NUI MAYNOOTH

Ollscoil na hÉireann Má Nuad

PACT:

Programming $\wedge$ Algorithms $\Rightarrow$ Computational Thinking

Aidan Mooney James F. Power

April 30, 2014

1 Introduction

The PACT programme is a partnership between researchers in the Department of Computer Science at NUI Maynooth and teachers at selected post primary schools around the country. Starting in September 2013 a number of Irish secondary schools took part in a pilot study, delivering material prepared by the PACT team to transition year students.

Initially the focus of this partnership is on teaching programming to transition year students, but ultimately the goal is to develop a framework for delivering a short course on *computational thinking* as part of the new Junior Certificate cycle.

2 Background and Motivation

NUI Maynooth has been delivering programmes in Computer Science at undergraduate and postgraduate level for 25 years and, since 2012, has been delivering a unique *BSc in Computational Thinking*. This degree is a blend of Computer Science, Mathematics and Philosophy, and aims to provide a deeper education in computing than the more traditional degrees, which often have a strong vocational orientation. The PACT initiative aims to build on the visibility of the *Computational Thinking* brand to bring the ideas at the heart of this degree to second-level students.

What is Computational Thinking? The phrase *Computational Thinking* was originally coined in the context of mathematics education by by Seymour Papert [8], but came to prominence in Computer Science following an influential article by Jeannette M. Wing [10]. Wing's article described computational thinking as

> "Computational thinking builds on the power and limits of computing processes, whether they are executed by a human or by a machine. Computational thinking confronts the riddle of machine intelligence: What can humans do better than computers? and What can computers do better than humans? Most fundamentally it addresses the question: What is computable?"

This perspective has been developed by the *Center for Computational Thinking* at Carnegie Mellon University[1] (sponsored by Microsoft Research), and echoed in other programmes, including those by Google[2] and the *International Society for Technology in Education*[3].

Crucial elements of this perspective are that:

- Computational thinking is a way that humans, not computers, think.
- Computational thinking is a fundamental skill for everyone, not just for computer scientists.

Computational Thinking in Schools. There have been numerous efforts over the years to introduce computational concepts into Irish secondary schools but, at an official level, these never proceeded much beyond basic elements of information technology. The reform of the Junior Certificate cycle offers an opportunity to get this right: a course designed by Computer Scientists to display the depth and beauty of the field in a way that can challenge and engage second-level students. An emphasis on *Computational Thinking* allows us to explore the key concepts the underlie Computer Science, without necessarily having to achieve the full rigour of the professional scientific discipline.

We can identify three key levels of understanding in Computer Science:

- **Programming** is a threshold concept in Computer Science, as it introduces some of the basic challenges of the discipline.
- **Algorithms** involves studying solutions in computational terms: which solutions are better, in what circumstances, and why?
- **Computability** is the study of *problems* in computational terms: what can be computed, and why?

The goal of the PACT programme is to guide students through the key topics in programming and algorithms towards the ultimate goal of studying the process of computation itself.

The PACT group at NUIM has collaborated with teachers across 9 secondary schools to develop a flexible module which will engage Transition Year students. The focus of the module is not on learning facts about computers but on developing creative ideas and new ways of thinking. Continuing feedback will be used to expand the module into a full Junior Cycle short course.

[1] http://www.cs.cmu.edu/~CompThink/
[2] http://www.google.com/edu/computational-thinking/
[3] http://www.iste.org/learn/computational-thinking

3 Related work

The NCCA has developed a specification for eight Junior Cycle short courses for use from 2014, and one of these is titled *Programming and Coding* [7]. This is a useful document as it provides an example of addressing the general skills requirements of a short course. The computing content seems to consists of basic programming skills along with some multimedia (web page) design, with a goal of developing students teamwork skills.

There is, however, no mention of a broader Computational Thinking agenda, nor any clear indication that the module is even intended to be scientific in nature. From a Computer Science perspective this curriculum is quite limited in scope (just as its title would suggest), and we would seek to develop a much broader understanding of the core principles of our discipline.

Other initiatives similar to PACT: There are at least two current initiatives in Ireland that are similar in some respects to PACT:

- The Lero group in the University of Limerick established an education and outreach programme starting in 2007, originally to support teaching using MIT's Scratch environment in Irish post-primary schools. They have developed a website[4] with support for primary and post-primary schools, including extensive lesson plans and teaching material, and run teacher training sessions as well as an annual competition.

  While the structure of the Lero initiative is similar to ours, we believe that a graphical language like Scratch is fundamentally limited in terms of teaching programming, even though it may serve as a useful introduction to some programming concepts. Further, the materials seem to be limited to teaching programming in Scratch, and do not seem to have a broader agenda in Computer Science or Computational Thinking.

- Bridge21[5] is an education programme based in Trinity College Dublin and supported by Google. It claims to be based on a new model of learning that can be adapted for use in secondary schools [5], and they provide professional development workshops for teachers to train in "21st Century Computer Science Teaching Skills".

  This initiative is directly targeting the post-primary (and Junior cycle) area, but appears to be more broadly based than our initiative, recently offering support for English, Maths and History classes. While the specific Computer Science deliverable is vague, it seem to be based at present on programming (and the Raspberry Pi), and it is not clear what their precise CS agenda is.

Even though both of these groups are established in the field, we believe that their focus in each case lacks a clear, distinctive Computer Science perspective. Our PACT initiative can fill this gap and, we believe, offers an alternative that provides a deeper insight into the fundamentals of our discipline.

[4] http://www.scratch.ie/
[5] http://www.bridge21.ie/

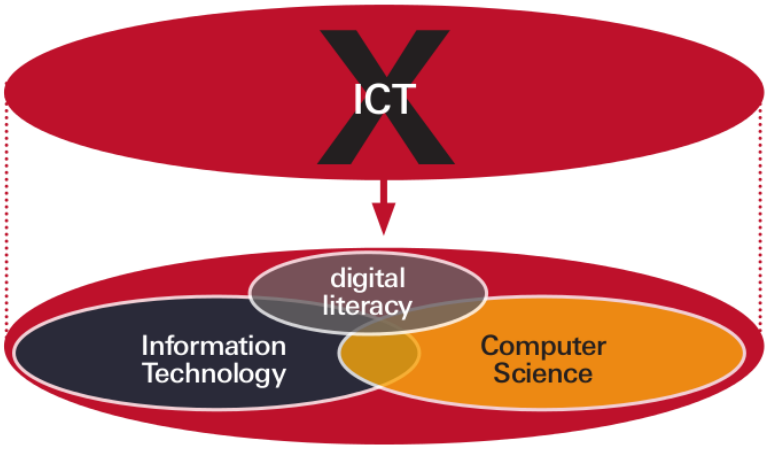

Figure 1: Replacing 'ICT': suggested terminological reform from the *Computing in Schools* initiative, emphasising the distinctive identity of Computer Science.

Recent initiatives in the UK: There has been a series of interesting related work in the UK culminating in a series of reports over the last two years.

The *Computing in Schools*[6] project was an initiative coordinated by the Royal Society in cooperation with 18 UK universities as well as several industry bodies culminating in a report published in January 2012 [9]. It notes the limitations of the previous National Curriculum in ICT, and emphasises the need to switch to 'creative, rigorous and challenging Computer Science'. It identifies that Computer Science is a rigorous academic discipline and that significant investment in continuing professional development for teachers will be needed.

Their suggested 'terminological reform' encapsulates the understood position of Computer Science from our perspective, but is nonetheless worth replicating here since it is not widely understood outside the discipline (Figure 1). While much of this document is discusses long-term funding needs and is thus directed at policy-makers, the emphasis on depth and rigour is similar in spirit to our PACT initiative.

The *Computing at School*[7] initiative by the British Computer Society has produced a curriculum for schools that has been endorsed by Microsoft, Google and Intellect an d published in March 2012 [4]. This curriculum develops a clear agenda for computing-related concepts, stratified in five stages that cover primary and post primary education. It clearly identifies Computer Science as a STEM discipline, distinguishes it from Information Technology, and emphasises core skills such as programming as well as more abstract skills such as designing algorithms and computational thinking. While this *Computing at School* is a much more ambitious agenda that the PACT initiative, it shares many of the themes and goals of PACT.

The summary of key points from a related *Computing at School* briefing note published in November 2011 [3] is worth quoting in full, since it covers many of the key points that have informed the PACT initiative (see Figure 2).

[6] https://royalsociety.org/education/policy/computing-in-schools/
[7] http://www.computingatschool.org.uk/

- It is vital to make a clear distinction between Computer Science as a rigorous subject discipline on the one hand, and IT applications and/or digital literacy on the other.
- Many countries, from the USA to India, routinely make Computer Science available to young people at school from an early age.
- There is increasing clarity that "Computer Science" means a lot more than "Learn to program in Java or C++". Programming is central to computing, but the underlying principles of algorithms, data structures, and computational thinking skills are both more fundamental and more durable.
- The confusion between Computer Science and ICT skills means that ICT is often delivered by non-specialists. Meanwhile, the continuing strong employer demand for IT professionals reduces the supply of well-qualified potential teachers. These factors conspire to mean that ICT/Computing teachers are under-valued and under-qualified. There is a desperate need for teacher training in Computer Science.

Figure 2: Key points identified by the *Computing at Schools* group based on a study of the international experience with teaching computing in schools (direct quotation from [3]).

Computer science education includes the following elements: design (both software and hardware), creation of digital artifacts, abstraction, logic, algorithm development and implementation, programming paradigms and languages, theoretical foundations, networks, graphics, databases and information retrieval, information security and privacy, artificial intelligence, the relationship between computing and mathematics, the limits of computation, applications in information technology and information systems, and social impacts of computing.

Figure 3: Elements of Computer Science education for K-12, as identified by the CSTA (direct quotation from [1]).

Recent initiatives in the US: In the US, much of the focus on K-12 CS edutaion is directed through the ACM-supported *Computer Science Teachers Association* (CSTA)[8]. In their 2010 report they identify gaps in state standards for CS education, as well as major terminological confusion between digital literacy, information technology and computer science [1]. They identify the need to establish Computer Science as an academic discipline within the STEM fields; the key topics they identify in CS education are shown in Figure 3. The CSTA has established a *Computational Thinking Task Force* to collect and disseminate teaching resources and projects in this area.

The CS10K community[9] aims to place 10,000 qualified computer science teachers into high schools to broaden access to computing education. They have developed resources in two streams: *Exploring Computer Science* covers problem

[8] http://csta.acm.org/
[9] http://cs10kcommunity.org/

solving, web design, programming (using *Scratch*) and robotics, while *Computer Science Principles* has a stronger programming and algorithms focus (and uses *App Inventor*, a Scratch-like environment for Android app development). The CS10K Project is supported by the National Science Foundation as part of its Computing Education for the 21st Century (CE21) programme.

As a further development, *Computer Science: Principles*[10] is a proposed AP course being developed by the College Board, descigned as an "introductory college-level course for everyone" [2]. It emphasises what it calls *computational thinking practices*, such as developing computational artefacts, creativity, abstraction and communication.

Each of these courses is considerably broader than that proposed by our PACT initiative, but many of the topics overlap, and it should be possible to reuse some of the rich collection of resources they have developed.

In summary, the principal initiatives from the UK, USA and other countries emphasise:

- the need to reform CS education at post-primary level,
- the need to distinguish clearly between digital literacy, information technology and CS,
- the need to teach CS concepts as a rigorous academic discipline within the STEM subjects.

4 Our work to date

During the Summer of 2013 a number of schools were approached to gauge if there was interest in running such a pilot programme. The schools which came on board were:

- Maynooth Post Primary School, Co. Kildare.
- Pobalscoil Neasain, Baldoyle, Dublin 13.
- Confey Community College, Leixlip, Co. Kildare.
- Castelcomer Community School, Co. Kilkenny.
- Scoil Mhuire Community School, Clane, Co. Kildare.
- Jesus and Mary College, Goatstown, Dublin 14.
- Salesian College, Celbridge, Co. Kildare.

At least one teacher from each of these schools agreed to participate in the pilot programme.

Teacher training sessions: The PACT team began to structure the content for the pilot programme and on the 25th of May and the 1st June 2013 two training sessions took place in NUIM. These sessions involved introducing the teachers to the content within the programme and also getting to know each other. It was hoped that a community of knowledge and learning would be created between all the teachers as they progressed with the pilot.

A Moodle system was hosted in the Department of Computer Science to store the content that was delivered in the training sessions. The content was divided in to five sections, namely:

1. Introduction to Python I.

[10] csprinciples.org

2. Introduction to Python II.
3. Algorithms.
4. Graphics.
5. Recursion and self-reference.

The content was based on the *Python* programming language[11]. *Python* is a dynamically-typed programming language that is popular in the CoderDojos as a first programming language, but also is used in a number of professional and scientific applications. The choice of programming language is not essential for the syllabus, and we envisage providing options for teachers who wish to develop using other languages such as *Processing* or *Haskell*.

Sections 1 and 2 of the training sessions covered the basics of installation of *Python* and also looked at how to create and run programs written in *Python*. They looked at the main features of the language and gave sample programs for the teachers to use. An exercise book was created for all lessons within these sections which the teacher could use in class to get their students working in *Python*. In total 19 PowerPoint presentations were prepared for these sections which the teacher could break up into whatever format they felt worked for them.

Section 3 focused on algorithms and the process of analysing problems to devise abstract computational solutions. It showed how to write a step by step procedure to solve some classic problems in Computer Science. Section 4 looked at generating simple 2-D graphics using the *Pygame* set of modules. This section focused on allowing the participants to create fully featured games in the *Python* language, and can provide a useful focus for project-based programming work. Section 5 covered topics related to recursion which has been defined as "the process of repeating items in a self-similar way" and is fundamental to the core theory of Computer Science.

Should we put in learning outcomes here??

Competition: Towards the end of the academic year 2013-14 we are going to host a competition which is open to all the students who were involved with the PACT pilot programme. This competition is aimed at providing examples of best practice and each school participating in the pilot is invited to select up to two teams, consisting of 3-5 students, to participate in the competition. The selection of teams and team members is entirely at the discretion of the schools. Participants in the competition should highlight explicitly how their experience with the PACT programme has contributed to two or more of the six key areas identified by the NCCA [6], namely:

1. Managing Myself.
2. Staying Well.
3. Communicating.
4. Being Creative.
5. Working with Others.
6. Managing Information and Thinking.

[11] https://www.python.org/

5 Future Work

As of the 1st April 2014 we have already received interest from the following schools:

- Tallaght Community School, Dublin.
- Sandford Park, Ranelagh, Dublin.
- St. Munchin's College, Limerick.
- Beneavin college, Finglas, Dublin.

These schools have approached us directly from seeing our webpage or from teachers who are involved in the pilot programme this year. We hope to run this programme again for the academic year 2014-15 with a larger number of schools and take on board the feedback received from the teachers and students in the first year.

6 Contact Us

The PACT team can be contacted at the following email address: pact@cs.nuim.ie.

The member of the Computer Science department currently involved in the PACT initiative are: Aidan Mooney (coordinator), Susan Bergin, Joe Duffin, Phil Maguire, Rosemary Monahan, Tom Naughton and James F. Power.

Our website is: http://www.cs.nuim.ie/pact.

References

[1] Computer Science Teachers Association. Running on empty: The failure to teach K12 computer science in the digital age. http://www.acm.org/runningonempty/, 2010.

[2] Owen Astrachan and Amy Briggs. The CS principles project. *ACM Inroads*, 3(2):38–42, June 2012.

[3] Computing at School Working Group. Computing at school: International comparisons. http://www.computingatschool.org.uk/data/uploads/internationalcomparisons-v5.pdf, November 2011.

[4] Computing at School Working Group. Computer science: A curriculum for schools. http://www.computingatschool.org.uk/data/uploads/ComputingCurric.pdf, March 2012.

[5] C. Conneely, D. Murchan, B. Tangney, and K. Johnston. 21 century learning - teachers and students experiences and views of the Bridge21 approach within mainstream education. In *Society for Information Technology & Teacher Education International Conference*, pages 5125–5132, March 2013.

[6] NCCA. Key skills of Junior Cycle. http://www.juniorcycle.ie/NCCA_JuniorCycle/media/NCCA/Documents/Key/Key-Skills-Overview-Feb-2013.pdf, February 2013.

[7] NCCA. Programming and coding: Draft specification for Junior Cycle short course. http://www.juniorcycle.ie/Curriculum/Consultation/Short-Courses/Programming-and-Coding, October 2013.

[8] Seymour Papert. An exploration in the space of mathematics educations. *International Journal of Computers for Mathematical Learning*, 1(1):95–123, 1996.

[9] The Royal Society. Shut down or restart? the way forward for computing in uk schools. http://royalsociety.org/education/policy/computing-in-schools/, January 2012.

[10] Jeannette M. Wing. Computational thinking. *Communications of the ACM*, 49(3):33–35, March 2006.