



From Minecraft to Minds

Grady Booch

AS A COMPUTER scientist, I take the position that the mind is computable.

This annoys many people.

Well, perhaps “annoy” is too strong a word. Some reject this point of view on an emotional basis. As sentient beings, we often have a visceral reaction to anything that undermines our own sense of specialness in the universe. Such a reaction is not unique to our age of software-intensive systems. As Jonathan Sawday explores in *Engines of the Imagination: Renaissance Culture and the Rise of the Machine*, “Descartes had denied that reason could ever be produced mechanically”; and even now, “George Dyson has observed [that it] ‘is far settled after three hundred and forty years.’”¹

beautiful argument that hinges on Gödel’s theories, as he describes in *Shadows of the Mind*.² John Searle’s Chinese room thought experiment similarly attempts to disavow any purely mechanistic manifestations of mind.³ And yet, I find no compelling evidence to support Penrose’s assertion that the mind rises as a consequence of quantum processes that we do not—and cannot—understand. The meaning of mind is, to my reckoning, an emergent property of the interaction of billions upon billions of simpler, understandable, and computable systems, a concept that Marvin Minsky examines at length in *Society of Mind*.⁴

To be clear, I am neither a cognitive scientist nor a philosopher of the mind

the golems of Jewish folklore to the *Ars Magna* of Raymond Lull and even more recently to the Matrix trilogy or Sky-net from the Terminator series. On one hand, the possibility of a sentient machine entrances us. Who wouldn’t want an assistant/companion/friend like Andrew from *Bicentennial Man* or Data from the Star Trek universe? On the other hand, this disturbs us at a very deep level. What would such a sentient creation think of us, and would we as a species therefore become irrelevant?

As computing advances and contributes to advancing every element of the human experience, we continue to push back the edges of mystery. That very reality is also disturbing to some, but I ascribe to Feynman’s observation that “it does not do harm to the mystery to know a little more about it.”

Let us then try to unwrap some of that mystery.

We often have a visceral reaction to anything that undermines our own sense of specialness in the universe.

And yet, the rise of scientific thinking and the subsequent realization that many elements of the human body could be understood mechanistically and that the universe could be understood mathematically replaced more magical explanations. The meaning of mind is perhaps the next frontier. Others reject this point of view on cunning theoretical grounds.

Roger Penrose has a particularly

nor an AI researcher. My only qualification in this domain is that I possess a mind (well, at least I think I do, most of the time, perhaps) and that I self-identify an entity that I represent as “me.”

Alan Turing posed the question, “Can machines think?” in his paper “Computing Machinery and Intelligence” in 1950.⁵ Our human fascination with the possibility of thinking machines goes much further back, from

The Computability of the Mind

If one follows the arguments for and against the computability of the mind, one must inevitably confront the meaning of the words “mind” and “computable.” Minds are particularly lousy at understanding themselves, thus what mind is and what it is not will likely always be a slippery topic. Immediately, one comes against the issues of free will and consciousness. Some philosophers believe that both are ineffable; others (most notably Daniel Wegner in *The Illusion of Conscious Will*⁶) believe that both are just exquisitely beautiful illusions. The nature of qualia—how we feel about seeing the color purple, for

example—is a particularly contentious battleground for materialists and non-materialists regarding the philosophy of mind.

Ignoring the delicious verbal sparring one gets when you put a staunch materialist and a passionate nonmaterialist in the same room, I think it is fair to observe that once we step back from

observes and manipulates some small corner of our world. As I drive to the nearest Starbucks, algorithms control the brakes on my car. When I make a call on my mobile phone, a Viterbi algorithm facilitates a clear connection. When my smartphone suggests a route to take, it's typically using some variation of the A* algorithm.

If we peel back the layers of behavior at every level, as a computer scientist, I see algorithms, programs, and systems.

the event of machine-as-man, we still have some weighty issues to metabolize. Assume for the moment that we can't craft a machine that we would call sentient: Just how close might we get to the illusion of sentience? As we continue to surrender ourselves to computing, we find ourselves slowly, inexorably, and irreversibly moving to that event. Some might say that this is a bit like Zeno's paradox: we will never actually get to the other side of this machine-as-man room. And yet, we continue to advance.

The issues of the computability of the mind enter the public consciousness largely through books and movies (Wikipedia offers a list of interesting examples: http://en.wikipedia.org/wiki/List_of_fictional_robots_and_androids). From the outside, such creations do look quite magical and very much noncomputable. However, if we peel back the layers of behavior at every level, as a computer scientist, I see algorithms, programs, and systems behind the mystery.

Algorithm Soup

We swim in an ocean of algorithms, each operating as an invisible hand that

When explaining algorithms to a nontechnical audience, I often use the example of a Japanese tea ceremony. This ritual, when done with intent, is both sacred and purely algorithmic: in the ceremony, the preparing, pouring, and acceptance of the tea is the manifestation of a finite set of steps that eventually halts.

I suggest that the history of many software-intensive products is a history of algorithms. Similarly, the history of many software-intensive industries is a history of architecture and systems.

Hollywood Magic

Consider recent advances in filmmaking special effects. In the earliest days of 3D graphics, the hard problem was just rendering, but then Blinn's work on ray-tracing algorithms changed all that and opened the floodgates to innovation. Next came—in roughly this order—breakthroughs in algorithms for textured surfaces, basic motion, and then, fluids and fractals. Once these came together, artificial creatures and landscapes rapidly spread across the movie landscape, from the avatars in *Tron* to many of the land-

scapes in *Indiana Jones*. Next came breakthroughs in swarms (for example, the battles in *The Lord of the Rings*), hair and fur (Scully in *Monsters, Inc.*), then virtual clothing (Rose's scarf in *Titanic*), and now photorealistic human skin and faces.

These algorithms begat companies such as Industrial Light and Magic and Digital Domain, which could build systems around these algorithms to develop, deploy, and evolve economically viable solutions.

It's now possible to tell compelling stories in ways that would have previously been economically impossible or too dangerous. In his time, Cecil B. DeMille could afford large casts, but have you ever seen the cost of a few thousand well-equipped ogres at today's prices? The liability policies alone would be staggering, not to mention the thousands of sheep and cattle you'd need to feed them. And I think it's fair to say that Mr. Schwarzenegger's lawyers would never allow him to do most of his own stunts. A CGI instance of Arnold can endure take after take of virtual slaughter without the least wear and tear on its bits.

Beyond Minecraft

It's fascinating and delightful to hear the news that a school in Sweden has introduced compulsory classes on Minecraft.⁷ Carnegie Mellon's Center for Computational Thinking has a similar mission, with a focus first on scientific research and then on its implications on public life.

That algorithms have entered the public arena is inevitable and desirable. As more of our lives become controlled by the algorithms in our software-intensive systems, we are well served to understand the nature of what controlled us. For computer professionals, it's our responsibility to devise algorithms that serve users. To that end, I'm reminded of a story about Steve Jobs

in the early days of the Macintosh, in which he berated a programmer about a particular routine that was only slightly inefficient. Jobs observed that if you multiply a one-second savings in time by the millions of people who would endure that delay, it adds up to a significant human cost. Even small things, at scale, have huge implications.

As Turing has shown, one of the more amazing things about computing is that it's universal. Given enough time, a Turing-complete machine could serve as a calculator or a simulation of a mind. Is it possible—is it probable—for us to move from the kinds of programming found in Minecraft to the computability of the mind?

Someone—perhaps someone not yet born—might answer that question more decisively than I ever could. In the meantime, consider this an opportunity to explore the possibility. Engage your noncomputing family or friends on the computability of the mind and observe where that dialogue might lead with regard to the limits of our software-intensive systems. Let it serve as an inspiration to where we might take this technology in the advancement of the human experience. 

References

1. J. Sawday, *Engines of the Imagination: Renaissance Culture and the Rise of the Machine*, Routledge, 2007.
2. R. Penrose, *Shadows of the Mind*, Oxford Univ., 1996.
3. J. Searle, "Minds, Brains and Programs," *Behavioral and Brain Sciences*, vol. 3,

no. 3, 1980, pp. 417–457; doi:10.1017/S0140525X00005756.

4. M. Minsky, *Society of Mind*, Simon & Schuster, 1988.
5. A. Turing, "Computing Machinery and Intelligence," *Mind*, Oct. 1950, pp. 433–460; doi:10.1093/mind/LIX.236.433
6. D.M. Wegner, *The Illusion of Conscious Will*, Bradford, 2003.
7. O. Gee, "Swedish School Makes Minecraft a Must," *The Local*, 9 Jan. 2012; www.thelocal.se/45514/20130109.

GRADY BOOCH is an IBM Fellow and one of the UML's original authors. He's currently developing *Computing: The Human Experience*, a major trans-media project for public broadcast. Contact him at grady@computingthehumanexperience.com.



See www.computer.org/software-multimedia for multimedia content related to this article.

IEEE SOFTWARE CALL FOR PAPERS

Green Software

Submission deadline: 25 June 2013 • Publication: Jan./Feb. 2014

Information technologies (IT) requiring vast amount of energy and other resources are used in almost every field and process. Green IT is the study and practice of using computing resources efficiently to reduce negative impacts on the environment. Green IT is applicable to various high-tech domains, such as datacenters, mobile computing, and embedded systems. Recently, global carbon dioxide emissions reached 9.1 billion tons, the highest level in human history—49 percent higher than in 1990 (the Kyoto reference year). At least 2 percent of global carbon dioxide emissions can be attributed to IT systems, and further increases are expected, with new IT systems being deployed daily. Therefore, reducing the energy consumption and related carbon dioxide emission of IT systems is a crucial requirement. Reducing energy consumption also leads to reduced maintenance expenses and costs of ownership, giving manufacturers a competitive advantage.

Questions?

- For more information about the focus, contact the guest editors:
- Ayse Basar Bener, Ryerson University: ayse.bener@ryerson.ca
 - Maurizio Morisio, Politecnico di Torino: maurizio.morisio@polito.it
 - Andriy Miranskyy, IBM Toronto Software Lab: andriy@ca.ibm.com

Full author guidelines: www.computer.org/software/author.htm
 Submission details: software@computer.org
 Submit an article: <https://mc.manuscriptcentral.com/sw-cs>



So you are already a Certified Professional for Requirements Engineering at Foundation Level, then you have the solid basis you need to take the next step.

Go for the CPRE Advanced Levels

- **Requirements Elicitation and Consolidation**
Get the right skills to obtain all the facts from stakeholders and to separate the wheat from the chaff!
- **Requirements Modeling**
Acquire the best modeling techniques and information will never ever slip through your fingers!

www.ireb.org

The home of Requirements Engineering