Now More Than Ever: computational thinking and a science of design

Mark D Gross
Carnegie Mellon University
mdgross@cmu.edu

Abstract

Revolutions in desktop manufacturing and embedded computing are changing the way we make things. These changes will enable citizens to engineer and manufacture their own goods. The role of the designer is also changing, deciding on the manufacture of specific artifacts, to setting the bounds and rules for decisions that end-users make. Materials are also changing, and programmable matter made of ensembles of modular robots demands new and dynamic ways of describing designs. A science of design is an essential element for this, and it is likely that a science of design will be expressed computationally.

Keywords

Computational thinking; programmable matter; end-user design

1. Introduction

Forty years have passed since Herb Simon wrote his influential The Sciences of the Artificial in which he coined the term, the Science of Design [1]. At that time, in the late 1960s, people had a growing sense that the world we make and live in was growing so complex that the traditional ways of designing were no longer adequate to the task. If it was true then, it is truer today.

Yet for the past several decades, design research has been viewed with some skepticism. Many look down on design research, believing that (in the words of Alexander's 1971 preface to his Notes on the Synthesis of Form), "People who study design methods without also practicing them are almost always frustrated designers who have no sap in them, who have lost, or never had, the urge to shape things. Such a person will never be able to say anything sensible about "how" to shape things either" [2]. Within the engineering community, too, the idea that there might be a "science of design" has met with some skepticism. Many excellent engineers believe that research on design is a 'soft' field of study. They believe that engineering design is driven by the properties and behaviors of specific domains. They think that other than a need for "proper thinking" there is little interesting that we can say in general about design.

With due respect to these doubts, in light of the revolutions in manufacturing and technology that we are now experiencing, we can no longer afford to view a science of design as soft or as an irrelevant intellectual game. Rather, a science of design is a necessary foundation for the changes that are already beginning to pervade our everyday lives.

First let us clearly draw a line between the "science of design" and the "science of designers". Both (but especially the latter) have been the subjects of a great deal of research over the past several decades. The science of design is the study of design processes, regardless of who, or what, is doing the design. For example, researchers may investigate how a space of designs can be efficiently searched, or how a notation or language can compactly express a class of designs. The science of designers is the study of human designers, how they think, what they do, and how they communicate. For example, researchers analyze designers' drawings, ask designers to think aloud as they design, and videotape designers working together in groups [3]. Both fields of study are interesting, but they are quite different enterprises.

2. Radical changes in how we make things

Today we are in the early stages of a profound change in the way we design and construct our physical world. It is not the first time this has happened. Christopher Alexander tells this story in Notes on the Synthesis of Form [2]. At the beginning of the industrial age in the late 18th century our society moved from individual craftsmen making artifacts one by one, to assembling (by hand) objects from standard components, and thence to mass production in the 20th century [4]. With each shift in production has come a corresponding shift in designing. In the age of craft, design was implicit—based on a shared understanding of common goods. When you needed a new gate or a hammer you went to the blacksmith and explained what you needed and he made you one. The industrial age brought the need for explicit designing to consider the function of the artifact and to plan the materials and methods of producing it. The designer or engineer made paper drawings and models to plan the artifact. When the designing was done the drawings were used for manufacture. Mass production made design even more necessary, as high costs of tooling and setting up manufacturing lines demanded that we thoroughly think through an artifact before beginning to make it. Today, designers no longer use paper drawings to conceive, consider, and convey their designs. Instead, files are stored and transferred electronically from designer to manufacturer. Still we are mostly in the mode of making drawings and models to design the artifacts we desire.

Each shift in the design and production of our physical environment has resulted in broad and profound impacts on our society in myriad ways (health, education, social and economic order) that would have been almost impossible to predict. There is every reason to believe that the changes of our time will have even broader societal impact than those that have come before. The fundamental change in manufacturing and production that we are in the midst of now has the capacity to enable and empower ordinary citizens in ways that have never before been possible. By leveraging the science of design, the engineering of desktop manufacturing, and the software to bring the two together, we can make the vision of “democratizing innovation” [5] come true. I will argue that it is the software, and particularly computational ways of expressing design knowledge and expertise that will bring this dream to reality.

The shift in the production of the physical world derives not from a single technological advance but from developments in several arenas: computer controlled tools (e.g., desktop manufacturing), embedded computing, and science of design.

3. Desktop design & manufacturing revolution

Already underway is the shift to desktop manufacturing — people can afford to design and manufacture one-off artifacts for themselves. An early example was desktop publishing. Before we invented laser printers if you wanted a brochure, a newsletter, or a poster you worked with a graphic artist to design a layout, select typefaces, paper, and so on; and then with a printer who would execute the design and produce the final inked paper product. Today laser and inkjet printers are practically free, and using desktop software anyone can design and print their own newsletters, calendars, wedding invitations, and even books. The desktop publishing revolution was driven first by the development of laser printing technology. Application software enabled professionals at first, and eventually end users, to produce graphic work. A key component was the underlying Postscript language that applications use to produce page descriptions for laser printers.

Laser and water jet cutters, three-dimensional printers, and computer-numerically controlled milling machinery are now extending this shift from the mostly flat world of paper and graphic arts to the richer three dimensional world of physical objects. The first (or second) generation of hardware to support this revolution in “desktop manufacturing” is already commercialized and capabilities continue to advance as costs drop [6, 7]. The specific technologies vary from laser sintering to fused deposition modeling, but we are clearly moving along a trajectory from single material (e.g., plastic or metal) to multiple materials, to the ability to manufacture—in small quantities—unique physical objects with embedded electronic circuitry, printed displays and other actuators (http://fabathome.org; http://www.2objet.com). As with desktop publishing, software plays a key role: Computer-aided design and engineering applications to describe physical objects and simulate their behavior, and the underlying representations (analogous to PostScript) enable designers to do their work.

4. Embedded computing revolution

A second revolution —this one in computing—is also underway: We are embedding microcontrollers, actuators, and sensors into our physical environment, and the communication and control of these devices [9]. Advances in micro- (and nano-) electronics leading to low cost sensors and actuators, micro-controllers that are as powerful as yesterday's mainframes, and new wireless communications protocols fuel this revolution. We see its impact in everyday lives as our clothing, our furniture, our buildings, automobiles, and cities become computationally enhanced. Applications are simple so far but already we have the capacity to make things that exhibit computationally complex behaviors.

The desktop manufacturing revolution applies here as well. As recently as a decade ago only an experienced engineer could design and manufacture a printed circuit board (PCB). Today even a high school student can easily acquire the skills to design a board using off-the-shelf software and send the file to a fabricator for low-cost overnight manufacture. Inkjet printer companies are now envisioning affordable desktop PCB manufacture [8].

5. Design methods & science of design

In addition to advances in desktop manufacturing and ubiquitous computing, another relatively recent development is relevant: The recognition that the complexity of the things we make and their interaction in the world demands that we understand designing better. Although its roots go back further, in the mid-1960s researchers—in what became known as "design methods movement"—began to recognize that increasing complexity demanded a comprehensive understanding of designing—in Simon's memorable phrase, a "science of design" [1]. Although the focused intensity of the design methods movement faded, the agenda did not. Today the software engineering and human-computer interaction communities have embraced Alexander's "Pattern Language" approach [9]. Horst Rittel's "issue based information systems" [10] led to design rationale and knowledge management.

The recent US National Science Foundation "Science of Design: Software Intensive Systems" initiative [11] is further evidence that the needs that drove Simon, Rittel and others in the 1960s and 1970s—to understand designing in the face of increasing complexity—remain relevant today. We still lack a coherent fundamental science of design (an understanding of the structure of design decision making, abstracted from specific domains). Still, we have seen steady progress in modeling design processes and developing computational design methods and tools. As computer hardware advanced and more powerful programming environments became the norm, the early insights of the design methods movement took form in increasingly powerful computer-aided design (CAD) tools for architectural, mechanical, electrical, civil, and software engineering.

6. Code as the carrier for design expertise

The move, starting in the 1960s, from design by hand to design with computer tools enabled us to begin to automate some of the reasoning and decision making that is at the heart of designing. One of the earliest examples, of course, was Ivan Sutherland's Sketchpad program. Sketchpad is known for many things, but for the science of design, Sutherland's most important contribution in Sketchpad was to describe a design as a set of constraints that the program could manage as the human designer made changes. Later, during the 1980s and 1990s, researchers in expert systems, case based reasoning, and other fields of artificial intelligence, followed this general approach and applied these ideas to design in many different domains—from buildings to circuits to software to machines. Advances in computer hardware and software during the 1980s and 1990s made it possible to implement the ideas that the design methods researchers had worked on in the 1960s and 1970s.

What is important about this piece of history is that software became the medium for carrying the methods and techniques that the early design researchers developed. In a kind of chicken-and-egg process, as the software became more sophisticated, designers in practice began to adopt it and depend on it. In some fields, notably integrated circuit design, the software began to embed automated design methods that human designers could not perform in reasonable amounts of time. Design knowledge and expertise began to take the form of code. Designers began to adopt computational thinking [12].

Still, during the shift from design-by-hand to computer-aided design, the dominant model has been the computer program as tool or assistant to the designer. The designer is in control and makes all the decisions. The computer has served mostly to record and display the decisions the designer makes, and to calculate, look up, and render information about the design. Although adopting computer aided design tools has affected design practice, so far we have experienced only a small departure from the traditional way of making design decisions. That is about to change.

7. End-user design and computational thinking

The revolutions in desktop manufacturing and in embedded computing push us towards computational ways of thinking about design. One example is end-user designing, now becoming popular as 'co-creation' [13]. End-user design is the idea that as we move away from mass-production and embrace the idea of individualized or 'mass-customized' manufacturing, ordinary citizens will be able to design and make things for themselves. We are seeing the first wave of co-creation, in which citizens (sometimes called "consumers") participate in making decisions about a design. The examples are many, from shoes to cars to toys. (Although it is now becoming popular, enabling end users to directly make design decisions is an old idea: Beginning in the early 1960s Dutch design methodologist N. John Habraken developed a theory and method for engaging citizens in the design of their housing [14].) End-user design requires professional designers to set up a design space that citizens can work within. They specify the rules that govern the end-user designs. (This too, of course, is a design act.) Today the design and production process is usually computationally mediated, so the bounds of the space and the rules that govern designs are also expressed computationally.

The advances in personal desktop manufacturing that are empowering end-users to design and manufacture their own goods demand advances in software. We need representations to describe designs and applications to manage and manipulate those representations. The representations are design languages that machines can parse, recognize, and process. The applications are compilers and other development tools. Instructions in a high-level programming language like Ruby or Lisp describe the behavior we want a computer to perform. Instructions in a high-level design language describe what we want of our design artifact. A design compiler takes high-level descriptions of the behavior and generates implementation in the form of an object. For example, a compiler might generate code that a 3D printer, or other desktop manufacturing machine can execute to physically produce a design.

It might seem that this way of designing will limit creativity. The opposite is true. Computational descriptions of design will enrich, not impoverish opportunities for everyday creativity. It should be clear, then, that the way that the computational tools for design are configured will strongly color the ways in which citizens can be creative. Nakakoji has outlined an interesting and valuable framework for understanding—and designing—computational tools to support end-user creativity in design [15].

8. The programmable world

Another, perhaps even more profound, change is on the horizon: a physical world whose behavior that we can program. We already see microprocessors embedded into many of our everyday things—from clothing to transportation—and with that comes the ability to program their behavior. As our things and our world become enhanced with computation, we must find ways for citizens to program and reprogram their behavior. As with our end-user design story, citizens become designers of the dynamic behavior of things and places in the world.

A logical extension of the computationally embedded things we have today is a world built from ensembles of thousands of modular robots. Each robot would be able to sense its immediate environment, move itself and perhaps its robot neighbors, and communicate with other robots in the ensemble. The robots could be programmed to respond automatically to changes in their environment, or to change configurations on command. For example, a building made of robot building blocks [16] could reconfigure itself to adapt to different weather conditions, different uses, or to respond to emergencies such as earthquakes, fires, or floods. Although making this idea a reality may seem far in the future, several research groups are developing the core technologies for "programmable matter" today [17-19].

If programmable matter becomes an everyday reality, how will we design for it? As we saw with end-user design, the role of the professional designer will change. No longer will the job of a designer be to make informed decisions about a specific artifact. Instead, the job of the professional designer will be to program the artifact's dynamic and responsive behavior. Or rather, to program the dynamic and responsive behavior of the ensemble of modular robots of which the artifact is made. To the designers of today, this may seem a quite different kind of job than what we usually think of as design. Really, though, the designer's task will still be—as it always has been—to create things that meet certain needs. The difference is that instead of creating the things directly in a "one-off" fashion, the designer will program the materials to respond to different conditions.

9. Discussion

We began with a reference to Simon's lecture on the Science of Design. Simon made his remarks at a time of great social and technological change around the world. We are today again at a time of great change: enormous challenges face humanity—climate change and its effects, the need to feed a growing world population, mass urbanization, and so on. More than ever we need a science of design—a rigorous and systematic understanding of how to design.

A science of design promises to be domain-agnostic. That is, the idea of a science of design is that, apart from the domain specific expertise of rockets, hearing-aids, anti-retroviral drugs, or public policy, there is also knowledge and expertise in 'how to design' that we can bring to bear on each domain. As we mentioned above, this idea has met with some skepticism in the design, engineering, and scientific communities. And to be honest, so far it has not borne the fruit that we optimistically hoped for in the early days of the field.

I have argued that the way to a science of design —a thorough and systematic understanding of the processes to reach desired outcomes—lies in the approach of computational thinking. Over the past decades, computer-aided design has become widely practiced in every design domain—from architecture, to industrial design, to electronics engineering. The most important contributions of computer-aided design have not been in more realistic renderings or performance simulations. Certainly, these have been valuable. But the real contribution has been to offer computation as a way of conceiving design, as a medium for expressing and exploring design ideas. Computational representations — not only of the form of things, but also of their interactive behavior—are a powerful way to represent designs and design processes. That is why I believe that if we are to have a science of design, it will likely be computationally expressed.

I argued also that the technological changes in our world today are already moving us toward a profoundly computational view of designing. In this world, the designer's role will not be merely to make objects for people, but to describe design spaces and the rules that bound them, in ways that will enable citizens to design their own things, and that will provide "programmable matter" with dynamic and responsive behavior.

References

1. Simon, H. *Sciences of the Artificial*, Cambridge, MA: MIT Press, 1969.
2. Alexander, C. *Notes on the Synthesis of Form*, Cambridge: Harvard University Press, 1966.
3. Suwa, M and Tversky B. What architects and students perceive in their sketches: A protocol analysis. *Design Studies*, 1997(18): p. 385-403.
4. Giedion, S. *Mechanization Takes Command: A Contribution to Anonymous History*: Oxford University Press, 1948.
5. von Hippel, E. *Democratizing Innovation*,

Cambridge,MA: MIT Press, 2005.

6. Arabe, KC. (2004) 3D Printing Makes Its Mark. Industrial Market Trends, http://news.thomasnet.com/IMT/archives/2004/09/3d_printing_mak.html
7. Hansell, S. Beam It Down From the Web, Scotty, *New York Times*. 5/7/2007
8. Epson, I. (2004) Epson Inkjet Technology Used to Fabricate World's First Ultra-Thin Multilayer Circuit Board. 11/1/2004, http://www.epson.co.jp/e/newsroom/news_2004_11_01.htm
9. Alexander, C, Ishikawa S, and Silverstein M. *A pattern language : towns, buildings, construction*, New York: Oxford University Press, 1977.
10. Rittel, W and Kunz W. Issues as elements of information systems. 1970, Center for Planning and Development Research, University of California, Berkeley.
11. Freeman, P and Hart D. A science of design for software-intensive systems. *Commun. ACM*, 2004; **47**(8): p. 19-21.
12. Wing, JM. Computational thinking. *Communications of the ACM* 2006; **49**(3): p. 33-35.
13. Trendwatching. Customer Made. 2006 http://trendwatching.com/trends/CUSTOMER-MADE.htm.
14. Habraken, NJ. *Supports: an Alternative to Mass Housing*, London: The Architectural Press, and New York: Praeger, 1972. (Originally published in Dutch under the title: De Dragers en de Mensen. Amsterdam: Scheltema en Holkema, 1962.), 1962.
15. Nakakoji, K. Meanings of Tools, Support, and Uses for Creative Design Processes. in *Proceedings of the International Design Research Symposium'06*, Seoul 2006; 156-165.
16. Weller, MP and Do EY-L. Architectural Robotics: A new paradigm for the built environment. in *EuropIA.11: 11th International Conference on Design Sciences & Technology* Montreal 2007; 353-362.
17. Goldstein, SC, Campbell JD, and Mowry TC. Programmable Matter. *IEEE Computer*, 2005; **38**(6): p. 99-101.
18. Murata, S, et al. M-TRAN: self-reconfigurable modular robotic system. *Mechatronics, IEEE/ASME Transactions on*, 2002; **7**(4): p. 431 - 441.
19. Yim, M, et al. Distributed Control for 3D Metamorphosis. *Autonomous Robots*, 2001; **10**(1): p. 41-56.