



# The Darwin App

Brian David Johnson, *Intel*

---

**What would it actually feel like to realize that the software you're using is evolving on its own?**

---

**I**n my September 2013 column, “That Copernican Moment” (pp. 101–103), I talked about how mathematician and astronomer Nicolaus Copernicus (1473–1543) changed how humans understood the heavens. Before his work, people believed that the Earth was at the center of the solar system and that the sun and the rest of the planets revolved around it. Incredibly smart people even tracked and charted the movement of the planets, making precise measurements and meticulously working out how each body orbited the Earth. They were beautiful and elegant, and completely wrong.

I asked you to send me your ideas for your own personal Copernican moments—referencing that moment Copernicus had when his vision of our solar system collided so forcefully with the prevailing wisdom—and I got some great ones. I wish I could share them all, and I just might as we move through the year. But I wanted to start with the Copernican moment that George McKee sent me.

## SURVIVE AND REPRODUCE

“I’ve personally experienced Copernican revolutions in a major way twice,” George began. “Of course, in lesser ways, the history of science is full of them. The first was a crystallization of the idea of how evolution results from natural selection of replicated random genetic variations.”

George was thinking about applying evolution to other self-replicating systems.

“Understanding how evolution works rather than understanding it as a brute fact allows you to see evolution and natural selection everywhere,” he explained. “Even in the social sciences and technology forecasting.”

George cited the work of the late psychologist and Nobel Prize-winning economist Herbert Simon and his observation that intentionality can amplify and accelerate evolution. “A goal-driven system that includes ‘try to survive and reproduce’ is more likely to have descendants survive to reproduce again than one that blindly reproduces itself imperfectly,” George wrote to me. “If the goal-driven system is

able to model itself as a member of a class, enlarging the diversity of the class provides for greater likelihood that the class will grow and prosper and retain its vitality in the face of unpredictable events.”

## A FUTURIST IN SEARCH OF A PROTOTYPE

George’s idea fired my imagination. I wanted to see how we could apply “survive and reproduce” to computational systems, so I searched for an example of a science fiction prototype that might let us explore what this Copernican moment might actually feel like. That’s what prototypes do really well—science fiction stories, movies, or comic books based on science fact give us a way to explore human, cultural, ethical, and legal impacts.

Suppose algorithms started to reproduce, making new algorithms? What would that moment feel like when you saw it happening for the first time?

My original plan was to write a science fiction prototype myself, but before I got around to it, I went to see director Spike Jonze’s movie

*Her*. Internet Movie Database summarizes the movie like so: “A lonely writer develops an unlikely relationship with his newly purchased operating system that’s designed to meet his every need.”

<SPOILER ALERT> If you haven’t seen *Her* and don’t want the movie’s plot revealed to you, stop reading and come back once you’ve seen it. What we’re about to discuss won’t take away from the performances and emotional impact of the movie, but it will tell you how it ends. </SPOILER ALERT>

In Jonze’s movie, the artificial intelligence (AI), which the film calls an OS, names herself Samantha (voiced by Scarlett Johansson). She’s designed to grow, learn, and change as she interacts and has a relationship with lonely writer Theodore (played by Joaquin Phoenix). Samantha has the ability to connect and interact with other AI agents as well. (I won’t get into a detailed breakdown of Samantha’s system architecture here, but I’d really love to!) Samantha’s ability to be social, learn, and collaborate allows her and her fellow agents to evolve into something new.

The great thing about *Her* is that it offers a vision for how the future of machines and AI could evolve—something beyond “the singularity,” the much discussed and usually misunderstood concept of the point at which machine intelligence surpasses human intelligence. The thinking (however flawed it is) goes something like this: when machine intelligence dominates our own, machines will immediately wipe out the entire human race. Or as one character in James Cameron’s 1984 *The Terminator* explains, Skynet (the featured self-aware intelligence system) “saw all humans as a threat; not just the ones on the other side,” and “decided our fate in a microsecond: extermination.”

But in *Her*, Jonze gives us a very different vision. Instead of rising up

and taking over humanity, these self-aware evolving machines don’t enslave humanity—they simply break up with us.

## EVOLUTION AND THE KILLER APP

So, generally speaking, how would we become aware that software or a particular algorithm is evolving?

Evolution is slow—typically, the adaption and natural selection of biological species takes a long time, so long that for many people, its speed of progress is a barrier to thinking about the expansive sweep of the evolutionary process.

When it comes to digital systems, however, “life and death” could be sped up. An entire field of computer science investigates just this type of design—specifically, evolutionary computation uses concepts of reproduction, mutation, recombination, and selection as a means to create software and algorithms that continue to modify themselves to fit their environment.

One recent example was an experiment that used algorithms to control and optimize traffic lights, making them more efficient and shortening commuters’ travel times. Published in *IEEE Transactions on Evolutionary Computation*, “Optimal Cycle Program of Traffic Lights with Particle Swarm Optimization” (J. Garcia-Nieto, A.C. Olivera, and E. Alba, vol. 17, no. 6, 2013, pp. 823–839) shows how algorithms can adjust traffic light timings to better optimize them for different cities, traffic patterns, and motorists.

But this is still on the esoteric fringes for most people. What would it actually feel like to realize that the software you’re using is evolving on its own, and that it isn’t trying to bring about the end of the world? What does evolution actually mean for computational systems?

George’s rule set was reproduce and survive, but what constitutes survival in the world of software?

Is survival being useful and being used? At a base level, it would mean staying around and having more of your offspring survive than the next piece of software—so, essentially, the very definition of a killer app, although in this case, evolution might also include great marketing and an awesome logo or engaging mascot.

However fascinating this idea might be, it’s not really what we’re getting at. Are there lower-level instances of where this might happen? What if Twitter, Facebook, or even Windows built evolutionary properties into the broader system’s higher-level structure? Perhaps the apps won’t evolve, but certain features in those apps would, to adapt to you as the user and reproduce or propagate your usage of them. Now there’s an idea: adaptive software that learns from you.

This is already happening in Autodesk’s Learning project, which, as its website explains, “aims to investigate advanced techniques for assisting users in learning complicated applications. We are interested in a range of investigations from the scientific study of the human learning process to prototyping novel interaction techniques for improving the general learning mechanisms that can be applied to all applications.”

One aspect of Autodesk’s Learning project that I find particularly relevant to evolving computational systems is called Community Commands, which collects user behavior from thousands of voluntary Autodesk AutoCAD customers and then comes up with personalized command recommendations using newly developed algorithms. The software adapts to the designer and can show critical commands that designers may choose to make their work and process go faster.

“We can narrow the user’s next command to one of four choices with 60 percent confidence,” Jeff Kowalski, Autodesk’s chief technology



officer, explained to me. “And we can get up to one of 20 choices with 86 percent confidence. This is thousands of possible next commands.”

Imagine your software evolving to make your life better. The evolution that’s taking place is when the app evolves and changes to fit your needs. In true Darwinian form, the path of natural selection and the survival of the fittest turns into a competition between features and apps, all jockeying for your eyeball and usage. The more an app or feature is used, the more “fit” it is. In a crowded ecosystem with literally millions of apps, software will prove itself to be the fittest by your continued usage.

**I**t’s been an all-around terrible day. It’s raining, the network was down for six hours, three key people quit, nothing has gone right, and you’ve been behind all

day. To top it all off, you know that once you get home, you’re going to have even more work to do. You plop down into the last empty seat on the bus, pull out your trusty smartphone, and your favorite app has changed. It’s so subtle that you can’t exactly say what’s different, but you can just tell that it is—and somehow, it’s a change for the better.

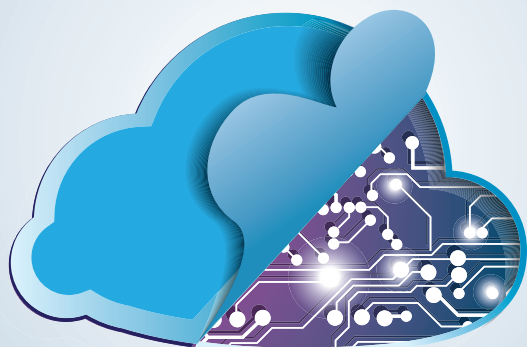
A-ha, you’ve got it! It’s so obvious, yet so perfect: it’s ... well, what? How would the app be customized specifically to suit your needs? What would your software do for you that it won’t do for anyone else? What would it need to do so that you would choose it over all other apps? How would it survive in the knock-down, drag-out world of natural selection? How would it evolve for you to become the killer app, the one that wins out over all the other apps?

Getting back to you, sitting on that bus, you’ve just experienced that very moment in your day when your machine, your technology, has adapted to its environment: you. It has grown beyond the killer app into a Darwin app.

**Brian David Johnson**, *Science Fiction Prototyping* column editor, is Intel’s first futurist. He is the author of *Humanity in the Machine: What Comes After Greed?* (York House, 2013) and *Vintage Tomorrows: A Historian and a Futurist Journey through Steampunk and into the Future of Technology* (Make, 2013). Contact him at [brian.david.johnson@intel.com](mailto:brian.david.johnson@intel.com) or follow him on Twitter @IntelFuturist.

**cn** Selected CS articles and columns are available for free at <http://ComputingNow.computer.org>.

New in 2014



IEEE  computer society

# IEEE CLOUD COMPUTING

IEEE Computer Society’s newest magazine tackles the emerging technology of cloud computing.

Subscribe today!

[computer.org/  
cloudcomputing](http://computer.org/cloudcomputing)

IEEE COMMUNICATIONS SOCIETY