School of Computer Science
University of Birmingham UK
CN - CR
Every intelligent ghost must contain a machine
an information-processing machine

NB LINK TO FULL CONFERENCE PROGRAMME NB

Invited talk at CAS 2014 Conference University of Birmingham
Local Arrangements

**This is an invitation to influence my presentation,
by sending me suggestions for classroom implementations
of the ideas below.
Write to: a.sloman@cs.bham.ac.uk
(All contributions will be acknowledged)**

**Talk to be presented at CAS Conference on June 21st 2014
at the University of Birmingham at 2.35pm**

What forms of computational thinking will our children need when they grow up?

(This outline is work in progress: Liable to change considerably)

Aaron Sloman
School of Computer Science, University of Birmingham.
(Philosopher in a Computer Science department)

Installed: 20 May 2014
Last updated: 20 May 2014; 14 Jun 2014

This abstract is
http://www.cs.bham.ac.uk/research/projects/cogaff/misc/cas-2014-compthink.html
A PDF version may be added later.
A partial index of discussion notes is in
http://www.cs.bham.ac.uk/research/projects/cogaff/misc/AREADME.html

Update 14 Jun 2014: I’ll include the Turing Test

The presentation will include background information on the Turing test and the relevance of chatbots to education in computational thinking.

I was one of the judges at the recent Turing Test at the Royal Society London, where it was claimed that one of the competing chatbots, Eugene Goostman, had been the first to pass the Turing test. This produced an enormous furore. Here’s a tiny sample of comments on the world wide web:

http://www.reading.ac.uk/news-and-events/releases/PR583836.aspx
http://metro.co.uk/2014/06/09/actually-no-a-computer-did-not-pass-the-turing-test-4755769/
http://mashable.com/2014/06/12/eugene-goostman-turing-test/
http://www.bbc.co.uk/news/technology-27762088
http://iangent.blogspot.co.uk/2014/06/why-passing-of-turing-test-is-big-deal.html
http://www.ibtimes.co.uk/why-turing-test-not-adequate-way-calculate-artificial-intelligence-1452120
http://www.thestar.com/news/world/2014/06/13/was_the_turing_test_passed_not_everyone_thinks_so.html
http://www.theguardian.com/commentisfree/2014/jun/11/ai-eugene-goostman-artificial-intelligence
http://anilkseth.wordpress.com/2014/06/09/the-importance-of-being-eugene-what-not-passing-the-turing-test-really-means/
http://www.wired.com/2014/06/beyond-the-turing-test/
http://www.theverge.com/2014/6/11/5800440/ray-kurzweil-and-others-say-turing-test-not-passed
http://www.newscientist.com/article/mg22229732.200-why-the-turing-test-needs-updating.html
http://www.washingtonpost.com/news/morning-mix/wp/2014/06/09/a-computer-just-passed-the-turing-test-in-landmark-trial/

(Click here to see lots more!)

The idea that Turing proposed his imitation game as a test for machines with intelligence is a myth, believed mainly by people who have never read his 1950 paper.

Some people think that there could be a behavioural test for intelligence but the so-called Turing Test is not stringent enough, so they propose alternative, more demanding tests. I think they are all missing the inappropriateness of attempting to provide such a test. Could any one test enable us to decide which types of animals are intelligent? The search for a behavioural test for a machine being intelligent is as misguided as a search for a behavioural test for a machine being computational.

Instead we need a deep theory of varieties of intelligence and their properties and means of implementation, and an investigation of which forms were produced by biological evolution and which forms can and cannot be implemented on digital computers, neural computers, chemical computers or some combination. This would extend the deep theories developed since the early 20th Century of varieties of digital computation, what they can and cannot do, varieties of implementation, what they can and cannot explain, and much more. A project to develop such a deep theory based on a study of achievements of biological evolution is summarised here.

I suspect that that is one of the things Alan Turing would have worked on if he had lived longer, extending his pioneering work in 1936 that contributed so much to the theory of discrete computation, as well as his later work on developmental processes in organisms.

Adam Ford interviewed me about the Turing Test, its limitations, and alternative approaches to understanding intelligence, on 12 June, and posted the video here:
https://www.youtube.com/watch?v=ACaJlJcsvL8
I have an extended discussion here (work in progress).

Relevance to Computing at School

The supposed Turing test is really just a test for whether a text-only chatbot can, for a while, deceive certain types of human, into thinking the chatbot is also human. That may be an interesting and mildly (sometimes more than mildly) entertaining type of project, even if its scientific and philosophical importance has been hugely over-rated. However, there is an important educational function. Teaching children to develop chatbots using a succession of increasingly sophisticated programming methodologies to enhance their chatbots could be a major contribution to computing education, teaching students many forms of computational thinking of varying depth and sophistication.

For example, they could start off using only long lists of explicit conditional instructions (e.g. ’if input = "Hello" then ...’), then progress to using pattern matching, then grammars, parsers, semantic interpreters, fixed databases, changing databases for simulating actions, rule-based systems for combinng different sorts of knowledge in generating a reply, logical inference systems, constraint-propagation systems, neural nets, evolutionary computation, connections to microphones and speakers, connections to cameras and manipulators, embedding in a mobile robot, and many others.

From my experience of teaching AI to beginners at Sussex University many years ago, I suspect it would help to attract more girls into deep forms of programming and computer science, as well as contributing to the education of computationally informed psychologists, linguists, neuroscientists, biologists, philosophers, and others.
Here’s a 30 year old chatbot used in that teaching process.

Peter Millican (Philosophy Department Oxford University),
who will be talking earlier at the conference has a web site
Elizabeth: An Educational Chatterbot for Windows
With downloadable code
http://www.philocomp.net/ai/elizabeth.htm

ORIGINAL ABSTRACT FOR CAS 2014

This was my proposed abstract before the Turing Test became a possible focus.

What forms of computational thinking will our children need when they grow up?

I was asked to give a talk at the CAS 2014 conference based on this 2012 presentation:

> What is computational thinking? Who needs it? Why?
> How can it be learnt? (Can it be taught?) (PDF) (FLASH at slideshare.net)
> Also presented at ALT 2012, in this video (talk without slies):
> http://www.youtube.com/watch?v=QXAFz3L2Qpo

Since the previous presentation is freely available, I thought that this time I should focus more on generally unnoticed requirements for the future of computing education -- with help from members of the audience, if possible.

IDEAS INVITED:
Anyone with ideas related to this topic and Computing at School is welcome to email me at a.sloman[AT]cs.bham.ac.uk. Let me know if you will not mind my adding your comments to this document, with or without your name (please specify). Suggestions or questions from teachers particularly welcome.

Note added 4 Jun 2014: Seymour Papert: Let’s Tie the Digital Knot

I have just come across this paper by Seymour Papert challenging many educators and researchers on education, about 16 years ago:
http://www.ait.net/technos/tq_07/4papert.php **Let’s Tie the Digital Knot**
in Technos Quarterly Winter 1998 Vol. 7 No. 4
His answer to the Question: "What sort of force can enable deep changes to happen in education?"
"I call the force Kid Power."

ABSTRACT
What forms of computational thinking will our children need when they grow up?

For some people "Computational thinking" means designing algorithms and writing code. Around 2005 Jeannette Wing (see below) proposed a broader interpretation emphasising everyone, including children, learning to think "like a computer scientist", echoing earlier educational ideas of Alan Kay, Seymour Papert and Marvin Minsky. New forms of computational thinking are now being developed, replacing older, shallower, forms of thinking, in many fields, including evolutionary research, neuroscience, psychiatry, cancer studies, developmental biology, chemical engineering, linguistics, economics and even philosophy, among many others: all of which study naturally occurring systems that acquire, store, manipulate, and use information of many different kinds encoded in many different media. Even Alan Turing, a major pioneer of electronic computers saw the need for a broader view, e.g. in his 1952 paper on morphogenesis in biology, where chemistry controls development. His famous 1950 paper introducing the ’Imitation game’ claims that chemistry is also essential to what brains do. As more and more academic and applied disciplines discover the relevance of computational thinking broadly understood, can we prepare our brightest children for the challenges they will encounter as **future** thinkers and leaders, including problems of understanding the workings of our own minds and how they evolved, and how they develop, rather than simply preparing youngsters to do more of what is already being done now?

This may include introduction of radically new forms of computation into future education.

The 2006 version of Jeannette Wing's ideas on Computational Thinking:
http://www.cs.cmu.edu/afs/cs/usr/wing/www/publications/Wing06.pdf

---

Problems, obstacles, challenges, and opportunities
How to make progress.

TO BE EXPANDED: SUGGESTIONS WELCOME ESPECIALLY FROM TEACHERS WITH EXPERIENCE OF INTRODUCING "NON-STANDARD" TEACHING MATERIALS

A selection of previous discussions and presentations

- http://www.cs.bham.ac.uk/research/projects/cogaff/misc/gc-ed-sloman/
  Education Grand Challenge proposal (2004)
  A New Kind of Liberal Education
  Making People Want a Computing Education For Its Own Sake

- http://www.cs.bham.ac.uk/research/projects/cogaff/misc/alevel-ai.html
  A Possible Artificial Intelligence/Cognitive Science GCE/A-Level Syllabus (2007)

- http://www.cs.bham.ac.uk/research/projects/cogaff/talks/#talk100
  Architectures for more or less intelligent life
  How to turn philosophers of mind into engineers
  -- to help them solve old philosophical problems
  Guest lecture for Philosophy of Cognitive Science Students, Birmingham Feb 2012.

- http://www.cs.bham.ac.uk/research/projects/cogaff/misc/cas-teach-share-2011.html
  What is computing? Why should it be taught? CAS teach-share 2011

- http://www.cs.bham.ac.uk/research/projects/cogaff/talks/#cas11
  http://www.slideshare.net/asloman/sloman-casteachshare
  Computing: The Science of Nearly Everything. Including Biology!
  Slides for CAS "TeachShare" presentation, 8 Jun 2011

- http://www.cs.bham.ac.uk/research/projects/cogaff/talks/#talk106
  Talk 106: Thinking Architecturally
  Talk at Architectural Thinking Workshop Cambridge 28-9 Nov 2012

- http://www.cs.bham.ac.uk/research/projects/cogaff/sloman-clowestribute.html
  EXPERIENCING COMPUTATION: A Tribute to Max Clowes
  Originally published in 1984, recently updated and enlarged.

- http://www.cs.bham.ac.uk/research/projects/poplog/examples/kinds
  Kinds Of Programming For Learners, including
  - Numbery Programming
  - Bumpy Programming
  - Gadgety Programming
  - Arty Programming
  - Presentation Programming

  - Lifey Programming
  - Modelling Programming
  - Exploratory Programming
  - Utility (or "Appy") Programming
  - Gamey Programming
  - Teachy (tutorial) Programming
  - Thinky Programming

- http://www.cs.bham.ac.uk/research/projects/poplog/examples/#thinky
  Tips on teaching ’thinky programming’.

- http://www.cs.bham.ac.uk/research/projects/cogaff/talks/#ncsl
  Also on slideshare.net in flash format.
  Talk 72: Some thoughts and demos, on ways of using computing for deep education on many topics.
  As a change from teaching:
  - "useful" skills (of various kinds),
  - uses of computing,
  - computer science
  - computer/software engineering.

- http://www.cs.bham.ac.uk/research/projects/cogaff/misc/eric-schmidt-lecture.html
  Comments on Lecture by Eric Schmidt (Google)
  MacTaggart lecture Edinburgh Festival 2011
  Friday 26 August 2011
  Reported by BBC http://news.bbc.co.uk/today/hi/today/newsid_9574000/9574156.stm

  Why didn’t he consider the computational needs of other sciences, and the implications for computing education? Few do.

Maintained by Aaron Sloman
School of Computer Science
The University of Birmingham