’s geometry. They ushered in an era in which architectural CAD systems were seen primarily as more efficient substitutes for hand drafting of construction documents within otherwise very traditional design and documentation processes. (In their essential features, these processes had been established in the nineteenth century.) It was the era in which AutoCAD rose to market dominance.

2. CAD and 21st Century Industry

Meanwhile, the technologies of three-dimensional geometric modeling and computer graphics continued to develop. Throughout the 1970s, 1980s, and
1990s, there were enormous advances in curved surface modeling capabilities
and tools for modeling solids and assemblies of solids. Most of the motivation
and research and development funding for this work came from outside of
architecture, but some architectural researchers and practitioners kept a close
eye on these developments and there was an ongoing discourse in academia –
paralleling the more commercial strand of CAD in practice – that explored
the potentially radical uses of these technologies in design.

A breakthrough came in the early 1990s, when Frank Gehry and the
technical team in the Gehry office began seriously to explore the use of these
technologies on large, complex projects that did not fit the paradigm of either
industrialized component building or Euclidean geometric discipline. The goal,
here, was different – to support efficient design and construction of buildings
with curved surfaces and generally complex, non-repetitive forms. Quite
understandably, then, no architectural CAD system met Gehry’s needs, and he
went to Catia – a system that had been developed primarily for use in
manufacturing industry, and that offered a high degree of integration of design
and CAD/CAM fabrication capabilities. In some ways, this represented a
return to CAD’s earliest roots in numerically controlled machining technology.

In projects like the Barcelona Fish, the Bilbao Guggenheim, the Disney
Concert Hall, and the Stata Center at MIT, this enabled fabrication and on-
site assembly, within acceptable boundaries of time and cost, of highly non-
repetitive buildings. Gehry demonstrated that it was no longer necessary to
make the modernist bargain of trading off complexity for scale. A generation
of younger architects eagerly began to explore the implications of this
fundamental shift.

Ironically, though, there was a price to pay at the design stage. Complex,
non-repetitive buildings entail more explicit decisions per square meter than
repetitive buildings. Furthermore, detailed three-dimensional modeling of
buildings is generally more time consuming and costly than two-dimensional
drafting of plans, elevations, and sections. And by showing more, they can
also transfer more liability to the architect – which shows up in greater design
effort, more comprehensive checking processes, and higher insurance rates.
Architects must justify their correspondingly higher fees (as Gehry generally
succeeds in doing) in terms of savings at the construction stage and ultimate
benefits to building occupants and owners.

It is possible to reduce the price at the design stage by introducing
parametric and rule-based generative design software that treats varied
conditions as instances of a general type. The software reduces explicit design
decision-making by automatically generating instances appropriate to
particular conditions and producing corresponding CAD/CAM production
information. This is the strategy successfully followed, for example, in
Foster’s design for the non-repetitive roof structure for the British Museum
courtyard. In general, we can probably expect to see a growing demand for
customized variation in buildings (and other products of 21st-century industry) being met by rule-based software allied with CAD/CAM production. In the early 2000s, the forty-year-old, long-neglected idea of parametrics finally seemed to make practical sense.

3. CAD, Communication, and Collaboration

As the Internet era unfolded, there was a growing realization that the geometric model and material specification of a building – traditionally thought of as the essence of “the design” – constituted only a fraction of the information required to get a project designed and built. This motivated attempts to digitally capture, rationalize, and manage the entire information environment surrounding a project – or, at least, a much larger proportion of it than in the past – through use of OCPM (Online Collaboration and Project Management) systems and similar software. This promised particular advantages in the increasingly common context of multi-player, geographically distributed, asynchronously operating design and construction teams.

In an OCPM system, the body of stored information grows and evolves throughout the life of the project. “The design” is thus expressed as a large, open-ended collection of decisions, agreements, and answers to queries recorded over time. Some of this information takes the form of geometric models and drawings (often many of them, maintained in different locations), some the form of written specifications, some that of responses to requests for information (RFIs), some that of spreadsheets and budget documents, some that of informal verbal responses and sketches, some that of contracts and purchase orders, and so on.

The viewpoint represented by OCPM systems sits uncomfortably with those who would like to see designs as closed and complete intellectual products, like musical scores, which are to be executed (or performed) as faithfully as possible. (This is also a perspective that privileges the role of the individual “designer” who has personal authorship.) But it resonates with those who – informed by poststructuralist literary and cultural theorists – tend to think of intellectual products as inherently incomplete and imperfect, continually contested, negotiated, and reinterpreted, and therefore constantly evolving outcomes of social processes.

OCPM systems are usually justified on very pragmatic grounds of providing efficient distribution of current information, providing trails for use in dispute resolution, and the like. But their ultimate importance may be more fundamental – deriving from their introduction of a much broader, more realistic conception of design activities and products. Designing does not
reduce to editing geometries and selecting products – the functions that early CAD systems happened to support.

4. CAD and the Design of Intelligent Environments

Currently, the use of CAD systems in architecture typically amounts to the application of twenty-first-century tools to the design of nineteenth-century objects. Buildings are still big, dumb things that consist mostly of skeleton and skin, plus relatively unsophisticated mechanical and electrical systems. But this situation is now changing as buildings acquire wired and wireless digital nervous systems and elements of embedded intelligence ranging from simple RFID tags to sophisticated sensors, processors, and communication devices.

We are gradually entering the era of the Internet of things, in which hitherto dumb objects – from light bulbs to closet doors – have IP addresses and local intelligence, and function as elements in digitally coordinated systems that serve their functions effectively through awareness and intelligent response. Architects need to begin to think of buildings and their components – like computers – as digitally programmable entities. In the City of Bits, software-defined behavior is a fundamental design attribute.

In recognition of this, CAD tools will eventually need to be extended to allow not only specification of geometry and association of material and cost properties with design elements and subsystems, but also the specification and association of intelligent behaviors. Simulation tools will need not only to render visual appearance, evaluate thermal performance, and so on – but also to “run” and debug the design under various scenarios of external conditions and user demand – much as in animation, video game, and robotics design tools.

Some of today’s more advanced architectural design tools do make use of intelligent objects that respond and evolve to changing contexts as a design develops, but these are typically “frozen” in one state before the design is constructed. In the future, through the use of embedded intelligence, this dynamic response and evolution will continue after construction.