Full paper

Eight years of computing education papers at NACCQ

Simon
University of Newcastle, Australia
simon@newcastle.edu.au

Judy Sheard
Monash University, Australia
judy.sheard@infotech.monash.edu.au

Angela Carbone
Monash University, Australia
angela.carbone@sims.monash.edu.au

Michael de Raadt
University of Southern Queensland, Australia
deraadt@usq.edu.au

Margaret Hamilton
RMIT University, Melbourne, Australia
margaret.hamilton@rmit.edu.au

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

Errol Thompson
New Zealand
kiwiet@acm.org

Abstract

The 157 computing education papers from the past eight NACCQ conferences are categorised and summarised by a group of researchers from multiple institutions, with steps taken to measure and improve the consistency of classification. The papers are set predominantly in programming subjects, hardware/architecture/systems/ network subjects, and capstone projects. The bulk of the papers are about teaching/learning techniques, assessment techniques, teaching/learning tools, curriculum, and educational technology. Most of the papers are set within single subjects, a few in multiple subjects within a single program or department, and fewer still in a range of subjects across the whole institution or multiple institutions. Nearly a quarter of the papers either expound a position or outline a proposal; a large but diminishing proportion report on something such as a change of curriculum or approach; and a large and increasing proportion are clearly research papers, focusing on the analysis of data to answer an explicit research question.

Keywords: Computing education, literature.

This quality assured paper appeared at the 21st *Annual Conference of the National Advisory Committee on Computing Qualifications* (NACCQ 2008), Auckland, New Zealand. Samuel Mann and Mike Lopez (Eds). Reproduction for academic, not-for-profit purposes permitted provided this text is included. www.naccq.ac.nz

1 Introduction

Now in its twenty-first year, NACCQ is the prime computing education conference in New Zealand. While it includes papers on non-educational research topics, a large proportion of its papers deal with either research or practice in computing education.

Simon (2007) devised a system for classifying computing education publications, and applied the scheme to computing education papers published in the preceding three years at NACCQ and at ACE (the Australasian Computing Education conference).

We have applied Simon's system to the past eight years of NACCQ papers, working as a group to try to offset any bias that might spring from individual classification. Because the system was devised specifically for computing education papers, we have not applied it to non-education papers from the conference.

The NACCQ proceedings distinguish between 'full papers' and other papers, some of which are called 'concise papers' and some 'poster papers'. There are also sometimes keynote papers.

We have considered only the full papers, as they are the papers that have generally come through a thorough peer review. Not all of the web-based proceedings explain the different types of paper; but, for example, the 2003 proceedings indicate that "Full papers are 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." On the other hand, "Concise papers are reviewed on abstract by a review panel. The completed papers are reviewed by the editorial panel and may be rejected or returned for modification at that point." (Mann & Williamson 2003).

Acknowledging that the word 'paper' can mean either an academic publication or a unit of classroom teaching, we shall use the word only for the former concept, and shall use the word 'subject' for a unit of teaching.

Our classification of these computing education papers is intended to be exploratory. We aim to form an overview of the papers from the past eight conferences. We have no preconceptions as to what we might find, and thus no research hypothesis. We do not suggest that papers in one category are inherently superior to papers in another. Our research question is 'What can we learn about the computing education papers published at NACCQ from 2000 to 2007, and how can we summarise what we learn?'

2 Simon's classification

Simon's system classifies a paper according to four distinct aspects. The *context* describes what sort of subject a paper is set in; the *theme* describes what the paper is about; the *scope* gives some sense of the breadth of the work; and the *nature* describes what sort of paper it is, in an expansion of the notional divide between practice papers and research papers.

In the remainder of this section, the references are to NACCQ papers that provide examples of the categories being mentioned.

2.1 Context

At first sight, many computing education papers appear to be about first-year programming, or operating systems, or database design, etc. On closer inspection, though, the papers are not actually about those topics. Rather, they are set in subjects in which those topics are taught. The *context* dimension of the classification system captures this notion. Most computing education papers are clearly set in an identifiable subject, such as information systems (Toki 2000), webpage development (Li 2007), or software engineering (Surendran & Young 2000). Others, such as this paper, are set in the literature. Still others are set in no particular subject (Holt 2006), or in a range of different subjects (Tupu, Ngatuere, & Young 2004).

A paper's classification in the context dimension will therefore be a specific subject area, literature, no context, or multiple contexts.

2.2 Theme

The *theme* of a paper is what it is actually about. For example, a paper in the context of a programming course might be about a particular assessment technique (Plimmer 2000), a tool to assist with teaching (Burrell & Melchert 2007), issues of language or culture (Prasad, Sanders, & McTaggart 2004), the ability or aptitude of students (Lister 2007), etc.

It is these topics, and others like them, that make up the theme dimension of the classification.

2.3 Scope

The *scope* dimension attempts to quantify the breadth of a paper's involvement with the computing education community. A paper set in a single subject could well be written with no community involvement by the authors, whereas a multi-institutional study of students' results really necessitates the involvement of colleagues from each institution.

Scope captures this breadth of involvement by noting whether a paper involves a single subject (Hu 2006), a range of subjects within the program or department (Nesbit 2003), a range of subjects within the whole institution (Baker & Nesbit 2006), or multiple institutions (Rudsar, Joyce, & Kolahi 2005). There are also papers, such as this one, that are not set in subjects at all, and that would therefore be given a scope of 'not applicable'.

2.4 Nature

Many education researchers are aware of the distinction between practice papers and research papers. The former report on classroom innovations, such as a new way of teaching or a new tool for assessing, while the latter describe the design, implementation, data collection, and analysis of a project that sets out to answer a specified research question. The *nature* dimension applies a finer grain to this traditional binary distinction, with four distinct categories.

A *study* paper (Young & McSporran 2001) reports on a research project that begins with a research question, designs a study or survey to answer that question, carries out the study or survey, gathers the data, and analyses it to see what light it sheds on the question.

An *analysis* paper (Clear & Young 2006) is somewhat similar, but analyses existing data, such as a body of literature or students' results in past courses, instead of carrying out a study or survey to generate the data.

A *report* paper (Athauda 2007) is the characteristic practice paper, describing something that was tried in the classroom (or occasionally elsewhere), and reporting on what was done and what outcomes it had.

A *position/proposal* paper (Corich 2006) describes something that the authors believe, or something that they are proposing to do, without as yet having anything to report upon.

3 The method

For the paper that introduced his system, Simon (2007) worked alone to classify a body of computing education papers. Still working alone, Simon (2008) applied the scheme to all six years of Koli Calling, the annual Baltic Sea conference on computing education.

We have worked as a group, in the hope of eliminating any bias that classification by one individual might have introduced.

The current paper arises from work intended to familiarise more researchers with the classification system, to share the load of classification among these additional researchers, and to test the system's inter-rater reliability.

At a two-day workshop in January 2008, Simon familiarised the rest of us with the classification system and with his reasons for having devised it. After some explanation of the system, we jointly categorised the papers from NACCQ 2003. There was occasional disagreement on some of the classifications, and indeed on whether certain papers were computing education papers, but this disagreement was resolved in group discussion. We then did the same for papers from NACCQ 2004.

In the next phase of our work, we individually categorised the papers from NACCQ 2005, and then gathered to discuss our findings. As with the previous phase, the discussion resulted in a consensus classification of the papers.

Again working individually, we next categorised the papers from NACCQ 2006. This time, rather than seeking consensus, we compared our findings using a measure of uniformity known as Fleiss's kappa (Fleiss 1971).

Fleiss's kappa is a statistical measure of inter-rater reliability that can be applied to multiple raters, and is designed to compare the level of agreement with the level that would be expected to arise through chance if all raters made their ratings randomly.

Application of Fleiss's kappa results in a percentage agreement, in this case for each dimension of the system. On this and other kappa measurements, an agreement of less than 40% is generally considered to be poor, between 40% and 75% is considered fair to good, and more than 75% is rated excellent (Banerjee et al 1999).

Our last task at the workshop was to consider how the system might be adjusted to make it easier to apply. We agreed that 'theme' was a better name for that dimension than Simon's original 'topic' (which is too easily confused with context); agreed that 'study' was a more accurate name for that nature than Simon's original 'experiment'; and added a number of themes and contexts that we had found in our examination of these papers.

Following the workshop, we individually categorised the papers from NACCQ 2007, and again compared our results using Fleiss's kappa. Then we formed pairs to discuss and merge our individual findings, and applied Fleiss's kappa once more, this time to the paired classifications.

Table 1: measures of agreement over three applications of Fleiss's kappa

	2006 papers (individual)	2007 papers (individual)	2007 papers (paired)
context	44%	56%	65%
theme	57%	37%	54%
scope	54%	43%	59%
nature	32%	47%	79%

Table 1 shows that there was general improvement over the three applications of Fleiss's kappa, although even after the paired classification only one dimension, nature, had an agreement in the excellent range. It has been noted that the kappa is higher when there are fewer categories (Sim & Wright 2005), so we might expect better agreements for scope and nature than for context and theme.

At the same time, there are papers that as a group we found unusually difficult to classify. In one outstanding case (Brook & Gasson 2007) the seven raters classified the same paper into seven different themes and all five possible scopes. The paired discussions settled the theme comfortably into educational technology, but still left some disagreement as to the scope.

Finally we divided up the papers from NACCQ 2000, NACCQ 2001, and NACCQ 2002, and classified those individually. We had thus, using various approaches, classified all of the computing education papers from the past eight offerings of the conference – the offerings for which we were able to find the proceedings on the web.

4 The findings

There is substantial fluctuation in the numbers of papers at successive NACCQs. Table 2 shows the total number of papers presented at each of the years under consideration, the number of those that are listed as full papers, and the number of those that we classified as computing education papers, as opposed to elucidations of non-education research projects. It is possible that there were one or two more computing education papers: several papers could not be downloaded from the web proceedings, as their links there were invalid, and for one or two of those we were unable to find alternative sources for the papers. Those papers we simply had to leave out, and so they are not counted among our tally of computing education full papers. We thus classified a total of 157 papers.

Table 2: numbers of NACCQ papers by year

Year	Papers	Full papers	CompEd full papers
2000	84	63	48
2001	101	23	14
2002	57	15	8
2003	123	21	12
2004	147	23	15
2005	96	14	9
2006	66	42	24
2007	64	35	27
Total	738	236	157

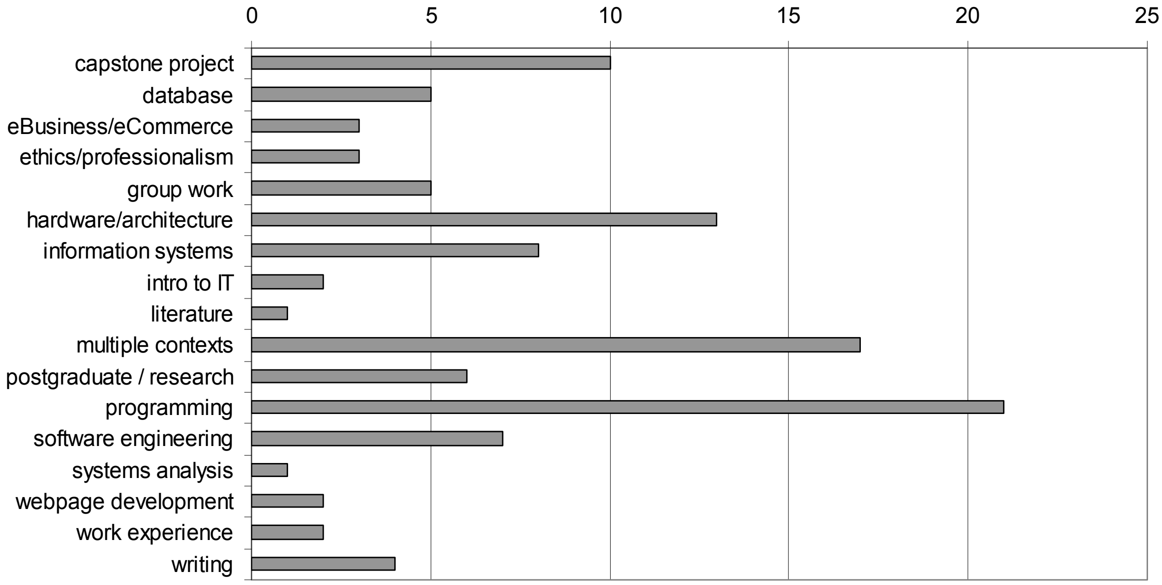

Figure 1: contexts of the 110 papers that have a context

4.1 Context

Of the papers we considered, 47 did not have an identifiable context; that is, they were not reporting on computing education work carried out in the literature or in the teaching of one or more subjects. Figure 1 shows the contexts of the remaining 110 papers.

We see that there are more papers (21) situated in programming subjects than in any other context. There are almost as many (17) set in multiple contexts. The number of papers (13) in hardware/architecture is a little misleading, as this context is broader than its name suggests: it includes subjects in networking, operating systems, and related areas. The next single context of interest is capstone project, which is the setting for 10 papers. The high frequency of papers in the programming context is consistent with Simon's (2007) analysis of recent NACCQ and ACE papers, whereas the frequency of capstone project is notably higher.

4.2 Theme

Figure 2 shows the themes of the papers – what they are about. The field is clearly dominated by curriculum (what we choose to teach) and teaching/learning techniques (how we choose to teach it). In the next group are assessment techniques (how it is assessed), teaching/learning tools (software to assist with the teaching), and educational technology (technological advances that impact on the teaching).

4.3 Scope

Figure 3 summarises the scopes of the papers. Nearly half of them (70/157) are set in a single subject. Some 18% deal with multiple subjects across the program or department, 3% with a broader range of subjects across the institution, and 9% involve subjects at two or more institutions.

As suggested earlier, this means that nearly half of the

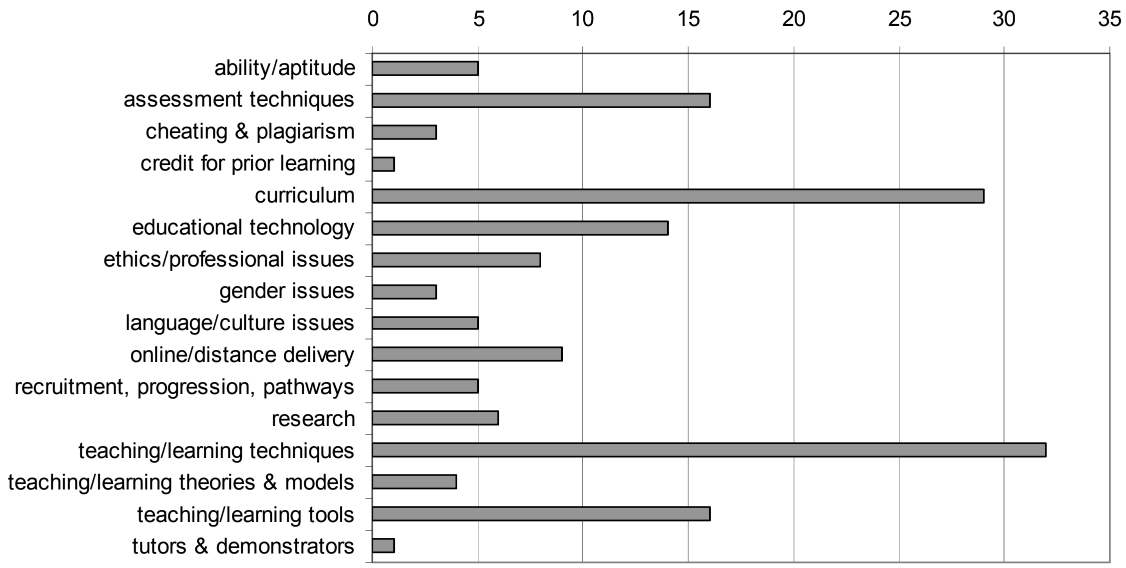

Figure 2: themes of all papers

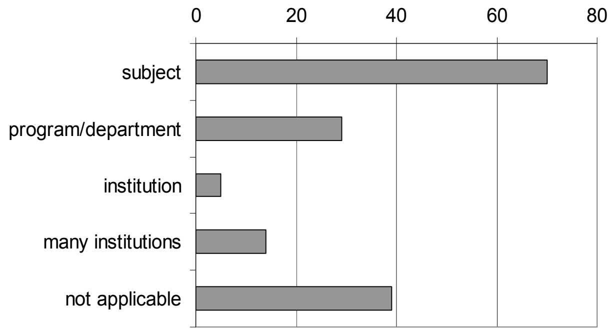

Figure 3: scopes of all papers

papers need not have involved any collaboration with other members of the computing education community, while 12% show almost certain evidence of such collaboration.

4.4 Nature

The natures of all 157 papers are summarised in Figure 4. Nearly a quarter of the papers are position papers – expressions of a belief – or proposals of work not yet begun. The highest proportion, 40%, report on something, typically something novel, that has been done in the classroom or some other education-related context. These are the papers that are sometimes referred to as practice papers.

The remaining 37% of papers are what we can unequivocally call research papers. The thrust of these papers is the analysis of data and the reporting of the outcomes, whether the data is pre-existing (analysis papers) or generated expressly for this project (study papers). In making this point, it is not our intention to devalue the position/proposal and report papers. These will always have their place in computing education conferences. But in the current academic climate, with its increasing emphasis on research, we feel it is important to recognise the publications that fall clearly into the research bracket. Indeed, we shall now proceed to see whether there has been a perceptible increase in the proportion of these papers over the period of the study.

4.5 Drilling deeper into the data

With four distinct dimensions to the classification system, along with the time dimension made possible by our study of eight years of NACCQ conferences, there is great scope for further exploration of the data. Here we present three of our more interesting findings from this deeper level of analysis.

4.5.1 Timeline for nature

Because of the low numbers of computing education papers in 2002 and 2005 (as shown earlier in Table 2), it would not be statistically sound to examine each year for trends. Instead we have divided the years into three 'periods', with a reasonable number of papers in each. These periods are 2000-2001 (62 papers), 2002-2004 (35 papers), and 2005-2007 (60 papers).

Figure 5 shows the proportions of papers by nature for each of these periods. The proportion of position/proposal papers has remained relatively steady at about 23% of the papers in each period. The proportion of analysis papers, while generally small, shows a slight increase. But the clearly visible trends are the decreasing proportion of report papers, from 47% to 32%, and the corresponding increase in study papers, from 22% to 36%. It is tempting to think of these changes as a response to the increasing pressure on academics to conduct recognisable research. We note, however, that these trends are not statistically significant by the traditional criterion of $p<0.05$.

4.5.2 Themes in programming

As there are more papers in the programming context than in any other, we thought it might be interesting to explore the themes of these particular papers.

Figure 6 shows the themes of the 21 papers whose context is programming. The overall list of themes (shown earlier in Figure 2) was dominated by teaching and learning techniques (20%) and curriculum (18%). Within the programming context, curriculum papers take up a comparable proportion (10%) of the papers, but teaching and learning techniques are the theme of 48% of these 21 papers. We speculate that there are disproportionately more papers on this theme in this context because programming and software engineering are recognised as subjects that are hard to teach and learn, and therefore continually spawn fresh attempts to find more effective ways to teach them.

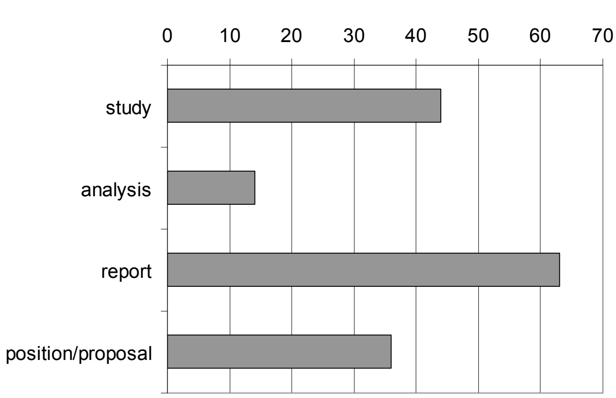

Figure 4: natures of all papers

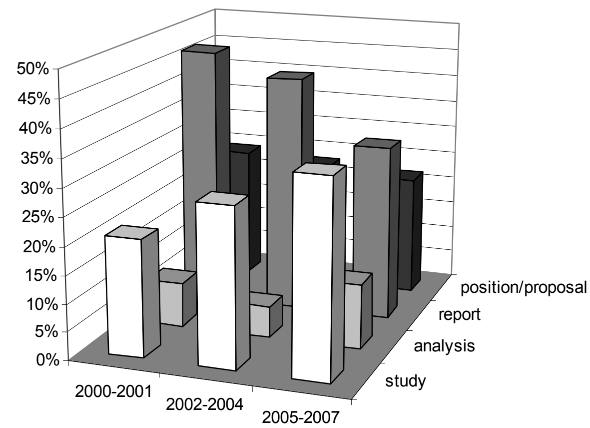

Figure 5: proportions of papers by nature for each period

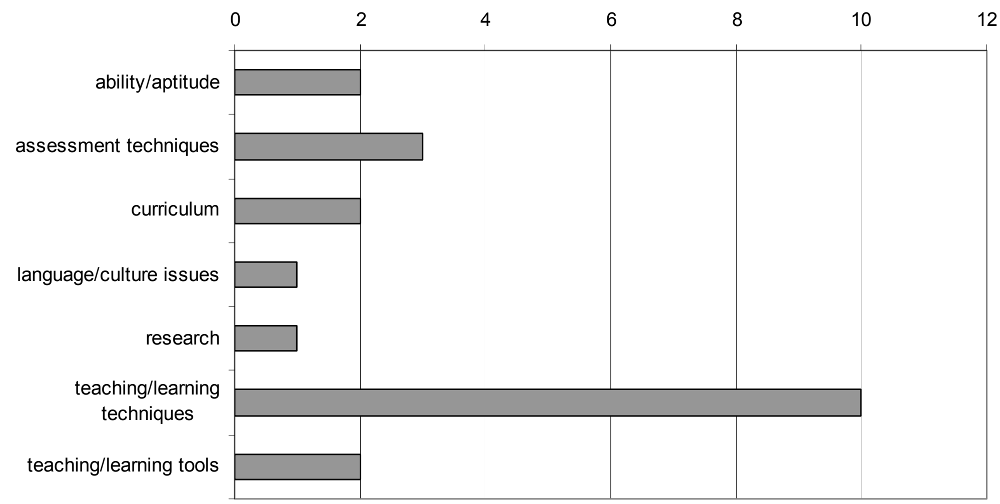

Figure 6: themes of the papers whose context is programming

4.5.3 Reduced nature and scope

We have already suggested that study and analysis papers are clearly research papers whereas report and position/proposal papers are more likely to be practice papers. We have also suggested that a broader scope might indicate a greater involvement with the computing education community.

To test these ideas, we have merged the study and analysis papers into a single ‘empirical’ group and the report and position/proposal papers into a ‘non-empirical’ group, and have plotted these groups against scope (Figure 7).

Here we find a statistically significant correlation between this reduced nature and scope: non-empirical papers are likely to have a narrower scope, empirical papers to have a broader scope. That is, papers that simply report on something or propose something are not so likely to entail involvement with the community, whereas papers that involve conducting a study and reporting findings will tend to entail more involvement in their local or broader community.

5 Conclusions

We have applied Simon’s classification system to the 157 computing education papers published at NACCQ conferences between 2000 and 2007, giving an overview of the themes of these papers (what they are about) and their contexts (what sort of subjects they are based in). The contexts of these papers are dominated by programming subjects, hardware/architecture/systems/ network subjects, and capstone projects. Their themes are dominated by teaching/learning techniques, assessment techniques, teaching/learning tools, curriculum, and educational technology.

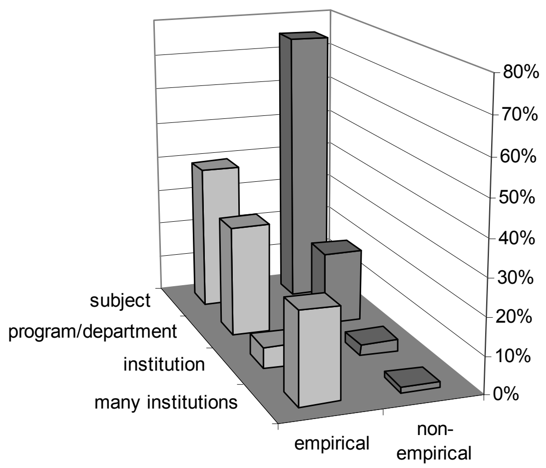

Figure 7: reduced nature vs scope

The scopes of the papers show that 45% of them are set within single subjects, 18% in multiple subjects within a single program or department, and only 12% in a range of subjects across the whole institution or multiple institutions.

Analysis of the natures of the papers shows that a fairly steady 23% of the papers at NACCQ either expound a position or outline a proposal; that a large but diminishing proportion are practice papers, reporting on something that has been done; that a small and fairly steady proportion are papers analysing existing data to answer a research question; and that a large and increasing proportion report on studies, which involve first generating data and then analysing it. In the latter two categories we recognise an increasing proportion of NACCQ papers that can unequivocally be categorised as research.

We find that the context of programming subjects has a disproportionately high representation of papers expounding teaching and learning techniques. And we find that study and analysis papers, when grouped together as empirical papers, are more likely to entail a broad involvement with the computing education community than are report and position/proposal papers.

6 Acknowledgements

This study was supported by a Special Projects Grant from the ACM Special Interest Group in Computer Science Education (SIGCSE). The authors thank the University of Wollongong, and Meghan Gestos and Espi Riley in particular, for assistance with arrangements, accommodation, and catering for the workshop.

7 References

Athauda, RI (2007): A revision of a database course with a real-world project. *20th Annual Conference of the National Advisory Committee on Computing Qualifications (NACCQ 2007)*, Nelson, New Zealand, 229-236.

Baker, A and Nesbit, T (2006): Transnational education – the students coming onshore: a case study. *19th Annual Conference of the National Advisory Committee on Computing Qualifications (NACCQ 2006)*, Wellington, New Zealand, 11-15.

Banerjee*, M,* Capozzoli, M, McSweeney, L, and Sinha, D (1999): Beyond kappa: a review of interrater agreement measures. *Canadian Journal of Statistics* **27**(1):3-23.

Brook, P and Gasson, J (2007): Raising the bar on VC: using video conferencing and NACCQ modules in secondary schools. *20th Annual Conference of the National Advisory Committee on Computing Qualifications (NACCQ 2007)*, Nelson, New Zealand, 35-40.

Burrell, C and Melchert, M (2007): Augmenting compiler error reporting in the Karel++ microworld. *20th Annual Conference of the National Advisory Committee on Computing Qualifications (NACCQ 2007)*, Nelson, New Zealand, 41-46.

Clear, T and Young, A (2006): An exploratory study into the impact of NACCQ research. *19th Annual Conference of the National Advisory Committee on Computing Qualifications (NACCQ 2006)*, Wellington, New Zealand, 45-52.

Corich, S (2006): The case for an ITP collaborative computing degree. *19th Annual Conference of the National Advisory Committee on Computing Qualifications (NACCQ 2006)*, Wellington, New Zealand, 61-66.

Fleiss, JL (1971): Measuring nominal scale agreement among many raters. *Psychological Bulletin* **76**(5):378-382.

Holt, A (2006): An analysis of negative marking in multiple-choice assessment. *19th Annual Conference of the National Advisory Committee on Computing Qualifications (NACCQ 2006)*, Wellington, New Zealand, 115-118.

Hu, MJ (2006): Component programming. *19th Annual Conference of the National Advisory Committee on Computing Qualifications (NACCQ 2006)*, Wellington, New Zealand, 127-133.

Li, X (2007): Incorporating a code review process into the assessment. *20th Annual Conference of the National Advisory Committee on Computing Qualifications (NACCQ 2007)*, Nelson, New Zealand, 125-131.

Lister, R (2007): The neglected middle novice programmer: reading and writing without abstracting. *20th Annual Conference of the National Advisory Committee on Computing Qualifications (NACCQ 2007)*, Nelson, New Zealand, 133-140.

Mann, S and Williamson, A (eds) (2003): Proceedings of the 16th Annual Conference of the National Advisory Committee on Computing Qualifications (NACCQ 2003). Accessed March 2008. http://www.naccq.ac.nz/conference04.html?page=50

Nesbit (2003): 2 classes, 2 languages, 2 methodologies and 1 assignment. *16th Annual Conference of the National Advisory Committee on Computing Qualifications (NACCQ 2003)*, Palmerston North, New Zealand, 135-142.

Plimmer (2000): Assessment: central to learning. *13th Annual Conference of the National Advisory Committee on Computing Qualifications (NACCQ 2000)*, Wellington, New Zealand, 273-277.

Prasad, C, Sanders, KE, and McTaggart, JA (2004): Decoding Programming Concepts in EAL and EFL Students: Is there a difference? *17th Annual Conference of the National Advisory Committee on Computing Qualifications (NACCQ 2004)*, Christchurch, New Zealand, 168-173.

Rudsar, S, Joyce, D, and Kolahi, S (2005): Status of wireless networks in educational institutes in Auckland. *18th Annual Conference of the National Advisory Committee on Computing Qualifications (NACCQ 2005)*, Tauranga, New Zealand, 91-96.

Sim, J and Wright, CC (2005): The kappa statistic in reliability studies: use, interpretation, and sample size requirements. *Physical Therapy* **85**:257-268.

Simon (2007): A classification of recent Australasian computing education publications. *Computer Science Education* **17**(3):155-169.

Simon (2008): Koli Calling comes of age: an analysis. *Seventh Baltic Sea Conference on Computing Education Research (Koli Calling 2007),* Koli National Park, Finland, November 2007. Conferences in Research and Practice in Information Technology **88**:119-126.

Surendran, K and Young, FH (2000): Teaching software engineering in a practical way. *13th Annual Conference of the National Advisory Committee on Computing Qualifications (NACCQ 2000)*, Wellington, New Zealand, 345-350.

Toki, I (2000): An argument for the information system educators' preferred model of internet-based business simulations. *13th Annual Conference of the National Advisory Committee on Computing Qualifications (NACCQ 2000)*, Wellington, New Zealand, 381-387.

Tupu, J, Ngatuere, J, and Young, A (2004): Te korowai hou. *17th Annual Conference of the National Advisory Committee on Computing Qualifications (NACCQ 2004)*, Christchurch, New Zealand, 190-196.

Young. S and McSporran, M (2001): Confident men - successful women: gender differences in online learning. *14th Annual Conference of the National Advisory Committee on Computing Qualifications (NACCQ 2001)*, Napier, New Zealand, 185-193.