RN/02/20 Updated version of that discussed at "Workshop on Grand Challenges for Computing Research" Edinburgh, 24-26 November 2002

Grand Challenges for Computing Removal of the Man-Machine Interface Bottleneck

"Do what I ment not what I said"

W. B. Langdon
Computer Science
University College, London

Riccardo Poli
Department of Computer Science
University of Essex

October 18, 2002

The Knowledge Elicitation Bottleneck

Programming computers to do ever more intelligent things for us is likely to require huge amounts of information to be transferred to a computer in one way or another (this may mean the production of huge amount of software, or the creation of huge databases of knowledge, or more likely both). There seems no way around that. Who is going to provide that information? Given the low bandwidth with which any human can communicate with a machine, there is no hope; not even putting together large groups of programmers/knowledge-engineers working at full speed for the duration of their lifetime; not even if we could communicate with computers as efficiently as we can communicate with other humans (unless we find ways of interfacing our brains directly with computers, and maybe not even then).

So, while memory sizes and hardware speeds are still growing at an exponential speed, and for all we know, they will probably continue to do so for the next twenty years (the time horizon for the grand challenges), software development is facing harder and harder times, since the MMI bottleneck induces a software production bottleneck (which in turn induces a bottleneck in the improvement of software functionality and intelligence).

Our challange is overcoming the man-machine communication bottleneck.

The Dream: DWIM

The knowledge elicitation bottleneck and its consequences are made worse by the poor quality of today's Man-Machine Interfaces (MMIs). Since the very beginning of automatic computation, people have been viewed as the servants of the machine, forced to pay homage to it via appalling interfaces. Banks of switches 20 or more long, manual card punches and the like were common. So it is common to point to the modern PC and say "Look! This is so much better than what we had before". Indeed the IT industry is proud of having forced the USA developed Qwerty WIMP (Window, Icon, Menu, Pointing device) system onto the world. But this is still an enormous waste of the most precious resource, people. Today people are still forced to adapt to the computer. But these days, instead of a few "boffins", it's millions of people. We are wasting the time of nearly the entire population.

The MMI bottleneck takes two forms. Firstly sheer clumsiness, which we are all familiar with. Using computers is far from intuitive. Today, huge numbers of people have to be trained to do it.

And then retrained, at the next software upgrade. The second form of MMI bottleneck is much harder and more important; it arises because YOU HAVE TO SAY WHAT THE COMPUTER HAS TO DO. It does not work out what you want. Computers do what they are told. As well as deskilling the process of instructing them, we need much more "intelligent" computers, so they can "Do what I ment (DWIM) not what I said". That is, we need computers that can alleviate the MMI bottleneck by inferring more from the limited communication they can have with humans.

This creates a sort of chicken and egg situation. In order to alleviate the MMI bottleneck we need more intelligent interfaces, but, as we mentioned in the previous section, in order to get intelligence in a machine we need to have removed the MMI bottleneck first. How can we break out of this loop?

A Possible Way Forward

Animals, particularly humans, appear to have solved the communication bottleneck problem: their offspring have a variety of mechanisms to acquire knowledge and behaviour from their environment in a largely autonomous way. Some of these mechanisms are innate, while others require time to be developed (learning to learn). This is a sort of bootstrapping process, which overcomes the human-to-human communication bottleneck. It is arguable that a similar type of bootstrapping process could solve the MMI bottleneck.

Maybe one possible way to achieve this is to:

1. Partly instruct machines through human programming and knowledge elicitation.

2. Provide them also with mechanisms to induce part of their own software/knowledge from direct experience in the real world.

3. At the same time make them use those mechanisms also to induce better interfaces to do 1. and new techniques to 2. better. (Note that this process is recursive/bootstrapping, since any improved mechanisms to do 1. and 2. could be applied to 3. too.)

Software evolution (genetic programming) could be one of many mechanisms used to achieve this; machine learning and HCI could provide others; automatic deduction methods could provide some more; etc.

What would the future be like without the MMI bottleneck

As one example of what hardware and MMI improvements could support, consider that, in the future we could see photo realistic virtual reality personal assistants. All sorts of applications are possible. We expect a single VR "personality" to be able to provide them all. Many practical applications are feasible with modest improvements to existing algorithms. Even if the whole system lies some distance in the future. Such a VRP could act as an office assistant one moment, travel agent the next. With connection to existing GPS and on line maps and guides the VRP could act as a multi-lingual travel guide. With better embodiment of the virtual object within the real environment, applications like chauffeur and night watchman, become feasible.

Without machine intelligence and automated production of software all these applications will remain beyond the existing software industry. With existing approaches, there will never be enough people to code even a tiny fraction of these applications.

"What would computing be like if computers were infinitely fast, infinitely small and cost tuppence?" While obviously absurd, this is the logical extrapolation of existing hardware trends. Computers and their tasks would be very different from today but without removing the MMI bottleneck today's software problems will remain.