Computing Education 2010

Proceedings of the
Twelfth Australasian Computing Education Conference
(ACE 2010), Brisbane, Australia,
January 2010

Tony Clear and John Hamer, Eds.

Volume 103 in the Conferences in Research and Practice in Information Technology Series.
Published by the Australian Computer Society Inc.

Published in association with the ACM Digital Library.

Conferences in Research and Practice in Information Technology, Volume 103.

Copyright ©2010, Australian Computer Society. Reproduction for academic, not-for-profit purposes permitted provided the copyright text at the foot of the first page of each paper is included.

Editors:

Tony Clear
School of Computing and Mathematical Sciences
Auckland University of Technology
Private Bag 92006
Auckland 1020, New Zealand
Email: tony.clear@aut.ac.nz

John Hamer
Department of Computer Science
University of Auckland
Private Bag 92019
Auckland, New Zealand
Email: j.hamer@auckland.ac.nz

Series Editors:
Vladimir Estivill-Castro, Griffith University, Queensland
Simeon J. Simoff, University of Western Sydney, NSW

crpit@scm.uws.edu.au

Publisher: Australian Computer Society Inc.
PO Box Q534, QVB Post Office
Sydney 1230
New South Wales
Australia.

Conferences in Research and Practice in Information Technology, Volume 103.
ISSN 1445-1336.

Printed, December 2009 by UWS Press, Locked Bag 1797, South Penrith DC, NSW 1797, Australia
Document engineering by Susan Henley, University of Western Sydney
Cover Design by Matthew Brecknell, Queensland University of Technology

The Conferences in Research and Practice in Information Technology series aims to disseminate the results of peer-reviewed research in all areas of Information Technology. Further details can be found at http://crpit.com/.
Table of Contents

Proceedings of the Twelfth Australasian Computing Education Conference (ACE 2010), Brisbane, Australia, January 2010

Preface .................................................................................................................. vii
Programme Committee ....................................................................................... viii
Organising Committee ........................................................................................ ix
Welcome from the Organising Committee ....................................................... x
CORE - Computing Research & Education ...................................................... xi
ACSW Conferences and the Australian Computer Science Communications .............................................. xii
ACSW and ACE 2010 Sponsors ............................................................................. xiv

Invited Papers

Textbooks: How We Choose Them, How We Use Them, Shall We Lose Them? .................. 3
   Nell B. Dale

Does Quality Assurance Enhance the Quality of Computing Education? ......................... 9
   Arnold N. Pears

Computer Science in New Zealand High Schools .................................................. 15
   Tim Bell, Peter Andreac and Lynn Lambert

Panel

Internationalisation and Cross Cultural Issues in Computing Education .................. 25
   Kathryn Egea, Jie Lu, Jitian Xiao and Tony Clear

Contributed Papers

Introductory Programming and the Didactic Triangle ............................................ 35
   Anders Berglund and Raymond Lister

An experience report on using collaboration technologies for distance and on-campus learning ...... 45
   David Carrington, Soon-Kyeong Kim and Paul Strooper

Study Habits of CS 1 Students: What do they do outside the classroom? .................... 53
   Donald Chinn, Judy Sheard, Angela Carbone and Mikko-Jussi Laakso

Engaging Students in Programming ...................................................................... 63
   Malcolm Corney, Donna Teague and Richard N. Thomas
Constructive Controversy as a Way to Create True Collaboration in an Open Ended Group Project .......................... 73
Mats Daniels and Asa Cajander

Introductory Programming in a Web Context .................................................. 79
Michael de Raadt

Approaches used by cross-cultural and cross-discipline students in teamwork for a first-year course in web design ................................................................. 87
Kathryn Egea, Soon-Kyeong Kim, Trish Andrews and Karin Behrens

Effects of Course-Long Use of a Program Visualization Tool ..................................... 97
Erkki Kaila, Teemu Rajala, Mikko-Jussi Laakso and Tapio Salakoski

The case for ICT work-integrated learning from graduates in the workplace ................ 107
Tony Kopp, Sylvia Edwards, Judy Sheard, Fazel Naghdy and Wayne Brooks

Cross-Cultural Education: Learning Methodology and Behaviour Analysis for Asian Students in IT Field of Australian Universities .................................................. 117
Jie Lu, K. L. Chin, Juan Yao, Jun Xu and Jitian Xiao

Student Perceptions of ICT: A Gendered Analysis ............................................. 127
Christine McLachlan, Annemiek Craig and Jo Coldwell

The Quality of a PeerWise MCQ Repository ..................................................... 137
Helen Purchase, John Hamer, Paul Denny and Andrew Luxton-Reilly

Using a primary-school challenge in a third-year IT course .................................. 147
Simon

Improving the Learning of Graduate Attributes in the Curriculum: a Case-Study in IT Management ............................................................ 155
Alan Sixsmith and Andrew Litchfield

An Adaptable Framework for the Teaching and Assessment of Software Development across Year levels 165
Richard Thomas, Moira Cordiner and Diane Corney

Author Index ........................................................................................................... 173
Preface

Welcome to the Twelfth Australasian Computing Education Conference (ACE2010). This year, the ACE2010 conference, which is part of the Australasian Computer Science Week, is being held in Brisbane Australia from 18-21 January, 2010.

We again see a strong international presence at the conference with 51 authors coming from Canada, Finland, New Zealand, Scotland, Sweden, United States and Australia. The Chairs would like to thank the Program Committee for their excellent efforts in the double-blind reviewing process which resulted in the selection of 14 full papers from the 30 papers submitted, giving an acceptance rate of 47%.

The keynote speakers this year are hosted by the other ACSW conferences, in line with the policy that the plenary sessions will rotate between the ACSW conferences each year. We are lucky though to have the services of three international invited speakers for this year’s ACE conference. Nell Dale from Texas will talk on the topic of CS textbooks, and as a prolific textbook author herself we are lucky to have her share her insights. Arnold Pears from Sweden will address the topic of Quality Assurance in Computing Education and some current conundrums. Tim Bell from New Zealand will profile some promising new developments in the high school computing curriculum in New Zealand.

We have a very diverse set of topics this year. Papers and presentations include collaboration technologies and Web 2.0, models and pedagogical frameworks for computing education, studies of novice programming students, student motivations and perspectives, the use of technology in computing education, course content, curriculum structure, methods of assessment, web development, online learning, and work-integrated learning through to graduate attributes. The high quality papers this year continue to push the frontiers of opportunities for research and innovation in computing education, and this conference will enable these educators to meet and share their experiences in a new forum. We will be holding a Panel on Internationalisation in Computing Education, profiling institutional developments, a major ALTC funded study into international students in Australian Universities and a critical perspective on Internationalisation and the ‘Export Education' industry.

In keeping with the ACE tradition, there will be conference workshops, two prior to the conference and one at the end of the conference continuing to build research and expertise in computing education in Australasia. The BRACElet workshop this year is thankful for the kind sponsorship from the Australian Council for Learning and Teaching through their fellowship to Raymond Lister and Jenny Edwards.

We are grateful to SIGCSE for sponsoring the Conference jointly with the ACM. We thank everyone involved in Australasian Computer Science Week for making this Conference and Proceedings publication possible, and we thank CORE, our hosts Queensland University of Technology, Brisbane, and the Australasian Computing Education Executive for the opportunity to chair this ACE2010 Conference.

Tony Clear  
Auckland University of Technology, New Zealand

John Hamer  
University of Auckland

ACE 2010 Programme Chairs  
January 2010
Programme Committee and Additional Referees

Chairs

Tony Clear, Auckland University of Technology, New Zealand
John Hamer, University of Auckland, New Zealand

Members

Alison Young, CPIT, New Zealand
Angela Carbone, Monash University, Australia
Margaret Hamilton, RMIT University, Australia (senior co-chair)
Judy Kay, University of Sydney, Australia
Judy Sheard, Monash University, Australia
Josh Tenenberg, University of Washington, USA
Mats Daniels, Uppsala University, Sweden
Michael de Raadt, University of Southern Queensland, Australia
Raymond Lister, University of British Columbia, Canada
Sally Fincher, University of Kent, UK
Simon, University of Newcastle, Australia

Additional Reviewers

Arnold Pears, Uppsala University, Sweden
Jacqueline Whalley, AUT University, Auckland
Gwyn Claxton, AUT University, Auckland
Graham Bidois, AUT University, Auckland
Gordon Grimsey, AUT University, Auckland

Conference Webmaster

Graham Bidois, AUT University, Auckland
Organising Committee

Co-Chairs
Dr. Wayne Kelly
Prof. Mark Looi

Budget and Facilities
Mr. Malcolm Corney

Catering and Booklet
Dr. Diane Corney

Sponsorship and Web
Dr. Tony Sahama

Senior Advisors
Prof. Colin Fidge
Prof. Kerry Raymond

Finance and Travel
Ms. Therese Currell
Ms. Carol Richter

Registration
Mr. Matt Williams

DVD and Signage
Mr. Matthew Brecknell

Satchels and T-shirts
Ms. Donna Teague
Welcome from the Organising Committee

On behalf of the Australasian Computer Science Week 2010 (ACSW2010) Organising Committee, we welcome you to this year’s event hosted by the Queensland University of Technology (QUT). Striving to be a “University for the Real World” our research and teaching has an applied emphasis. QUT is one of the largest producers of IT graduates in Australia with strong linkages with industry. Our courses and research span an extremely wide range of information technology, everything from traditional computer science, software engineering and information systems, to games and interactive entertainment.

We welcome delegates from over 21 countries, including Australia, New Zealand, USA, Finland, Italy, Japan, China, Brazil, Canada, Germany, Pakistan, Sweden, Austria, Bangladesh, Ireland, Norway, South Africa, Taiwan and Thailand. We trust you will enjoy both the experience of the ACSW 2010 event and also get to explore some of our beautiful city of Brisbane. At Brisbane’s heart, beautifully restored sandstone buildings provide a delightful backdrop to the city’s glass towers. The inner city clusters around the loops of the Brisbane River, connected to leafy, open-skied suburban communities by riverside bikeways. QUT’s Garden’s Point campus, the venue for ACSW 2010, is on the fringe of the city’s botanical gardens and connected by the Goodwill Bridge to the Southbank tourist precinct.

ACSW2009 consists of the following conferences:

- Australasian Computer Science Conference (ACSC) (Chaired by Bernard Mans and Mark Reynolds)
- Australasian Computing Education Conference (ACE) (Chaired by Tony Clear and John Hamer)
- Australasian Database Conference (ADC) (ADC) (Chaired by Heng Tao Shen and Athman Bouguettaya)
- Australasian Information Security Conference (AISC) (Chaired by Colin Boyd and Willy Susilo)
- Australasian User Interface Conference (AUI) (Chaired by Christof Lutteroth and Paul Calder)
- Australasian Symposium on Parallel and Distributed Computing (AusPDC) (Chaired by Jinjun Chen and Rajiv Ranjan)
- Australasian Workshop on Health Informatics and Knowledge Management (HIKM) (Chaired by Anthony Maeder and David Hansen)
- Computing: The Australasian Theory Symposium (CATS) (Chaired by Taso Viglas and Alex Potanin)
- Asia-Pacific Conference on Conceptual Modelling (APCCM) (Chaired by Sebastian Link and Aditya Ghose)
- Australasian Computing Doctoral Consortium (ACDC) (Chaired by David Pearce and Rachel Cardell-Oliver).

The nature of ACSW requires the co-operation of numerous people. We would like to thank all those who have worked to ensure the success of ACSW2010 including the Organising Committee, the Conference Chairs and Programme Committees, our sponsors, the keynote speakers and the delegates. Special thanks to Justin Zobel from CORE and Alex Potanin (co-chair of ACSW2009) for his extensive advice and assistance. If ACSW2010 is run even half as well as ACSW2009 in Wellington then we will have done well.

Dr Wayne Kelly and Professor Mark Looi
Queensland University of Technology
ACSW2010 Co-Chairs
January, 2010
CORE welcomes all delegates to ACSW2010 in Brisbane. CORE, the peak body representing academic computer science in Australia and New Zealand, is responsible for the annual ACSW series of meetings, which are a unique opportunity for our community to network and to discuss research and topics of mutual interest. The original component conferences ACSC, ADC, and CATS, which formed the basis of ACSW in the mid 1990s, now share the week with seven other events, which build on the diversity of the Australasian computing community.

In 2010, we have again chosen to feature a small number of plenary speakers from across the discipline: Andy Cockburn, Alon Halevy, and Stephen Kisely. I thank them for their contributions to ACSW2010. I also thank the keynote speakers invited to some of the individual conferences. The efforts of the conference chairs and their program committees have led to strong programs in all the conferences again, thanks. And thanks are particularly due to Wayne Kelly and his colleagues for organising what promises to be a strong event.

In Australia, 2009 saw, for the first time in some years, an increase in the number of students choosing to study IT, and a welcome if small number of new academic appointments. Also welcome is the news that university and research funding is set to rise from 2011-12. However, it continues to be the case that per-place funding for computer science students has fallen relative to that of other physical and mathematical sciences, and, while bodies such as the Australian Council of Deans of ICT seek ways to increase student interest in the area, more is needed to ensure the growth of our discipline.

During 2009, CORE continued to work on journal and conference rankings. A key aim is now to maintain the rankings, which are widely used overseas as well as in Australia. Management of the rankings is a challenging process that needs to balance competing special interests as well as addressing the interests of the community as a whole. ACSW2010 includes a forum on rankings to discuss this process. Also in 2009 CORE proposed a standard for the undergraduate Computer Science curriculum, with the intention that it be used for accreditation of degrees in computer science.

CORE’s existence is due to the support of the member departments in Australia and New Zealand, and I thank them for their ongoing contributions, in commitment and in financial support. Finally, I am grateful to all those who gave their time to CORE in 2009; in particular, I thank Gill Dobbie, Jenny Edwards, Alan Fekete, Tom Gedeon, Leon Sterling, and the members of the executive and of the curriculum and ranking committees.

Justin Zobel
President, CORE
January, 2010
ACSW Conferences and the Australian Computer Science Communications

The Australasian Computer Science Week of conferences has been running in some form continuously since 1978. This makes it one of the longest running conferences in computer science. The proceedings of the week have been published as the *Australian Computer Science Communications* since 1979 (with the 1978 proceedings often referred to as Volume 0). Thus the sequence number of the Australasian Computer Science Conference is always one greater than the volume of the Communications. Below is a list of the conferences, their locations and hosts.

2011. Volume 33. Host and Venue - Curtin University of Technology, Perth, WA.

2010. Volume 32. Host and Venue - Queensland University of Technology, Brisbane, QLD.


2008. Volume 30. Host and Venue - University of Wollongong, NSW.

2007. Volume 29. Host and Venue - University of Ballarat, VIC. First running of HDKM.

2006. Volume 28. Host and Venue - University of Tasmania, TAS.


2001. Volume 23. Hosts - Bond University and Griffith University (Gold Coast), Venue - Gold Coast, QLD.


1998. Volume 20. Hosts - University of Western Australia, Murdoch University, Edith Cowan University and Curtin University. Venue - Perth, WA.


1995. Volume 17. Hosts - Flinders University, University of Adelaide and University of South Australia. Venue - Glenelg, SA.


1990. Volume 12. Host and Venue - Monash University, Melbourne, VIC. Joined by Database and Information Systems Conference which in 1992 became ADC (which stayed with ACSW) and ACIS (which now operates independently).

1989. Volume 11. Host and Venue - University of Wollongong, NSW.


1987. Volume 9. Host and Venue - Deakin University, VIC.

1986. Volume 8. Host and Venue - Australian National University, Canberra, ACT.


1983. Volume 5. Host and Venue - University of Sydney, NSW.

1982. Volume 4. Host and Venue - University of Western Australia, WA.

1981. Volume 3. Host and Venue - University of Queensland, QLD.

1980. Volume 2. Host and Venue - Australian National University, Canberra, ACT.

1979. Volume 1. Host and Venue - University of Tasmania, TAS.

1978. Volume 0. Host and Venue - University of New South Wales, NSW.
### Conference Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Conference Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACDC</td>
<td>Australasian Computing Doctoral Consortium</td>
</tr>
<tr>
<td>ACE</td>
<td>Australasian Computer Education Conference</td>
</tr>
<tr>
<td>ACSC</td>
<td>Australasian Computer Science Conference</td>
</tr>
<tr>
<td>ACSW</td>
<td>Australasian Computer Science Week</td>
</tr>
<tr>
<td>ADC</td>
<td>Australasian Database Conference</td>
</tr>
<tr>
<td>AISC</td>
<td>Australasian Information Security Conference</td>
</tr>
<tr>
<td>AUIC</td>
<td>Australasian User Interface Conference</td>
</tr>
<tr>
<td>APCCM</td>
<td>Asia-Pacific Conference on Conceptual Modelling</td>
</tr>
<tr>
<td>AusPDC</td>
<td>Australasian Symposium on Parallel and Distributed Computing (replaces AusGrid)</td>
</tr>
<tr>
<td>CATS</td>
<td>Computing: Australasian Theory Symposium</td>
</tr>
<tr>
<td>HIKM</td>
<td>Australasian Workshop on Health Informatics and Knowledge Management</td>
</tr>
</tbody>
</table>

Note that various name changes have occurred, which have been indicated in the Conference Acronyms sections in respective CRPIT volumes.
ACSW and ACE 2010 Sponsors

We wish to thank the following sponsors for their contribution towards this conference.

CORE - Computing Research and Education,
www.core.edu.au

CEED,
www.corptech.com.au

CSIRO ICT Centre,
www.csiro.au/org/ict.html

SAP Research,
www.sap.com/about/company/research

Association for Computing Machinery,
www.acm.org

ACM Special Interest Group on Computer Science Education,
www.sigcse.org

The Commonwealth Scientific and Industrial Research Organisation,
www.csiro.au

Australian Learning and Teaching Council,
www.altc.edu.au
INVITED PAPERS
Computer Science in New Zealand High Schools

Tim Bell
Department of Computer Science and Software Engineering
University of Canterbury
Christchurch, New Zealand
tim.bell@canterbury.ac.nz

Peter Andreae
School of Engineering and Computer Science
Victoria University of Wellington
New Zealand
Peter.Andreae@ecs.vuw.ac.nz

Lynn Lambert
Physics, Computer Science and Engineering Department
Christopher Newport University
Newport News, Virginia, USA
llambert@cnu.edu

Abstract
The New Zealand Ministry of Education has recently released a new “Digital Technologies” proposal for delivering computing topics in the final three years of High Schools. The proposal aims to address a number of issues by offering topics that will be academically challenging for students, and provide them with a broader view of the kinds of advanced topics they might study beyond High School. The proposed structure includes having Digital Technologies as a separate area in the technology curriculum, and includes a strand called “Computer Science and Programming” that has sufficient coverage to communicate to students what the subject area is really about.

This paper reviews the circumstances that led to this proposal, describes the international context (especially in the US) for High School computing curricula, and examines the published proposal in some detail. It also considers the issues that are likely to come up in the implementation of the proposal, and how they might be addressed.

Keywords: Computer Science curriculum.

1 Introduction
A secondary school Computer Science curriculum can have a significant influence on student career paths, both for laying the groundwork for further study, but more importantly, for exposing students to the topic. The latter is particularly valuable because the discipline of Computer Science is not well understood by High School students, who often make career choices based on an inaccurate perception and bad experiences unrelated to the topic itself (Margolis and Fisher 2002).

In recent times, New Zealand schools have rarely taught Computer Science – at best there have been courses on programming at some schools, but often computing education has been focused on general purpose applications and skills. Even worse, sometimes courses that teach “computing as a tool” have given students the impression that CS must be an extension of these topics. Of course, it is important for students to be able to use computers effectively, but often this has been a distraction from getting students involved in “computing as a discipline”.

For a period (1974-1985) computing was taught as a discipline in NZ schools through “Applied Maths”. However, there have been several changes since then, and recently assessment for computing courses that go
beyond just using applications have typically been via “unit standards”, which are pass/fail skills-based standards that are not attractive to top students who would like to get high grades to reflect their academic achievements.

In addition to these issues, NZ had adopted a technology curriculum that provided generic assessment tools for teaching areas ranging from food technology to digital technology, which meant that computing wasn’t a subject area of its own, and the same kind of assessment criteria would be used for a large range of technologies (such as meal planning and software development). Because computing was combined with other technologies, it was less accessible as a discipline in its own right, and less attractive to students who were specifically interested in the discipline of computing.

Some progress had been made towards a broader and deeper approach to the curriculum, such as the “Fluency in IT” project (Clear and Bidois 2005), but mapping such proposals onto the new national technology curriculum was proving to be problematic.

In 2008, two reports were released that very clearly pointed out the weaknesses of the current offerings in schools, and called for action from the government agencies that set the standards used to determine the courses delivered by schools (Grimsey and Phillipps 2008; Carrell, Gough-Jones, and Fahy 2008).

This resulted in the Ministry of Education calling together a “Digital Technologies Experts Panel” (DTEP) representing industry, tertiary and High Schools, to develop a plan to address the issues raised in these two reports. The panel first met in November 2008, and by mid 2009 it had produced a body of knowledge and recommendations for a way forward.

The DTEP recommendations were released in May 2009, along with an agreement negotiated with the Ministry of Education for moving forward. The agreement included:

- A specific area called “Digital Technologies” within the technology learning area.
- An aggressive timeline for implementation, with guidelines to be made available to schools in 2010, so that courses in this area could be run from 2011.
- A body of knowledge in five strands of Digital Technologies (see below).

The DTEP report also mentions the following, which seem likely to be achieved by the changes planned:

- Better alignment of school material to tertiary and industry expectations of areas such as Computer Science.
- Assessment standards that are academically challenging, and help students to meet entrance standards for tertiary study.
- Use of ICT-related terminology in schools that reflects the usage in industry and tertiary organisations.
- Urgent professional development for teachers.

Considerable work had already been done in previous years on “Digital Technology Guidelines” (DTG), and the DTEP recommendations included adapting these guidelines to match the recommended body of knowledge, rather than starting from scratch. This meant that the new material could be delivered taking advantage of the existing momentum achieved by the DTG.

The new proposal (called “Technological Context Knowledge and Skills”) was posted publicly for comment in August 2009. At the time of writing, the plan is being updated to take account of feedback, and the version reported on here is the August 2009 version. The public discussions of the proposal indicate that it is being well received by industry, tertiary institutions, and schools, which bodes well for its implementation.

If the new proposal is successful, it should inspire and prepare students to contribute to the growth of New Zealand’s economy through innovative work based on good foundations in subject knowledge, and give students an understanding of the different areas in ICT so that students can make sensible choices. It should also address the issue of computing not being regarded as having a high academic status.

---

1 The full version is available from http://www.techlink.org.nz/curriculum-support/tks/

2 http://dtg.tki.org.nz/

in schools, and so should attract more high academic achievers into the field.

Section 2 reviews some of the international work on the development of CS curricula that constitutes a context for the new NZ guidelines; section 3 describes the proposed NZ learning area for Digital Technologies in general, and section 4 focuses on the proposed Computer Science topics within Digital Technologies. The issues surrounding implementation of the new guidelines are discussed in section 5, and we draw conclusions in section 6.

2 CS in school curricula

Computing in school curricula is often diluted because it has to cover three quite different directions: (1) using computers as a tool for teaching (e.g. e-learning), (2) using computers as a tool for general purpose applications (sometimes called ICT), and (3) computing as a discipline in its own right (including programming and CS). Sometimes administrators and leaders confuse these roles, and this can make it difficult for Computer Science to be visible as a discipline in its own right.

Although computing as a tool is commonly taught around the world, relatively few countries have a significant CS curriculum, and even fewer make such a curriculum mandatory for schools (Ragonis 2007).

In the United States, there is no federal organization that guides the curriculum of Computer Science. The main federal law regarding education in the United States is the “No Child Left Behind” act. This law states that all teachers must be highly qualified except teachers in non-core areas (Wilson and Harsha 2009). The non-core areas are: Physical Education, Computer Science, and vocational education. There are several implications of this: first, Computer Science is not considered a core area, so schools tend to emphasize it less than core areas (the states receive funding based on how well they are doing in core areas). Second, it is up to the 50 states to implement Computer Science programmes. Thus, there is no single technology or Computer Science program in the United States, but many.

Although there are no federal guidelines and differing state requirements, several different organizations have developed technology or computer guidelines for the United States. The Computer Science Teachers Association (CSTA) standards differentiate technology and using computers in the support of education in general from the field of Computer Science in its model K-12 Computer Science curriculum (Tucker et al. 2006). The International Society for Technology in Education (ISTE) has developed technology standards called NETS, National Education Technology Standards4. The National Association for Educational Progress (NAEP) is developing technology literacy guidelines5 that will become part of the nation’s report card in 2012, although these are not specifically computing technology. With no federally enforced standard, the de facto standard for High School curricula are Advanced Placement exams – of 23,000 high schools nationally, 17,000 offer some AP courses. AP Computer Science, which has the same curriculum and test across the country, is almost exclusively programming (currently in Java). A group of prominent Computer Science educators is attempting to redesign the AP curriculum to create a course that is less centred on programming (Cuny 2009).

Many of the difficulties implementing effective computing curricula are common to a number of countries, and the NZ experience has reflected the experience of others, including the rapid decline in tertiary CS enrolments after the year 2000 (Vegso 2008). The recent developments in NZ appear to have addressed many of the issues raised, and no doubt there will be lessons to learn and new material developed as the new proposal is implemented in schools.

3 New Digital Technology Guidelines

The proposed guidelines for Digital Technology in NZ schools are given in a “Technological Context Knowledge and Skills” (TCKS) document released by the NZ Ministry of Education.

4http://www.iste.org/AM/Template.cfm?Section =NETS

5 A draft is available at http://www.edweek.org/media/nagb_assessment _devel_comm_aug_7-09.pdf
Education (Dinning 2009). The guidelines have five “contexts” for Digital Technologies:

- Digital Information (digital tools and systems for managing information),
- Digital Infrastructure (hardware and networks, including installing software),
- Digital media (video, audio, layout/design, web, graphics, animation, games, web),
- Electronics (electronic and embedded systems), and
- Programming and Computer Science (concepts from CS and Software Engineering, designing and implementing programs).

These five areas address a range of interests and career paths that students might take, and are aimed at the final three years of high school.

Each area contains a set of objectives specifying the different aspects of knowledge and skill in the area. Each objective is broken down into three levels (6, 7 and 8) of the NZ curriculum, which would correspond to the last three years of High School for most students (years 11 to 13) with a list of “indicators” that give more details of the objective for each level.

In addition to these subject-specific guidelines there are supplementary “generic” achievement standards on “Technological practice”, “Nature of technology” and “Technological knowledge”, which would be available to assess topics such as the history of computing, the effect of digital technology on society, or project management.

If the proposed guidelines are accepted, then the ministry will develop a set of achievement standards addressing the knowledge and skills described by the indicators in the document, and guidelines for teachers. The way the curriculum works in NZ, schools will then be free to make up courses that are built around their own selection of achievement standards, based on local interests and strengths. A likely outcome in many schools is that a year 11 student may study an introduction to several of the five areas in a single course, whereas at year 13 schools might offer a whole course based on just one or two of them.

Some of the “contexts” (especially Digital Information and Digital Media) will be more concerned with the computer as a tool, but the others are more concerned with the computer itself. In the following section we will look in detail at the “Programming and Computer Science” context.

4 CS in the new guidelines

What had previously been just “programming” in the existing Digital Technology Guidelines now appears as “Programming and Computer Science”, reflecting a concern for giving students a broader basis for making an informed decision about possible paths in the tertiary sector. The first of the three objectives in the published proposal addresses this broadening:

“Demonstrate an understanding of concepts across Computer Science and Software Engineering.”

The indicators for the above objective introduce the fundamental concepts of algorithms and programming languages at level 6. At level 7, these are expanded to include four further important concepts:

- The ideas of complexity/tractability/computability – that some problems are inherently difficult or impossible to solve on a computer.
- Coding (e.g., compression, error correction, encryption), and how it has enabled new technologies.
- That programming languages are specified precisely.
- The need for Software Engineering methodologies, and an appreciation of the steps in the Software Development Life cycle.

The indicator at Level 8 is broader, and allows a selection of topics from across Computer Science and Software Engineering.

The wording of (and the examples in) the indicators make it clear that the goal is an appreciation of what Computer Science and Software Engineering are about, and not an in-depth understanding of the content of the topics. For example, the concepts of complexity/tractability/computability at level 7 are intended only to have the students understand that some problems are very difficult to solve, no matter how clever the algorithm, and that some problems are impossible. It would not be appropriate at this level for the students to have to determine the complexity class of a problem or construct a proof of intractability, but they might appreciate
that binary search is significantly better than linear search, even if only for searching a telephone book. Students who gain an appreciation of these concepts will know that Computer Science is more than just programming and will be much better placed to start a tertiary qualification in Computer Science – or to decide that Computer Science is not what they want to study!

The other two objectives for the Programming and CS context address two complementary aspects of programming. One addresses the design of programs:

“Be able to understand, select and design data types, data structures, algorithms, and program structures for a program to meet specified requirements, and evaluate user interfaces.”

The other addresses the processes for actually constructing programs:

“Be able to read, understand, write, and debug software programs using an appropriate programming language, tools, and software development process.”

Although these two aspects would almost certainly be intertwined in teaching practice, distinguishing designing from constructing emphasises that programming involves understanding and design at a more abstract level, as well as knowing the technical details of a programming language and being able to read and write programs in that language. There is also a practical advantage in distinguishing them, at least for assessment, in that weaker students who cannot cope with the design aspects may still be able to pass achievement standards addressing the practical skills of reading, understanding, writing and debugging programs if they are given sufficient guidance and support on the design aspect.

The design objective also includes a component on evaluating user interfaces. Although actually designing a good user interface is a more advanced topic than is appropriate at school level, it is quite feasible for students to analyse existing interfaces from a design perspective. Because most students will already have used a wide variety of interactive programs in a range of contexts, including multiple programs for the same task (e.g., mobile phones, browsers, mail clients, picture viewers, DVD players), there will be plenty of options for getting students to compare, evaluate, and suggest improvements to existing user interfaces. At level 6, such evaluation would be strictly informal; at level 7, it would be based on lists of usability heuristics, and level 8 would use a wider range of Human-Computer Interaction (HCI) principles. Having students look at this aspect of computing means that they can broaden their view to appreciate the broader skills and knowledge (such as psychology for HCI or linear algebra for graphics) that are valuable for constructing successful digital systems.

Both programming objectives have a sequence of indicators at the three levels that build up from having students develop very simple programs. At level 6, the indicators require only programs that use variables, expressions, selection, and loops, and the primitive data types available in the chosen language. This is sufficiently simple that popular introductory languages (such as Scratch or Alice) could be used to assess them if delivered appropriately. Level 7 extends the indicators to include methods (or procedures/functions/subroutines) and compound data structures (e.g., arrays, or lists); level 8 adds the use of data from files and procedures with parameters and return values. The higher levels are likely to require a conventional general purpose language to cover the concepts listed (e.g. Java, Python, or Visual Basic).

Level 8 also adds a greater understanding of data types, with an appreciation of the properties and limitations of different data types. This might include an understanding of different ways of representing numeric data (binary, hexadecimal, fixed point, floating point, etc.) but it could equally be addressed in the context of representing textual data or image data. For many students, understanding different ways of representing pixel colour values for images may provide better motivation than traditional scientific calculation.

The proposal makes no mention of topics such as classes, packages, inheritance, or exceptions, and does not require programs to have graphical user interfaces. Of course, with some languages or program development tools,
students may be exposed to programs using such concepts, but they are not required.

No particular programming language is specified, and teachers could use any appropriate language they wish as long as it has sufficient constructs to cover the structures required. At level 6, a very wide range of languages would be possible, including domain or application specific languages, as long as they supported programming with variables, expressions, selection, and loops. At level 7, the selected language would need to support procedural abstraction. At level 8, a general purpose programming language is required, along with an appropriate software development environment.

The indicators also require some documentation in the students’ programs, but this is at the level of choosing good names, suitable layout, and using some appropriate comments. For the small programs that the students would be able to construct, elaborate documentation is not only unnecessary, but is generally unmotivating. Level 7 also introduces testing.

At Level 8, the indicator for the program construction objective requires some level of discipline in the programming process, with some problem analysis for simple requirements, and the use of a simple software development process. In general, this would be a simplified version of an agile process, emphasising repeated cycles with increasing requirements and careful testing against the requirements at each stage. The goal is not to gain a mastery of software engineering practice, but to become aware of the need for discipline in the programming process, and to appreciate the role of aspects such as requirement specification, testing, and debugging, in addition to the actual writing of program code.

5 Implementation

The proposed changes have a number of implementation challenges and implications for teacher training and student career paths.

Because the proposal introduces some topics not previously taught in schools, most teachers will require professional development to be able to teach them. For many, this may mean learning to program, and even for some of those who are comfortable teaching programming, it will require gaining some familiarity with the overview of Computer Science topics. Fortunately the new standards will be introduced over a period of four years, which gives teachers some time to get up to speed. During this time, there will need to be extensive communication with teachers so that they can keep in touch with developments, be aware of strong support for the transition, and feel engaged in the process.

New material will be needed for teaching topics that haven’t existed before in schools. The Computer Science academic community in NZ is offering extensive support to develop material and help with professional development. This material will be published online, so it can be used by the international community.

If the new guidelines bring about the hoped-for growth in the discipline, more teachers will need to be found to deliver the resulting classes. This in turn will require more training opportunities, particularly in the Colleges of Education where teachers receive their key qualifications. For new teachers, College of Education courses will need to expand to cover the new topics proposed. Some teachers who are already in service, but are not currently involved in computing, might elect to retrain in these areas, in which case distance-learning or flexible courses may be more accessible. Finally, students and graduates with a background in Computer Science might be recruited to become teachers and bring their expertise to the classroom, given appropriate training to qualify them for the role.

There will be implications for tertiary institutions because the students leaving school may now have more advanced knowledge and experience in the discipline. While some schools may not offer the full range of standards proposed, others could potentially have students graduating who are competent programmers in a language such as Java or Python, and who have a reasonable understanding of algorithm analysis, including simple searching and sorting algorithms. For some tertiary institutions, this may represent a similar standard to their first-year Computer Science courses, and they may need to consider alternative paths for such students. In either case, students who have done well in these areas can be identified and
potentially offered scholarships or more challenging courses.

In addition, students will need guidance on what other subjects to take at school. For example, Computer Science departments would typically be looking for students with strong mathematics and communication skills, and structures may need to be put in place to ensure that students are aware of the importance of these complementary subjects for success in their chosen career path. This will include communicating the new pathways to career counsellors and advisors.

6 Conclusions

New Zealand is on the verge of delivering an exciting programme for computing as a discipline in High Schools. The current design reflects enough technical material that students can get some insight into the career paths available to them, and also provide academic challenges to make the courses attractive to top students.

Implementing such changes requires a lot of careful design and testing, particularly for topics that have never been taught before. Teachers will need considerable help to become comfortable with new topics, and given the fast time scale, support from the tertiary community will be essential.

It is important that the change is not seen as a bigger list of things to learn in less time. The new topics are generally at the level of exposing students to them so they have an appreciation of their significance. More important than the details of what is taught is that students know what career paths are available, and are able to get a sufficiently accurate taste of the discipline to find out if it suits them or not, rather than making up their mind based on incorrect information and mislabelled topics.

Once the context and skills outlined above have been finalised, they will guide the creation of guidelines for teachers, and standards for assessment. This process will need to be complete enough by early 2010 that schools can plan and publish their Year 11 (level 6) programmes for students making decisions about their 2011 courses. Despite this rapid pace, the pipeline is three years long, and the first students to leave school having completed the new programmes will not be entering the tertiary institutions (or the workforce) until 2014, so it will be some time before the effects of the changes are fully realised.

7 Acknowledgement

We are grateful to an anonymous referee for detailed feedback and suggestions.

8 References


Tucker, A. (editor), Deek, F., Jones, J., McCowan, D., Stephenson, C., and Verno, A.
Author Index

Andreae, Peter, 17
Andrews, Trish, 89

Behrens, Karin, 89
Bell, Tim, 17
Berglund, Anders, 37
Brooks, Wayne, 109

Cajander, Åsa, 75
Carbone, Angela, 55
Carrington, David, 47
Chin, K. L., 119
Chinn, Donald, 55
Clear, Tony, 27
Clear, Tony, iii
Coldwell, Jo, 129
Cordiner, Moira, 167
Corney, Diane, 167
Corney, Malcolm, 65
Craig, Annemieke, 129

Dale, Nell B., 3
Daniels, Mats, 75
de Raadt, Michael, 81
Denny, Paul, 139

Edwards, Sylvia, 109
Egea, Kathryn, 27, 89
Hamer, John, iii, 139
Kaila, Erkki, 99

Kim, Soon-Kyeong, 47, 89
Koppi, Tony, 109

Laakso, Mikko-Jussi, 55, 99
Lambert, Lynn, 17
Lister, Raymond, 37
Litchfield, Andrew, 157
Lu, Jie, 27, 119
Luxton-Reilly, Andrew, 139

McLachlan, Christine, 129

Naghdy, Fazel, 109
Pears, Arnold N., 9
Purchase, Helen, 139
Rajala, Teemu, 99
Salakoski, Tapio, 99
Sheard, Judy, 55, 109
Simon, 149
Sixsmith, Alan, 157
Strooper, Paul, 47

Teague, Donna, 65
Thomas, Richard, 65, 167

Xiao, Jitian, 27, 119
Xu, Jun, 119
Yao, Juan, 119
Recent Volumes in the CRPIT Series

ISSN 1445-1336

Listed below are some of the latest volumes published in the ACS Series *Conferences in Research and Practice in Information Technology*. The full text of most papers (in either PDF or Postscript format) is available at the series website http://crpit.com.

Volume 103 - Computing Education 2010

Contains the proceedings of the 3rd Australasian Ontology Workshop (AOW 2007), Gold Coast, Queensland, Australia.

Volume 102 - Artificial Intelligence and Data Mining 2009

Contains the proceedings of the 2nd International Workshop on Integrating AI and Data Mining (AIDM 2007), Gold Coast, Australia. December 2007.

Volume 101 - Data Mining and Analytics 2009

Contains the proceedings of the 8th Australasian Data Mining Conference (AusDM 2009), Melbourne Australia.

Volume 100 - Safety Critical Systems and Software 2009

Contains the proceedings of the 13th Australian Conference on Safety Critical Systems and Software, Canberra Australia.

Volume 99 - Grid Computing and e-Research 2009

Contains the proceedings of the Australasian Workshop on Grid Computing and e-Research (AUCHE2009), Wellington, New Zealand, January 2009.

Volume 98 - Information Security 2009

Contains the proceedings of the Australian Information Security Conference (AISC2009), Wellington, New Zealand, January 2009.

Volume 97 - Health Data and Knowledge Management 2009

Contains the proceedings of the Third Australasian Workshop on Health Data and Knowledge Management (HHRM 2009), Wellington, New Zealand, January 2009.

Volume 96 - Conceptual Modelling 2009

Contains the proceedings of the Fifth Asia-Pacific Conference on Conceptual Modelling (APCCM2008), Wollongong, NSW, Australia, January 2008.

Volume 95 - Computing Education 2009

Contains the proceedings of the Eleventh Australasian Computing Education Conference (ACE2009), Wellington, New Zealand, January 2009.

Volume 94 - Theory of Computing 2009


Volume 93 - User Interfaces 2009

Contains the proceedings of the Tenth Australasian User Interface Conference (AUIUC2009), Wellington, New Zealand, January 2009.

Volume 92 - Database Technologies 2009

Contains the proceedings of the Twenty-first Australasian Database Conference (ACSC2009), Wellington, New Zealand, January 2009.

Volume 91 - Computer Science 2009

Contains the proceedings of the Thirty-Second Australasian Computer Science Conference (ACSC2009), Wellington, New Zealand, January 2009.

Volume 90 - Advances in Ontologies 2009

Contains the proceedings of the Knowledge Representation Ontology Workshop (KROW 2008), Sydney, September 2008.

Volume 89 - Australian Video 2009

Contains the proceedings of the 1st Australian Video Conference.

Volume 88 - Koli Calling 2007

Contains the proceedings of the 7th Baltic Sea Conference on Computing Education Research.

Volume 87 - Data Mining and Analytics 2008

Contains the proceedings of the 7th Australasian Data Mining Conference (AusDM 2008), Adelaide, Australia. December 2008.


Volume 85 - Advances in Ontologies 2007

Contains the proceedings of the 3rd Australasian Ontology Workshop (AOW 2007), Gold Coast, Queensland, Australia.

Volume 84 - Artificial Intelligence and Data Mining 2007

Contains the proceedings of the 2nd International Workshop on Integrating AI and Data Mining (AIDM 2007), Gold Coast, Australia. December 2007.

Volume 83 - Advances in Ontologies 2007

Contains the proceedings of the 3rd Australasian Ontology Workshop (AOW 2007), Gold Coast, Queensland, Australia.

Volume 82 - Artificial Intelligence and Data Mining 2007

Contains the proceedings of the 2nd International Workshop on Integrating AI and Data Mining (AIDM 2007), Gold Coast, Australia. December 2007.