A Preliminary Phenomenographic Study Concerning Student Experiences of Unix

Byline—researc
undertaken a
student and sta
membe
respectivel

Bernard Doyle
Faculty of Information Technology,
University of Technology, Sydney, Australia
bjd@it.uts.edu.au

Raymond Lister
Faculty of Information Technology,
University of Technology, Sydney, Australia
raymond@it.uts.edu.au

Abstract

This preliminary report is based upon the experiences of six students who had recently completed their first semester of studying Unix. A phenomenographic analysis of the interview transcripts identified four categones of how students experienced Unix. One of the categories is Unix as a resource, in which the student focuses on characteristics of Unix such as its cost, vulnerability to attack, robustness, and load capacity. The other three categories focus on the direct user experience of Unix. These three categories form an outcome space that is linear and hierarchical. Those three categories are, from lowest to highest: Unix as a set of commands, Unix as a tool for solving certain problems, and Unix as a professional computing environment. For this outcome space, there are indications of a direct relationship between the category most prominently manifested in each student's interview transcript and the student's final mark in the Unix course. There are also indications of a similar relationship between the outcome space and the student's performance on the R-SPQ-2F test for deep and surface learning.

Keywords: Computing education, Phenomenography, Deep and Surface Learning, Unix.

1 Introduction

Many students experience introductory Unix courses as a tedious and pointless chore. While the Unix command line is a powerful mechanism for getting a lot of work done with an economy of key strokes, a number of factors are a formidable barrier to student learning. These factors include: student perceptions of Unix as irrelevant in the real world; the irregularities inherent in Unix commands; and the use of an unfamiliar non-GUI mode of human-computer interaction. After studying Unix, many students subsequently minimise their use of it and prefer to use Microsoft Windows (henceforth, referred to simply as "Windows").

Unix in its many forms (e.g. Linux) is a very important part of the internet and the computing world in general. It is not in the best interests of students, or society, that many studentts who graduate with IT degrees are not comfortable with Unix.

In this paper, we report on a preliminary phenomenographic study of six studenis who had recently completed their first semester of studying Unix. The aim of this study is to identify the ways in which students experience Unix. Future reports of this study will drive a redesign of the teaching of Unix, with the aim of facilitating a more positive way of experiencing Unix.

2 The Learning Environment for Unix

Before describing the phenomenographic study, we briefly describe the environment at the authors' university in which the students learn Unix. In response to poor student response to learning Unix, but prior to the commencement of this phenomenographic study, some teaching changes had already been made with the aim of improving student attitudes to Unix, and encouraging a deep approach to learning. The basis for these changes was the intuition of the teachers.

2.1 Minilectures

Students are given a series of minilectures (15 minutes) intended to counter negative student impressions of Unix. These minilectures describe the widespread use of Unix in commercial IT. An historical perspective on Unix development is given, to explain the origins of the syntactic problems and inconsistencies with the Unix command line. The minilectures also discuss the strengths and weaknesses of both the command line and graphical user interfaces.

2.2 Discussion Groups

Students are encouraged to use online discussion groups, where they are able to discuss their problems with Unix and offer each other solutions. Most students took advantage of this facility.

2.3 Independent Learning

Students were encouraged to setup their own Linux system at home. For the obvious reason of security, students were not given root access to the machines at the university. By setting up their own Linux system at home, they had total freedom to explore Unix. It is not known how many students took up this option.

This quality assured paper appeared at the *19th Annual Conference of the National Advisory Committee on Computing Qualifications* (NACCQ 2006), Wellington, New Zealand. Samuel Mann and Noel Bridgeman (Eds). Reproduction for academic, not-for profit purposes permitted provided this text is included. www.naccq.ac.nz

3 Method and Motivation

3.1 Phenomenography

Phenomenography is a research technique that has been used to examine the ways in which students experience their learning. Phenomenographic researchers typically gather data by interviewing a number of human participants about a particular phenomenon. The phenomenographer then analyses the transcripts, looking for critical variations in the way the interviewees articulate their experience of the phenomenon. In many past phenomenographic studies, the data from interviewees has been classified into a small number of qualitatively different ways of experiencing the phenomenon. For example, across many interviews of students, in many studies, students have been found to adopt a small number of qualitatively different approaches to their study, including the well-known deep and surface approaches (Ramsden, 1992; Biggs, 1987 & 1999). We discuss deep and surface learning at greater length below. Also, in phenomenographic studies of teachers, a small number of qualitatively different ways of experiencing teaching have been identified, including (1) the teacher-centred, transmission-oriented experience of teaching and (2) the student-focussed experience of teaching (Trigwell, 2000).

Phenomenography has been used to study student learning across many disciplines, including statistics (Reid and Petocz, 2003), physics (Booth and Ingerman, 2002), and music (Reid, 1997). Within computing, the seminal phenomenographic work was by Booth (1992). She studied the experience of students who were learning to program. There has since been more work on programming (Bruce *et al.*, 2003; Stoodley, Christie, and Bruce,. 2004, Eckerdal and Thun, 2005). Other computing phenomenographic studies not concerned with programming have included studies of students learning about information systems, (Cope, 2002) the TCP protocol (Bruce, Buckingham, Hynd, McMahon, Roggenkamp, and Stoodley, 2004) and object-orientation (Box & Lister, 2005). At least one phenomenographic study in computing education has focussed on the teachers – a study of lecturers' understandings of the purpose of teaching elementary data structures (Lister, Box, Morrison, Tenenberg, and Westbrook, 2004).

Prior to this study, there has not been a phenomenographic study of students learning Unix. There has been at least one non-phenomenographic study of students learning Unix (Greening, 1996).

Why study (via phenomenography or any other method) how students experience their learning? For their study of students who were learning statistics, Reid and Petocz (2003) offered the following justification, which applies equally to the learning and teaching of Unix:

> "*... students are generally unaware of the range of variation in their fellow students' conceptions, and making them aware of this range gives them the opportunity to broaden their views ... this is commonly done and successfully done in professional development courses with university lecturers ... it seems reasonable that university students would also benefit from such an exploration. ...* [also] *... there are strong relations between teachers' conceptions of their discipline and the way that they go about teaching. ... the learning environments that* [teachers] *set up in their class can encourage those students who identify with the lower, fragmented levels* [of experiencing the content] *to engage with their learning at a higher level.*" (p. 292)

3.2 Deep and Surface Learning

The authors' long term goal is to foster a deep approach to student learning of Unix, and to discourage surface learning. The concepts of deep and surface learning are now well entrenched in texts read by academics seeking to improve their teaching (e.g. Ramsden, 1992; Biggs, 1999). A deep approach to learning in a student is characterised by: (1) the desire to understand, (2) attempts to relate what is learnt to existing knowledge, (3) attempts to see what is learnt as a coherent body of knowledge, and (4) trying to see beyond the immediate goal of exams and other assessment. In contrast, surface learning by a student is characterised by: (1) rote learning of facts, (2) not relating what is learnt to pre-existing knowledge, (3) learning facts without attempting to look for relationships between those facts, and (4) the primary learning goal is to pass the assessment.

The interviewees were asked to complete the R-SPQ-2F survey (Biggs, Kember, and Leung, 2001). This is a questionnaire developed to measure the extent to which a student has adopted a deep or surface approach when studying a particular subject. Our intent is to validate the R-SPQ-2F survey within the context of learning Unix, by comparing it to our analysis of the interview transcripts. If the survey can be validated, then future innovations in our teaching might be more economically assessed via a class survey, rather than via interviews.

3.3 Interviewee Background

The first author interviewed six students who had just completed a semester of learning about Unix. Out of a total enrolment of 95 students, only these 6 volunteered to be interviewed. Four of the interviewees came from a non-English speaking background.

The students had varying backgrounds with Unix, and computing in general, prior to starting the course. Three of the interviewees had some systems administration experience, but primarily for Windows systems. One interviewee had administration experience exclusively on Windows systems. Another interviewee had limited exposure to system administration with Unix, while the third interviewee had extensive exposure to Windows system administration and limited exposure to Unix system administration. The remaining three interviewees had very little prior exposure to Unix but were quite comfortable with using Windows.

3.4 Interview Structure

The interviews were semi-structured, with the following question set formulated prior to the interviews being conducted:

1. Tell me about your experience with Unix prior to this subject.
2. What does the word "Unix" mean to you?
3. What do you think is the role of Unix in this subject?
4. What do you think is the role of Unix in this degree?
5. What do you think is the role of Unix in computing in general?
6. What do you think is the role of an operating system, whether it be Unix, Windows, or any other operating system?
7. What do you understand by the term "script"?
8. In what ways are scripting languages and application level programming languages different?
9. In what ways are they the same?
10. What do you understand by the term "file system"?
11. What do you understand by the term "pipe"?

The interviewer routinely diverged from the above set of questions, to pursue interesting issues that arose during the interviews.

4 Results and Analysis

4.1 Overview

We analysed the data using established phenomenographic techniques. Åkerlind (2005) described the variations in phenomenographic practice. In her terms, our analysis used the following variations: (1) we considered excerpts from transcripts, (2) the two authors analyzed the data jointly, (3) our analysis focussed primarily on attempting to resolve mismatches between our respective initial, independent interpretations, (4) while the structure of our analysis did emerge from the data, it was also driven by structural regularities we had observed in earlier phenomenographic studies (see section 5.2) and (5) our focus is on pragmatic validity (i.e. the value of the analysis for providing insight into the teaching and learning of Unix).

From our preliminary analysis, we have identified four ways in which the six interviewees experience Unix:

- Unix as a set of commands
- Unix as a tool for solving certain problems
- Unix as a professional computing environment
- Unix as a resource.

These four ways of experiencing Unix are described in the next four subsections, with quotes from interviewees to illustrate each way of experiencing Unix. Each quote is followed by a code in square brackets, which identifies the interviewee.

4.2 Unix as a set of commands

One of the six interviewees articulated his experience spoke of Unix as a weakly related set of commands:

> "*Some of the topics* [in Unix] *were quite interesting, but the actual topic area itself is a little dry. After all it's just a series of commands at the end of the day.*" [S04]

The interviewee expressed a general preference for GUI interfaces over command line interfaces:

> "*... on the user side of things,* [Unix is] *not that great yet.... as opposed to say, Windows. ... sometimes the GUI is there* [in Unix] *but you couldn't do everything you wanted to.*" [S04]

4.3 Unix as a tool for solving certain problems

Students articulating this experience of Unix characterised Unix as a general purpose operating system available as an alternative to other operating systems. They appeared comfortable with the Unix command line interface, and were oriented towards using Unix as a tool in particular situations:

> "*You can use just one line commands to do a lot of stuff that you have to do using excel spreadsheet. Using one line you can do in 10, 20 or 50 steps using a spreadsheet.*" *... It has so many utilities that allow the system administrator to write just one line of commands to do a lot of stuff. If you want to search for something, in Windows you have to do maybe ten steps to do that. In Linux just one line of commands will do the stuff that you want. This is a powerful thing in Unix.* " [S03]

> "*The file structure is very flexible. One of the powerful things is text processing, scripting with the command line of Unix. ... Linux is not just a command line, but it can also support a very powerful graphic vision.*" [S01]

> "*Probably the most basic task which I crave in a Windows machine now is the powerful* [Unix] *command line. For moving round, copying data., ... I find it extremely great to be able to go in with vi, via the command line, and just go in, quickly locate something in the HTML, if I've got a spelling error, or something, manipulate the file, and log out of the server.*" [S05]

Students who strongly articulated this experience saw Unix and Windows as systems with differing strengths and weaknesses. Unix was seen as a particularly useful tool for solving some problems.

4.4 Unix as a professional environment

Students articulating this experience of Unix saw it as being a superior way to organise and use a computer:

> "*I think Unix and Linux is more powerful. .... It's more professional than Microsoft ... I discovered a new world of computing in Unix. ... I changed my study direction after a few hours* [of commencing study of Unix] *... Even though I have finished the subject, I still study more about Unix. I usually come here in the evenings to study Unix. The history, read some article, some things that have not been taught to me in this course.*" [S01]

> "*I am a systems administrator, and I used to use Microsoft based, and we have, you know, so many problems, with Microsoft, if you are a system administrator. Unix now, opens, I think a new track for me, to deal with system administration using Unix, and now I am planning to study Unix system*

> *administration next semester, because I would like to be a Unix systems administrator. Because I like ...* [Unix] *... so much. I think it's powerful, and will develop my future career in systems administrator.*" [S03]

In the case of student S03, quoted above, we note that his experience of Unix may be primarily formed as a reaction to his encounters with Windows prior to studying Unix, and not primarily as a result of his Unix learning experience.

4.5 Unix as a Resource

This experience of Unix is quite different from the previous three experiences. Unlike the earlier experiences, this experience makes no reference to the command line, or system architecture of Unix. Instead, the focus is on properties of Unix discernible even to someone with a non-technical background – its cost, its speed, its robustness, and the high level of security:

> "*For a server operating system, the best is Unix. If you want to host a website for the internet, you need a powerful server, that can hold the website, and which will have the minimum downtime. I think the downtime for Unix is less than other operating systems downtime.*" [S03]

> "*For Microsoft it's easier for the programmers to develop some programs like viruses and Trojan horses to attack the system. For Unix for the system administrators it's hard to develop viruses and Trojan ... it's more secure.*" [S03]

> "*With a Unix machine I can successfully set up a web server that can handle thousands of users at one time.*" [S05]

> "*... you can download* [a version of Unix] *from anywhere and it's a free download ... Whereas if you are wanting to play around with a Novell Server or a MS Windows server, the licences are two to three thousand.*" [S05].

> "*I did research on Google ... it also said FreeBSD is good as well... it's not really a bad system for implementing a firewall. ... Also, it was free.*" [S06]

4.5. The Outcome Space

In the previous four sections, we have described four ways, or categories, in which the interviewees experienced Unix. We now discuss relationships between these categories. In phenomenographic terms, the relationships between categories form an outcome space.

Outcome spaces are often linear and hierarchical. The higher categories subsume the lower categories. A higher category is a more sophisticated understanding than a lower category.

Three of the four experiences of Unix form such a hierarchy. At the lowest level of the hierarchy is the experience of Unix as a set of commands with little connection between them, followed by Unix as a tool for solving certain problems, with the most sophisticated experience being Unix as a professional computing environment. As we move through those three categories, from lowest to highest, the level of technical sophistication increases, and there is a greater degree of integration of various Unix skills and knowledge.

At this stage of the project, it appears that students do not connect the category "Unix as a resource" to the other three categories. For example, the students appear to see the superior security of Unix as an "accidental" property of Unix, not a consequence of the architecture of Unix. Perhaps, as we collect more interview transcripts, we will see students who do articulate such a connection. On the other hand, perhaps such a connection is not currently being articulated by the teachers.

4.6 Individuals and Categories

On the basis of quotes read out of context, such as the quotes given above, interviewees cannot be assigned to a single phenomenographic category. An inspection of the interview quotes given above demonstrates that, for example, quotes from interviewees S01 and S03 are used to illustrate both the category "Unix as a tool for solving certain problems" and the category "Unix as a professional computing environment". Since these two categories are part of a hierarchy, an interviewee who manifests the higher category will also manifest the lower category. In general, if an individual is shown the categories generated from phenomenographic research, they will identify with more than one category, to varying degrees.

However, it is possible to read the entire transcript of an interviewee and identify the highest category of a linear hierarchical outcome space that is articulated prominently by an interviewee. For the linear, hierarchical outcome space identified in this paper, we have performed such a whole-of-transcript analysis. Table 1 shows the categories to which we assigned the six interviewees. In the next section, we compare those assignments to the interviewees' scores on the R-SPQ-2F questionnaire.

4.7 The R-SPQ-2F Questionnaire

The R-SPQ-2F questionnaire is a points system for assessing the extent to which a student manifests a deep and/or surface approach to the learning of a given topic. It contains 20 statements, such as:

- *My aim is to pass the course while doing as little work as possible.*
- *I find most new topics interesting and often spend extra time trying to obtain more information about them.*
- *I learn some things by rote, going over and over them until I know them by heart even if I do not understand them.*
- *I work hard at my studies because I find the material interesting.*

The six interviewees were required to respond to these questions, as they applied to their study of Unix, on the following 5-point Likert scale:

- Never or only rarely true of me
- Sometimes true of me
- True of me about half the time
- Frequently true of me
- Always or almost always true of me

Each of the above responses is assigned a point value, from one to five, in the above order of responses. From a student's responses to the questionnaire, using the above response point scoring system, two scores are constructed for that student. One score is an index of the extent to which the student manifests a deep approach to learning. The other score is an index of the extent to which the student manifests a surface approach to the learning. For both indices, a student scores between 10 and 50; a score of 30 being neutral, lower scores being weaker and higher scores being stronger in the approach.

Table 1 summarises data for all six interviewees. For each student, the table shows the relationships between: (1) The student's final assessment mark for their study of Unix (a higher percentage is better); (2) the authors' whole-of-interview categorisation of the student, as described in the previous subsection, and (3) the student's deep and surface learning indices derived from the R-SPQ-2F questionnaire. The table is ordered according to the students' assessment mark. With only six interviewees, we make no claims of there being statistically significant trends in the relationships. However, these preliminary results suggest that there is a positive relationship between a student's assessment mark, the student's dominant experience of Unix as manifested in the interview, and their scores in the R-SPQ-2F questionnaire.

5. Reliability and Validity

This section discusses the degree of confidence we can have in the preliminary results reported in this paper.

5.1 Number of Interviewees

It is possible to perform a preliminary phenomenographic analysis with only six interviewees. Phenomenography is a qualitative research method, not a quantitative method. From the data presented, it would not be appropriate to speculate upon the frequency among students of any of the above categories. To make such conclusions would require significantly more data and a different research method. The aim of phenomenographic research is merely to capture the full spectrum of diversity, not quantify it.

We make no claim to have identified, at this stage, the full spectrum of diversity in student experiences of Unix. Even for a phenomenographic study, six interviewees is a relatively low number of data sources. However, phenomenographers routinely collect and analyze data concurrently, ceasing to collect data when they establish saturation – when several consecutive interviews do not lead to the identification of new categories (and the outcome space suggests that there are no missing categories in the existing data). With only six interviews we do not claim to have reached saturation – this paper is a report on a work in progress. Interviewing more subjects may elaborate but not invalidate this preliminary study. That is, interviewing more students will probably add more categories, and add further structure to the outcome space, but collecting more interviews is unlikely to invalidate the categories and outcome space we have identified in this paper, though it is possible. Having collected a subset of the data we will eventually collect, we believe our existing analysis captures a valid, possible subspace of the outcome space we will eventually construct.

5.2 Relationship to other computing studies

Another source of confidence in this preliminary analysis is its consistency with past phenomenographic studies in computing. We see a pattern emerging from those past studies, which is also present in our study. In phenomenographic computing studies that identify linear, hierarchical outcome spaces, there are usually either three categories, or the categories form three groups. In the lowest category (or group of categories), students focus on syntactic/notational/mechanical issues. In the next category (or group of categories) students focus on problem solving. In the highest category (or group of categories) students focus on social and/or professional issues. For example, both Booth (2001) and Berglund (2005) identify three categories that form a linear hierarchical space consistent with this general description (see Table 2).

6. Conclusion

Using the phenomenographic method of analysis, we have found categories describing the student experience of Unix. Three of these categories form a linear hierarchy, which is broadly consistent with the findings of some past phenomenographic studies on other aspects of computing. We also note that our analysis of the interviews triangulates well with the student scores on the R-SPQ-2F questionnaire. This paper is only a preliminary report, and more data needs to be collected and analysed before final conclusions are drawn. However, the consistency between this study, past studies, and the R-SPQ-2F questionnaire suggest that while we may not as yet have identified the whole picture, we have at least identified a self-consistent piece of that whole picture.

In future work, we will be continuing the same form of analysis with a larger pool of students. We will also be conducting similar interviews with academics, systems programmers and people who use Unix in their employment, so we can compare their conceptions of unix to the conceptions of our students.

7 References

Åkerlind, G. (2005) Variation and commonality in phenomenographic research methods. *Higher Education Research & Development.* **24**(4): 321-334.

Berglund, A. (2005) *Learning computer systems in a distributed project course: The what, why, how and where.* Acta Universitatis Upsaliensis, Uppsala Dissertations from the Faculty of Science and Technology 62. ISBN 91-554-6187-5.

Biggs, J. (1987) *Student approaches to learning and studying.* Australian Council for Educational Research.

Biggs, J. (1999) *Teaching for quality learning at university: what the student does.* Open University Press.

Biggs, J, Kember, D, and Leung, D. (2001) The Revised two-factor Study Process Questionnaire: R-SPQ-2F.

British Journal of Educational Psychology, **71**, 133-149.

Booth, S. (1992). *Learning to program: A phenomenographic perspective.* PhD thesis, University of Gothenberg, Sweden.

Booth, S. (2001) Learning to Program as Entering the Datalogical Culture: a Phenomenographic Exploration. In *9th European Conference for Research on Learning and Instruction (EARLI),* Fribourg, Switzerland.

Booth, S. and Ingerman, A. (2002) Making sense of Physics in the first year of study. *Learning and Instruction* **12:** 493-507

Box, I., & Lister, R. (2005). Variation in students' conceptions of object-oriented information system development. In O. Vasilecas, A. Caplinskas, W. Wojtkowski, W. G. Wojtkowski, J. Zupancic & S. Wrycza (Eds.), *Information systems development: Advances in theory, practice and education. Proceedings of 13th international conference on information systems development (ISD 2004) Vilnius, Lithuania, September 9–11 2004* (pp. 439–451): Springer.

Bruce, C, Buckingham, L, Hynd, J, McMahon, C, Roggenkamp, M, and Stoodley, I. (2004) Ways of Experiencing the Act of Learning to Program: A Phenomenographic Study of Introductory Programming Students at University. *J. of Information Technology Education,* **3**: 143-159. http://jite.org/documents/Vol3/v3p143-160-121.pdf [accessed May 2005]

Cope, C. (2002) Seeking Meaning: The Educationally Critical Aspect of Learning About Information Systems, *Proceedings of the Informing Science + IT Education Conference, Cork, Ireland.*http://proceedings.informingscience.org/IS2002Proceedings/papers/cope190seeki.pdf [accessed Apr. 2006]

A. Eckerdal, A. & Thun, M. (2005) Novice Java Programmers' Conceptions of "Object" and "Class", and Variation Theory. *Proceedings of the 10th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education,* pp. 89-93.

Greening, T (1996) Teaching and learning essential computer science skills: the UNIX example. *ACM SIGCSE Bulletin,* **28**(2): 21-24.

Lister, R., Box, I., Morrison, B., Tenenberg, J., Westbrook, S. (2004) The Dimensions of Variation in the Teaching of Data Structures. *9th Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE),* Leeds, UK, 28-30 June. pp. 92-96.

Ramsden, P. (1992) *Learning to Teach in Higher Education.* London Routledge (0-415-06415-5)

Reid, A. (1997), The Meaning of Music and the Understanding of Teaching and Learning in the Instrumental Lesson, in *Proceedings of the Third Triennial ESCOM Conference,* ed. A. Gabrielsson, Uppsala, Sweden: European Society for the Cognitive Sciences of Music, 3, 200-205.

Reid, A. and Petocz, P. (2003) Completing the Circle: Researchers of Practice in Statistics Education. *Mathematics Education Research J.,* **15**(3): 288-300.

Stoodley, I. Christie, R. and Bruce, C. (2004) Masters Students' Experiences of Learning to Program: An Empirical Model. *Proceedings of QualIT2004: International Conference on Qualitative Research*

Trigwell, K. (2000) *Phenomenography: Discernment and Variation* http://www.learning.ox.ac.uk/files/... ...Phenom_ISL_paper.pdf [accessed Mar. 2006]

Table 1: The relationships between assessment mark, categorisation by interview, and R-SPQ-2F indices.

Student	Assessment Mark (%)	Highest Experience of Unix Manifested Strongly in the Interview	Deep Learning Index	Surface Learning Index
S04	46	Unix as a set of commands	29	26
S02	76	Unix as a tool for solving certain problems	38	34
S05	80	Unix as a tool for solving certain problems	24	25
S01	96	Unix as a professional computing environment	37	15
S03	98	Unix as a professional computing environment	41	20
S06	100	Unix as a professional computing environment	40	21

Table 2: The linear, hierarchical, three-layer outcome space identified in this paper, and in earlier studies.

General Category	Student Experience of Unix (This paper)	Learning to Program (Booth, 2001)	TCP (Berglund, 2005)
Syntactic/Notational/ Mechanical	A weakly related set of commands.	Learning a programming language	Safe communication between two computers
Problem Solving	A tool for solving certain problems.	Learning to solve problems in the form of programs	A connection across a network
Social/Professional	A professional computing environment.	Learning as becoming part of the programming community.	A standard for communication

Proceedings of the Nineteenth Annual Conference of the National Advisory Committee on Computing Qualifications

edited by

Dr Samuel Mann

Dr Noel Bridgeman

7th - 10th July 2006

NACCQ

FOSTERING COMPUTING EDUCATION IN NEW ZEALAND

In cooperation with

acm Association for Computing Machinery
Advancing Computing as a Science & Profession

ISSN

ISSN 1176-8053

NACCQ

The National Advisory Committee on Computing Qualifications (NACCQ) was formed in 1988, and provides help and support to member institutions in the development, teaching and quality assurance of courses and programmes, from certificate through degree to postgraduate levels, in the field of Computing and Information Technology. The NACCQ also supports academic staff in several different ways, such as national workshops, research seminars, APL, moderation, the publishing of a journal, a research bulletin and also by organising this annual conference. NACCQ activities and a digital version of these proceedings are available from http://www.naccq.ac.nz

The philosophy of the conference is the encouragement and support of new, emerging and established researchers in a safe environment while encouraging excellence and academic discourse.

National Advisory Committee on Computing Qualifications
PO Box 11 050
Hillcrest, Hamiliton, New Zealand
Telephone 07-856 1916 Facsimile 07-856 1912
Email viviennew@wave.co.nz

The conference and these proceedings are an output of the NACCQ Research and Support Working Group.

Quality Assured

All full papers in this proceedings were double-blind refereed by at least three reviewers and then selected by an editorial panel. 17% of submitted papers were accepted for publication with a further 29% accepted after significant revision.

Peer revie

Reviewers:

Frina Alertyn	Eastern Institute of Technology
Dr Clare Atkins	Nelson Marlborough Institute of Technology
Bridie Atkins	Universal College of Learning
David Bremer	Otago Polytechnic
Dr Noel Bridgeman	Unitec New Zealand
Tony Clear	Auckland University of Technology
Dr Brian Cusack	Auckland University of Technology
Bob Gibbons	Manukau Institue of Technology
Patricia Haden	Otago Polytechnic
Dr Donald Joyce	Unitec New Zealand
Dr Donald Koh	Unitec New Zealand
Dr Xiaosong Li	Unitec New Zealand
Mike Lopez	Manukau Institue of Technology
Dobrilla Lopez	Manukau Institue of Technology
Dr Samuel Mann	Otago Polytechnic
Mike Mullany	Northland Polytechnic
Trevor Nesbit	Christchurch Polytechnic Institute of Technology
John Paynter	Auckland University
Krassie Petrova	Auckland University of Technology
Karen Phillips	Bay of Plenty Polytechnic
Garry Roberton	Waikato Institute of Technology
Janne Ross	Christchurch Polytechnic Institute of Technology
Lesley Smith	Otago Polytechnic
Michael Verhaart	Eastern Institute of Technology
Malcolm Weick	Christchurch Polytechnic Institute of Technology

Editors

Dr Samuel Mann
Associate Professor
Department of Information Technology
Otago Polytechnic, Dunedin, NZ
smann@tekotago.ac.nz

Dr Noel Bridgeman
Senior Lecturer
Director, CITRUS Research and Development Centre
School of Computing and Information Technology
Unitec, NZ
nbridgeman@unitec.ac.nz

Theme

The conference contains papers in the following areas:

- computing education
- computing practice
- computing research

The theme this year is Preparing for the future: Capitalising on IT.

The cover of the proceedings celebrates the inclusion in the conference of a Perspectives on IT Careers forum aimed to link stakeholders to ensure the continued availability of a highly skilled workforce to sustain a vibrant and healthy Information Technology industry in New Zealand.

The illustration is by Serge Ratten (rattesc1@tekotago.ac.nz) from a script by Samuel Mann.

Full paper

Full papers are double blind peer refereed on submission by a review panel and accepted/modified/rejected. The editorial panel reviews final versions. They may be rejected or returned for modification at that point.

Peer review

Full papers may be either long (8 pages) or short (4 pages) with no qualitative distinction.

Poster and demonstration papers

A 915*600mm poster is displayed at the conference along with one page in the proceedings. Research primarily carried out by students is identified with an NACCQ Capstone Research logo.

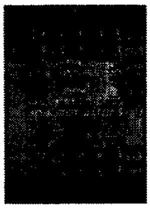
Online

This conference proceedings, along with those from recent years, can be found online at www.naccq.ac.nz From 2006, the proceedings will also appear in the ACM Digital Library.

2000: Wellington

2001: Napier

2002: Hamilton



2003: Palmerston North

2004: Christchurch

2005: Tauranga

2006: Wellington

Quality assured papers

ToC

Poster papers

Quality assured papers