The Second Computer Revolution

Author : Andras Tantos

Categories : Random Ramblings

I've had a glimpse into to the future. It was bright. It was delightful. It was alien and frightening. Nevertheless I have seen it and now I can't shake it out of my head. I'll write it down and hope that it will bring some piece to my mind. Or at least upsets yours…

The first revolution

When the computer was invented in the '40s (all right, some will for certain debate that, let's just say that the first electrical, digital computers were invented around then) it had a very immediate purpose; to help solve equations faster to:

- Break the encryption of Nazi messages
- Make better aiming apparatus for various cannons

In other words to help kill people faster.

But with that, the Genie was out of the bottle and people soon started to see that these machines could be used for other things as well. One of its first applications was processing payroll and the famous election night predictions. Soon anything and everything that had to do with numbers was 'computerized'.

If you look carefully however, you can already see the main characteristic of this revolution: computers are used to automate existing activities. They solve existing equations faster. They replace people in payroll. They become a better bean-counter, an enhanced paper, an improved pen. In other words, the computer became the universal tool.

This is not a great surprise if you think about it's origin: a computer is essentially a physical realization of the Universal Turing Machine. A machine, that can emulate all other (Turing) machines, a universal tool in deed.

With all that progress though you would be hard pressed to find an application of computers that really required one. Sure, it's nice to have a smart-phone in our packets, but we could make phone-calls without them. It's neat that we can program the microwave down to a second, but surely ovens predate computers. The latest hype – driverless cars – is just another addition to this long list. Sure you can sit in a car Today and get to places without driving one: it's called a taxi cab!

Few people saw the second revolution coming. Edsger W. Dijkstra was one of them (note the date on the article: 1988). I for one, wasn't. I needed to see it first-hand.

The second revolution

So what is this second revolution business? This is the phase, when computers start to perform truly unique tasks, things that would be impossible to do without them. When these machines stop being empty shells that we can shape to the tool we want it to be – a pen or a cab driver – and become a first-class tool on their own right. I will give you four examples of what is possible when you treat computing machines as a new class of tools:

Video games

I'm not talking about chess games with a computer here. Those belong to the previous phase, where the computer replaced your human opponent. I'm talking about true video games, like Space Invaders, Pac-Man or Mario Bros. Here – and this idea is coming from Bret Victor's talk – the gameplay is truly unique. These games could not have possibly existed without computers. You can't make Pacman out of paper or blocks of wood. You can't play Space Invaders against a human opponent. You can't write down the story of Mario in a book. Their existence is only possible due to the invention of computers. I could have chosen more recent examples, but I wanted to go with the classics to prove that the idea is not new. All these games were released around 1980. These old games also show that the computer is not there to make dazzling graphics, surround sound or photorealistic rendering. It's there to create a new art form if you wish, a new type of expression, a new way of engaging an audience. In fact a new type of audience as well: the gamer.

To give it a name, let's call the innovation here responsive interactivity. The idea that the object you play with not only interacts with you, but reacts and changes its behavior based on your actions. Which leads to my second point:

Interactive art

This is the general idea that Bret Victor further explores in his talk: Stop Drawing Dead Fish. Responsive interactivity – that computers can be taught to react to your input – can be exploited as an art medium. His example is to create an animated live performance on stage using simple shapes, pre-programmed behaviors and – of course – a computer. This is not a traditional cartoon however. Being a live performance adds spontaneity to it that never existed in that art-form. It's probably closest to a puppet-theater though that analog breaks down as well as at least part of the action of the 'puppets' is not controlled by the 'puppeteers'.

Internet

My third example is one of those over-used words that mean too many things to actually mean anything. To understand its true disruptive nature you have to look at the social change it creates. It is made most evident in the Arab Spring. Sure, revolutions have happened before, even waves of revolutions that swept continents happened, but the way the internet allows people to interact on a global level is new and unprecedented. The free flow of information and goods literarily tears the old social, economic and political fabric apart. Regional leadership becomes irrelevant and next to impossible. Central control of information – the main tool of mass manipulation – becomes very hard. The old 'bad word' for this is globalization, but the main driving force behind that is the Internet too. Friedman wrote a whole book on the subject.

The key idea is democratized information distribution. The next step over Guttenberg (who made it possible to everyone to consume ideas): make it possible for everyone to create and distribute ideas.

Computational Architecture

From these grand visions lets get back to something more mundane, and the main trigger for my little rant here: architecture. In his TED talk, Michael Hansmeyer talks about building unimaginable shapes, columns in his case. The point of his talk is that by using computers he was able to create forms that not only could not have been created without computers, they could not have even been imagined without them. The very process of creating these forms involves using computer algorithms (fractals if I understand what's going on behind the scenes correctly). When I first saw his work, I got very excited. This was the first time I've realized that I'm looking at something truly new. And since – at least in my view – big architectural changes have always been tied to enabling technologies – the invention of the arch or reinforced concrete – this new application of computers appears to be truly revolutionary. True, it's only involved in the esthetics, not in the structure – it's still a column, in fact arguably a column that can't support weight at the moment – but it's the first baby-step towards something much more important: a new artistic language for architecture, a truly new style.

The spark in this example is to make the algorithm and the human equal entities in the creative process.

Hallmarks of a revolution

Let me close by summarizing the key aspect of these wide-ranging examples: Analogies break down. In fact, if you find a good analog to explain what's going on (you store 'files' in a 'folder' just like you put 'papers' in 'filing cabinets') chances are, it's not revolutionary.

Consequently as Elon Musk have put it at the end of his TED talk (at 19:32 if you're not in the mood of listening through 20 minutes of unrelated stuff), you have to let go of reasoning through analogy to arrive at these truly new ideas. You have to use axiomatic thinking (or use the way physicists work as he put it). You have to boil things down to their fundamental building blocks and start building things up from there. Or, as I suspect in some of the cases above, just try random ideas, be brave enough not to give up and stumble upon something.

It's frightening to work on something that doesn't feel comfortable, known and familiar. It's all too easy to incrementally improve upon the known – putting a touch-screen on a cell-phone.

This is what Bret Victor is talking about in his 'The Future of Programming' talk.

This is what Alan Kay was talking about in ‘The Computer Revolution Hasn’t Happened Yet’.
This is what Dijkstra was talking about in ‘On the cruelty of really teaching computing science’.

Maybe we should start listening…