

---

# Computational Creativity

The Science Museum in London once exhibited some interesting machines made from Meccano which were able to perform complex mathematical calculations. As these machines were built in the 1930s, the Meccano magazine from June 1934 speculated about the future in an article entitled: "Are Thinking Machines Possible?" They couldn't have possibly known the impact the computing age would have, but they were already certain about one thing: to my horror, at the end of the article, the author said: "Truly creative thinking of course will always remain beyond the power of any machine".

A full ten years before programmable computers were born, and decades before they started to show any signs of intelligence, people were already condemning them to an uncreative future. This reaction is entirely understandable. We can't run as fast as tigers, swim like dolphins or climb like goats. But we have our smarts—that's our thing. It's been hard enough to admit that computers might rival us at tasks which require intelligence. So we won't easily relinquish our position as the only creative beings on the planet.

Under this weight of prejudice, research into computational creativity has lagged behind other areas of AI research a little. However, there is a small band of us who pursue the goal of getting computer programs to creatively produce poems, sonatas, paintings, theorems, jokes, and much more. We've recently reached the stage where there is a sufficiently large number of such programs for us to be able to compare and contrast them in a meaningful way. This has enabled us to begin to come to consensus about the main issues in computational creativity.

Firstly, we've realised that we're working in a different paradigm to the majority of AI researchers. When faced with an intelligent task to perform, AI people generally

think of the task in terms of solving a problem. That problem might be planning a route from London to Liverpool, proving a mathematical theorem, or recognising hand-written words. But in each case, AI techniques are developed that can (hopefully) solve the problem as well as or better than humans.

Computational creativity researchers, on the other hand, work in an artefact generating paradigm. Here, the task is to generate artefacts of real value to someone. Those artefacts may be plot lines for a play, a mathematical theorem, or a harmonisation for a Bach chorale. There is a lot of overlap between the two paradigms, but they each have their peculiarities. For instance, problem solving AI programs know when to stop: when they have solved the problem to a satisfactory degree. With AI artefact generation programs, however, it's often not clear when to stop them (humans have this problem, of course: it's never easy to know when to stop painting a picture).

Another difference is aesthetics. Usually, the value of solutions generated by problem solving programs is measured only by the success of the solution to the problem. Nice and simple. With artefact generation, however, there are many competing ways of assessing the artefacts and different aesthetic considerations have to be taken into account. Many of these will be specific to the musical, artistic, literary or scientific application at hand. Others will be more general, such as whether the artefact is novel, surprising, or evokes an emotional response.

So far, to engineer our creative programs, we've stolen anything we can from Artificial Intelligence and elsewhere. There are attempts underway to look at the methods being used and characterise some of them in terms of the kind of search they perform: do they just look very hard

through thousands of similar artefacts for a good one, or do they somehow transform the way in which artefacts are generated, which might be considered more creative. Often, if the methods involve some randomness, or are so complicated that we can't explain their actions, this may increase our perception of a program's creativity. If we can completely describe how a program produced an artefact, the chances are that no matter how pleasing the artefact is, we would not think of the program as being particularly creative.

Usually, creative artefact generation programs need three types of methods: those which mimic a human skill; those which mimic human appreciation of artefacts; and those which mimic our imagination. Imagine an artist missing one of skill, appreciation or imagination. Without skill, they would never produce anything. Without appreciation, they would produce things which looked awful. Without imagination, everything they produced would look the same. It is usually the imagination part that we have the most difficulty simulating, and often we have to approximate this by getting our programs to search through millions of possibilities.

A very important issue is the assessment of the creativity of programs. In the problem solving paradigm, if a new program solves a previously unsolvable problem, or solves a bunch of problems faster than all other programs, then clear progress has been made. As creativity is such a subjective notion (is your child really as creative as you say?), it's much more difficult for us to compare the creative abilities of different programs. However, much progress has been made towards telling whether we should use the word creative to describe a program and telling whether one artefact generation program is performing more creatively than another.

---

## Computational Creativity (cont.)

It is wrong to think that when computers start acting creatively, artists, musicians, and poets will be out of a job. Why on earth would Lucien Freud stop painting just because a computer can paint as well? Moreover, people will always appreciate the blood, sweat, and tears expended by creative people in producing their works. With this in mind, much research in computational creativity has looked at how to enhance or supplement the creativity of people undertaking creative endeavours. In the same way that a composer expects creativity in a performer, we can begin to expect that computers will act as creative collaborators in our projects.

As a society, we leave behind our creations, so surely it is worthwhile having more creativity in the world. And think of

the great gadgets we can have if computers become truly creative: a website to generate topical jokes for the speech you're giving tonight, iPods which can generate entirely new tracks to suit your mood, and fridges which can concoct a delicious recipe to fit their meagre contents. As the phrase goes, we are limited only by our imagination. But there is the crux of the matter: if we have such a limitation, can we really afford to ignore computational creativity?

Simon Colton  
Imperial College, London  
sgc@doc.ic.ac.uk  
<http://www.doc.ic.ac.uk/~sgc/>



## Conference Report: ICANN 2007

ICANN is one of the main European conferences on neural networks and it took place this year from the 9-13th September 2007 in Porto, Portugal. The conference program included one day of tutorials and half a day of workshops, which gave ample scope for discussions. The main emphasis of the conference was on machine learning, and there were also sessions on other topics, such as spiking neural networks and cognitive systems.

One of the most interesting presentations was given by Felix Schürmann on the Blue Brain Project. This is taking place at the Ecole Polytechnique Fédérale de Lausanne in Switzerland and its overall goal is to reverse engineer the mammalian brain. The current focus of the Blue Brain Project is on accurately reproducing the behaviour of in-vitro neural tissue from the rat and clarifying what kinds of data need to be recorded during in-vitro experiments (to enable in-vitro results from different groups to be reproduced and more systematically compared).

The Blue Brain project is starting with a single cortical column, which is the

basic functional unit of the cortex. Cortical columns occupy a cylindrical volume 0.5 mm wide by 2 mm high and contain around 10,000 neurons interconnected with 30 million synapses. This microcircuit is repeated millions of times across the cortex and it is similar between species - the main interspecies difference is that human brains have many more cortical columns than smaller mammals. The Blue Brain Project has been simulating single cortical columns using biologically accurate neurons with realistic connectivity. The simulations are being carried out on an IBM Blue Gene supercomputer, which contains 8192 processors and 2 TB of RAM—a total of  $22 \times 10^{12}$  teraflops processing power. The simulation generates 160 GB/s of data and the team has had to develop strategies to store and process this information and they have also created some impressive 3D visualizations. The software used for the simulation is a combination of the large scale Neocortical Simulator [7] and NEURON [8]. The first simulation of the rat cortical column was carried out in 2006 and it is currently

running at about two orders of magnitude slower than real time.

Another highlight of the conference was the presentation on 'Modelling Consciousness with Neural Architectures' by Igor Aleksander, which gave an overview of his work on machine consciousness. Aleksander aims to understand consciousness by building systems that are not living and might be attributed some form of consciousness, and his theoretical approach is based around five axioms that he claims are minimally necessary for consciousness[1]. These axioms are depiction, imagination, attention, planning and emotion, with depiction being the most important. According to Aleksander, these axioms are a preliminary list of mechanisms that could make a system conscious, which should be revised as our knowledge of consciousness develops—a useful starting point that can be used to test ideas and develop the field. These axioms were deduced by Aleksander using introspection, and he also identifies neural mechanisms that could implement them in the brain.

Aleksander has developed a kernel ar-