
This is an electronic reprint of the original article.
This reprint may differ from the original in pagination and typographic detail.

Author(s): Simon

Title: A classification of recent Australasian computing education publications

Year: 2007

Version: Post print

Please cite the original version:

Simon. 2007. A classification of recent Australasian computing education publications. Taylor & Francis, UK. Computer Science Education, volume 17, issue 3, pages 155-169. ISSN 0899-3408 (print)/ISSN 1744-5175 (online). DOI 10.1080/08993400701538021

Rights: © Taylor & Francis, UK. Reprinted with permission.

This is the author's accepted manuscript of an article published as the version of record in Computer Science Education © 30 May 2008. <http://www.tandfonline.com/doi/full/10.1080/08993400701538021>

This publication is included in the electronic version of the article dissertation:
Simon. Emergence of computing education as a research discipline.
Aalto University publication series DOCTORAL DISSERTATIONS, 150/2015.

All material supplied via Aaltodoc is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.

A Classification of Recent Australasian Computing Education Publications

Simon*

University of Newcastle, Australia

A new classification system for computing education papers is presented and applied to every computing education paper presented between January 2004 and January 2007 at the two premier computing education conferences in Australia and New Zealand. We find that while simple reports outnumber other types of paper, a healthy proportion of papers address and answer a research question. We find that more papers deal with programming courses than with other courses, and that more than half of all publications are situated in single subjects. To the extent that differing circumstances permit, we compare our results with those of an earlier study of the SIGCSE conference, and find that the Australasian publications include fewer simple reports and more papers describing analysis and experiment. We note a reasonable number of publications on multi-institutional work, which we interpret as evidence of a sense of computing education community with Australia and New Zealand.

Keywords: *Computing education research; literature review; classification*

1. Introduction

What is the current state of computing education research in Australia and New Zealand? What are the papers about, and how many of them are research papers as opposed to practice papers?

This paper answers those questions by examining and categorising every computing education paper accepted over the past three and a half years at either the Australasian Computing Education Conference (Australia and New Zealand) or the Conference of the National Advisory Committee on Computing Qualifications (New Zealand). This is by no means a complete set of recent Australasian computing education papers, as such papers also appear in journals and in more generalist conferences; but we believe that it is reasonably representative.

The original intention was to categorise the papers according to existing systems, but when it became clear that these systems were not suited to the task, either individually or together, a new system of classification for computing education papers was developed, and it is this system that has been used to categorise the papers.

A note on terminology: there are many different names for the discrete units of study that together comprise a degree program. They are variously called subjects, courses, papers, units, and perhaps other names. In this paper we shall use the term ‘subjects’.

2. Existing classification systems for computing education papers

As computing education research becomes older and perhaps more venerable, it necessarily moves to include itself among its legitimate topics of interest; the reach of computing education literature expands ever so slightly to include a scattering of papers about computing education research. These ‘meta-

* Corresponding author. School of DCIT, PO Box 127, Ourimbah, NSW 2258, Australia. Email: simon@newcastle.edu.au

research' papers will sometimes include a system for categorising the papers or the research that is their subject matter.

In her contribution to a panel paper (Clancy, Stasko, Guzdial, Fincher, & Dale, 2001) Sally Fincher suggests that computing education research communities might be categorised in two different ways: according to the subject area of the research and according to the temperament and methodology of the researcher. At first sight these ways appear to represent two distinct dimensions of categorisation, but Fincher then observes a fairly firm correspondence between the two, so in fact she appears to be describing two different facets of the same categories.

Fincher's classification can be summarised as follows . . .

- small-scale investigations of a single aspect of discipline or practice; characterised by practitioner research and 'action research' approaches
- investigations of specific mental and conceptual skills; characterised by quantitative and statistical methodological approaches
- investigations based within the educational tradition; tend to be theoretically motivated, applying educational theory to computing education situations and material
- investigations motivated by the use of tools in teaching and learning; tend to be technology-driven

Fincher does not suggest that these categories are hard and fast, and she takes care to use words such as 'often' and 'not exclusively'. The breakdown has been applied in at least one survey paper (Berglund, Daniels, & Pears, 2006), but we are not aware of Fincher's scheme having been used to categorise a corpus of papers. Further, her scheme pertains to research, and thus fails to categorise those many computing education papers that describe practice rather than research. The system presents a useful starting point, but would undoubtedly require significant tuning when used to comprehensively categorise a large body of work in research and practice.

Another research-oriented classification system was proposed by Randolph, Bednarik, and Myller (2005) and was applied to four years of publications from "Koli Calling", the annual Baltic Sea Conference on Computing Education. This system delves in fine detail into the methodology used in the research papers that make up 40% of the 74 papers in that corpus. It leaves few if any stones unturned in its analysis of the methodologies of those papers, but devotes little attention to the remaining 60% of the papers, categorising them as

- literature reviews, meta-analyses
- program descriptions without anecdotal evidence
- program descriptions with anecdotal evidence
- theoretical, methodological or philosophical papers
- technical investigations
- other

David Valentine (2004) surveyed 20 years of SIGCSE Technical Symposium papers dealing with first-year Computer Science subjects, putting each paper into one of six categories. The categories, which are appealing but somewhat quirky, can be summarised thus:

- experimental: papers including any sort of scientific analysis
- Marco Polo: descriptions of the application of a new curriculum, language, or course
- philosophy: attempts to generate debate on philosophical grounds
- tools: software tools developed to assist with aspects of teaching/learning or assessment
- nifty: innovative, interesting ways to teach abstract concepts
- John Henry: papers describing outrageously difficult ways of undertaking a simple task (the reference is to a US folk hero and the song that tells his tale)

As it was designed explicitly for CS1/CS2 subjects (typically first and second programming subjects, or a first programming subject and first data structures subject), this model might be difficult to extend to computing education papers that deal with other subjects.

Fincher and Petre (2004) propose ten subfields for computer education research:

- student understanding
- animation, visualisation and simulation
- teaching methods
- assessment
- educational technology
- transferring professional practice into the classroom
- incorporating new developments and new technologies
- transferring from campus-based teaching to distance education
- recruitment and retention
- construction of the discipline

While each of these subfields stands well in its own right, the distinctions between them are not necessarily clear cut. For example, animation, visualisation and simulation are often applied as teaching methods, and indeed transferring professional practice into the classroom will generally involve some description or analysis of teaching methods. Likewise, ‘educational technology’ will often have much in common with ‘incorporating new developments and new technologies’.

Pears, Seidman, Eney, Kinnunen, & Malmi (2005) combine Fincher’s and Petre’s subfields into four broader areas:

- studies in teaching, learning, and assessment
- institutions and educational settings
- problems and solutions
- computing education research as a discipline

These areas serve well for their intended purpose, which was to begin the identification of a core literature of computing education research, but are rather too broad for a detailed analysis of a large corpus of publications.

We spent some time seeking areas of overlap and difference between these classification systems in an attempt to merge them into a single all-encompassing system. For example, an obvious overlap between Fincher’s and Valentine’s systems is the tools category. In addition, Fincher’s ‘small-scale investigations of a single aspect’ might bear some relation to Valentine’s Marco Polo category. However, there is no clear relationship between their remaining categories. As another example, while Randolph’s methodological analysis could be applied to Valentine’s ‘experimental’ papers, there is no clear relationship between Valentine’s remaining five categories and Randolph’s six categories of ‘other’ papers.

The more serious problem that emerged during the attempt to combine the systems is that despite their linear appearance they are sometimes describing orthogonal aspects of the papers or the research. Valentine’s experimental, Marco Polo, and philosophy categories indicate *types* of paper, while a tools classification indicates a paper’s *subject matter*. Valentine deals with this by specifying an order for the categories and slotting a paper into the highest-level category possible, so that a paper describing a tool and some analysis on its application would be classified as experimental, but this solution sidesteps the very real difference between a paper’s type and its subject matter.

Pears et al. do introduce a second dimension, categorising papers as influential, seminal, or synthesis. Again, this is appropriate for the purpose of identifying a core literature, but would leave most computing education publications unclassified.

Concluding that no one-dimensional classification system can do justice to all aspects of interest in computing education papers, we have developed a system that meets our goals of categorising a large corpus of publications according to a number of broad aspects.

3. A new system for classifying computing education papers

An obvious first dimension is what a paper is about, which we shall call its **topic**. The topic of a publication is usually reasonably clear, but there are traps for the unwary. A paper such as *Teaching Java first: experiments with a pigs-early pedagogy* (Lister, 2004) might appear to be about first-year programming, but on closer consideration it is about a tool that was developed to assist in the teaching of first-year programming.

Nevertheless, first-year programming plays a significant part in the paper – a part that is better described as its **context**, the educational situation in which the tool was developed and used. Context therefore becomes a second dimension in our system.

A third dimension, the **nature** of a paper (for example, ‘report’ or ‘experiment’), classifies it along lines similar to Valentine’s first three categories.

Finally, the **scope** of a paper measures the breadth of its context. Is it based in a single subject, the set of subjects in a degree program, or the set of subjects in an institution; or is it based on work conducted at multiple institutions?

This classification system, like any other, is inescapably subjective. Not all of the dimensions are useful for all purposes, and further dimensions can be imagined with little effort. For example, Fincher’s ‘investigations based within the educational tradition’ suggests a further dimension. It would be orthogonal to all four of the current dimensions, in that almost any category in any of the dimensions can include both papers that are pedagogically founded and papers that are not. Likewise, for papers that include some form of analysis, some readers might like to see them divided into qualitative, quantitative, and mixed methods (Lister, 2005). But we must stop somewhere, even if that somewhere is arbitrary, and we have chosen to stop at the four dimensions listed above.

Some of the categories in each dimension were obvious as soon as the dimension itself was chosen, while others emerged from the exercise of categorising papers. The dimensions and their categories are presented here as a complete set, but readers should bear in mind that they are to some extent dependent on the set of papers to which they have been applied, and that applying them to further papers might lead to the emergence of further categories, particularly in the topic and context dimensions.

3.1. The Topic Dimension

Once the notions of topic and context were separated, a set of 17 topics emerged (Table 1). Other topics come readily to mind, but these are the only ones that were found in the computing education papers examined for this study. While some of the topics might appear to make good neighbours (for example, language/culture issues and gender issues) there is no natural order to the topics as a whole, so they are presented in simple alphabetic order.

Table 1: Categories of the *Topic* dimension

ability/aptitude	What makes students good at computing; how they can best prepare for it; what processes they apply when doing it.
about research	Papers written <i>about</i> research; for example, surveys of the sort of research being published, or papers on the design of research instruments.
assessment techniques	Techniques that have been devised for, or applied to, assessment of students (including the application of existing tools).
assessment tools	Tools that have been developed to assist with assessment of students.
cheating & plagiarism	Why students cheat; how they cheat; how often they cheat; what can be done to catch them

	cheating; what can be done to discourage them from cheating.
credit for prior learning	Credit given for subjects in recognition of learning already undertaken.
curriculum	The content of a subject or a degree program.
distance/online delivery	Special issues to do with delivery of subjects online or by other forms of distance education.
educational technology	Application to education of recent developments in technology.
ethics/professional issues	The ethics of computing, along with other professional issues.
gender issues	Consequences of and adjustments for the differences between female and male students.
language/culture issues	Issues faced by and with students from linguistic or cultural backgrounds that differ from those of the majority.
recruitment	Attracting students to the study of computing.
teaching/learning	Teaching and/or learning as an object of study; the processes involved in the teaching/learning activity.
teaching/learning techniques	Techniques that have been devised for, or applied to, teaching and/or learning (including the application of existing tools).
teaching/learning tools	Tools that have been developed to assist with teaching and/or learning.
tutors & demonstrators	What preparation and/or support is needed by tutors and demonstrators; how this can be provided.

Just as there is sometimes confusion between a paper's topic and its context, there is sometimes confusion between topics. A paper such as *A learning system engineering approach to developing online courses* (Bower, 2006) might be on distance/online delivery, on teaching/learning tools, or on teaching/learning techniques. The tools option is eliminated by the arbitrary decision to call the use of existing tools a technique, but that still leaves two options. In cases such as this, the rule of thumb is to choose the category that seems to dominate the content of the paper.

3.2. The Context Dimension

Once separated from topic, the notion of context becomes a fairly simple matter: in what sort of subject is the work situated? Of course not all computing education research is situated in particular subjects. Some papers, such as *An exploratory study into the impact of NACCQ research* (Clear & Young, 2006), are clearly situated in the literature. Others, for example *An analysis of negative marking in multiple-choice assessment* (Holt, 2006), are applicable to many subjects and written without reference to any, and so are categorised as broad-based. The list of contexts was quite large at one point of our analysis, but we eventually merged some of the poorly-populated categories with others (for example, operating systems with hardware and architecture) and combined the rest into a single 'other' grouping. Once again there is no natural order to these categories, so they are presented in alphabetic order, with the standard exception that the 'other' category is listed last.

Table 2: Categories of the *Context* dimension

broad-based	Work done in the context of a number of different subjects.
capstone project	Work done in capstone project subjects.
group work	Work whose focus is group work, even if it overlaps with, say, capstone projects.
hardware/architecture	A broad context including hardware, computer architecture, operating systems.
literature	Work that is based primarily in the literature (eg a literature review).
programming	A broad context including programming, software engineering, data structures, algorithms.
work experience	Sending students out to work in industry as part of their study.
other	All other contexts, with low individual representation, including basic computing skills, computing theory, database, eCommerce, HTML, information systems, systems analysis, and more.

Some readers might be concerned at the way contexts have been grouped. They might prefer a far longer list, in which, for example, programming would be separated into first-year programming, higher-level programming, software engineering, data structures, algorithms, and perhaps others. But to be useful, an analysis must group items; and the more categories there are, the fewer members each will have, until there are almost as many categories as there are papers. The current set of eight categories was chosen as a good compromise between too many to be useful and too few to be useful.

3.3. The Nature Dimension

The nature dimension was developed to distinguish between ‘practice’ and ‘research’ papers. The distinction is made by both Fincher and Valentine, the latter urging the computing education community to continue its movement from practice papers toward research papers. We have settled on four categories, and this time we believe that they do form a natural order, so they are presented in that order.

Table 3. Categories of the *Nature* dimension

position	A statement of belief as to what ought to be done, or a proposal to begin doing something.
report	A report on something that has been done, perhaps with a simple survey of student satisfaction.
analysis	Addressing a research question by analysing existing data (such as perhaps pass rates over several years).
experiment	Addressing a research question by designing and conducting a study (which might be a survey).

Position papers are akin to, but form a superset of, Valentine’s philosophy papers. While some position papers do indeed set out to engender debate on a particular topic, others are simply statements of the authors’ beliefs. There is a clear and definite place for position papers in the literature, and indeed some of the most seminal papers fall into this category; an excellent example is *Goto considered harmful* (Dijkstra, 1968). Nevertheless, we believe that position papers should make up only a small proportion of the corpus. Further, while some readers might prefer to see a distinction between seminal and non-seminal position papers, we would contend that this distinction is often apparent only in hindsight, and cannot therefore be built into a system that is to be used to categorise contemporary papers.

Reports correspond with Valentine’s Marco Polo, tools, nifty, and John Henry papers, and bear a close resemblance to Fincher’s small-scale investigations of a single aspect of discipline or practice. They are essentially papers describing something that has been tried, generally in a single subject, and often concluding that it appeared to be successful. Unlike Valentine, we categorise papers as reports even if they exhibit the minimal analysis of conducting a student survey to confirm the appearance of success.

Analysis papers are those in which a research question is addressed by conducting some significant analytical work, generally on existing data such as student results over several offerings of the same subject.

Experiment papers are those whose authors set out to answer a particular question, devised a study (which might be a survey) to assist in that regard, carried out the study, and analysed the results.

Analysis and experiment papers would normally both be considered research, and indeed Valentine’s experimental category includes both of them. Reports would generally be considered practice papers, and position papers will tend to fall outside both of these camps.

3.4. The Scope Dimension

As we define it, scope is a measure of the breadth of the work. Reports are typically based on single subjects, but might sometimes report on initiatives introduced across a whole department or degree, or even a whole institution. Analysis and experiments can exhibit these same scopes, but are more likely to extend further, to multiple institutions. A broader scope might indicate a greater engagement with the computing education community, and will often give more confidence in the generalisability of results. A paper whose scope is a single subject can be written with little community engagement, whereas a collaboration across multiple institutions suggests a great deal of engagement. Again there is an apparent natural order, and the categories are listed accordingly.

Table 4: Categories of the *Scope* dimension

subject	a single subject
program/department	many or all of the subjects within a department or a degree program
institution	many or all of the subjects or degree programs within an institution
many institutions	involving at least two institutions
not applicable	not all papers have a meaningful scope

The ‘not applicable’ category is required to capture those papers that are independent of this measure. For example, *Where have all the students gone? IT secondary education in New Zealand* (Howard & Atkins, 2006), a paper on recruitment, is not based on a subject, a degree, a particular institution, or even many institutions.

While the institution category in one sense fits naturally between program/department and many institutions, a paper dealing with multiple disciplines within a single institution tends to involve a broader form of collaboration than a paper dealing with essentially the same subject at multiple institutions, and we would not expect to find many papers exhibiting ‘institution’ scope.

4. Four years of Australasian computing education conferences

While many computing conferences include a scattering of computing education papers, there are two Australasian conferences at which these papers predominate. The annual Australasian Computing Education Conference (ACE) is an obvious candidate, as all of its papers fall in some way into the computing education category. In New Zealand, the annual conference of the National Advisory Committee on Computing Qualifications (NACCQ) always includes a good proportion of computing education papers among its offerings. At the time of writing there have been seven of these conferences since 2004: four ACEs (held at the start of the year) and three NACCQs (held at midyear).

We have applied our new classification system to every paper published at ACE since 2004, and to every computing education paper published at NACCQ in the same time span. This gives a total of 175 papers, 129 from ACE and 46 from NACCQ.

All of the categorisation has been conducted by a single researcher. Determining the reliability of the system has therefore been left for future work.

5. Papers categorised along each dimension

5.1. Topic

Figure 1 shows the number of papers in each of the identified topics.

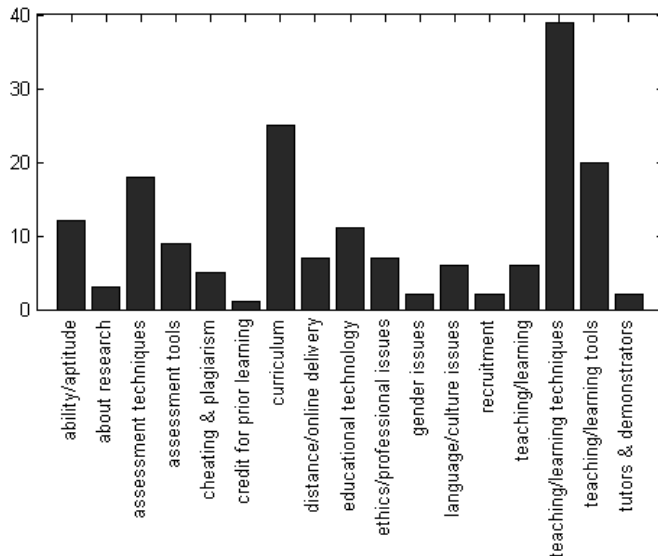


Figure 1. Count of papers for each topic

There are few surprises here. Teaching and learning techniques (making up 22% of the total) and tools (11%), assessment techniques (10%) and tools (5%), and curriculum issues (14%) are grist to the mill of computing education publications, and there will always be papers exploring the impact of advances in technology (6%). Perhaps the least expected of the well populated categories is ability/aptitude (7%), of which more will be written later.

5.2. Context

Having gone to some effort to distinguish between a paper’s topic and its context, we now present the number of papers in each context (Figure 2).

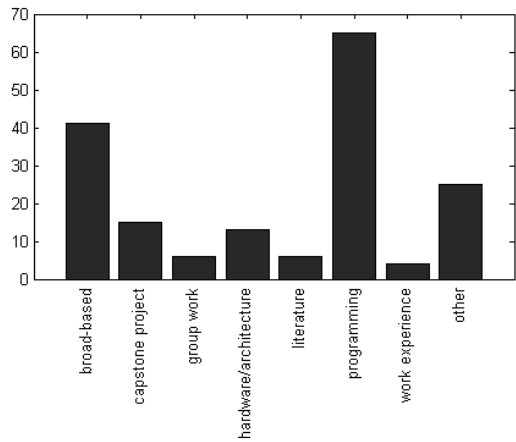


Figure 2. Count of papers for each context

While intuition might suggest that most computing education papers are *about* programming, we now see that programming is the context for, but not necessarily the topic of, the greatest proportion (37%) of computing education papers. In order to see what those programming papers are actually about, we will later analyse the programming papers by topic.

5.3. Nature

In figure 3 we see how many position papers, reports, analytical papers, and experiments are in the corpus.

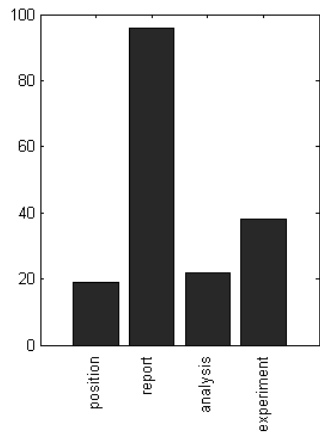


Figure 3. Count of papers by nature

It is no surprise that reports are in a big majority, making up 55% of the total, but it is pleasing to see reasonable numbers of analysis papers (13%) and experiment papers (22%) in the mix. We have a feeling, unsubstantiated at present, that reports were once even more dominant; and we join earlier classifiers in urging an increase in the number of analysis and experiment papers.

5.4. Scope

Figure 4 indicates how many papers fall into each of the identified scopes.

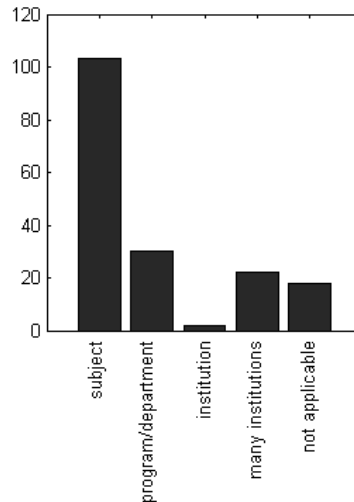


Figure 4. Count of papers by scope

Again, there are few surprises here. The bulk of the papers (59%) deal with a single subject. A reasonable number (17%) cover the whole of the department or degree program in which they are situated, and almost as many (13%) involve two or more institutions. We interpret this last point as indicating that there is a reasonably healthy sense of community in Australasian computing education. Because it is in many ways easier to collaborate with people in the same academic area than with people in the same geographic area, there are very few papers (2%) whose scope is the whole of the institution.

6. The promise of analysis along two dimensions

While the one-dimensional categorisations are interesting, there might be rather more to be discovered by pairing the dimensions. Although there is limited opportunity to do that within this paper, we offer a small taste by examining the topics of papers whose context is programming. Figure 5 shows the topics covered by the 65 papers (37% of the corpus) that are situated in a programming context.

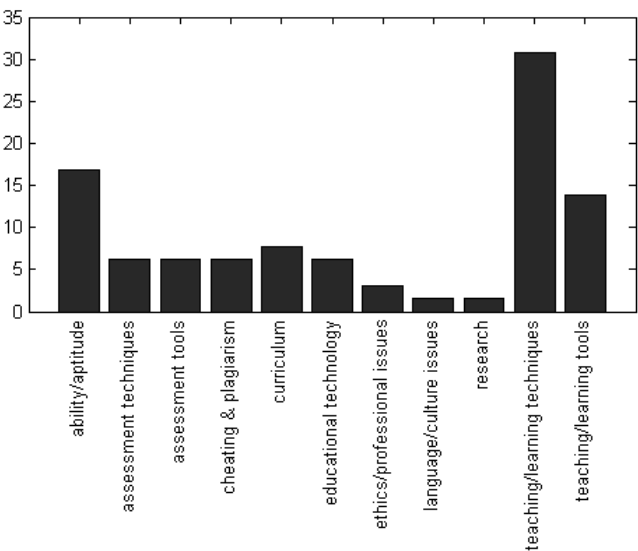


Figure 5. Proportion of programming papers in each topic

This breakdown is particularly illuminating. It is generally understood that one major function of computing education conferences is to host a ‘swap meet’ in which academics demonstrate and distribute techniques and tools that they believe have made their work easier. Yet while teaching/learning techniques do indeed dominate, at 31%, ability/aptitude sneaks into second place at 17%, just ahead of teaching/learning tools at 14%. Even if we were to combine teaching/learning tools with assessment tools (6%), ability/aptitude would not be far behind their joint 20%.

Programming aptitude is a perennial issue in computing education, but we believe that its strength in this corpus is attributable in part to a recent surge in multi-institutional studies of the topic such as those presented in Simon et al. (2006) and Whalley et al. (2006).

7. Discussion

One obvious question arising from this analysis is how the results compare with previous work. This is not an easy question to answer, as this system of classification is so different from previous ones.

The only prior analysis that admits of any reasonable comparison is that of Valentine (2004), whose classification has much in common with our nature dimension. His experimental papers relate fairly closely to our analysis and experiment, his philosophy papers to our position papers, and all the rest of his papers (Marco Polo, tools, nifty, and John Henry) to our reports.

Remembering that Valentine’s analysis was just of CS1/CS2 programming papers, we will compare his results not only with our overall results, but also with the subset of our papers that fall into the programming context (see Table 5).

Table 5. Comparison of our Australasian results with Valentine’s SIGCSE Technical Symposium results

	Valentine (CS1/CS2) (N=444)	Simon (overall) (N=175)	Simon (programming) (N=65)
position (philosophy)	10%	10%	2%
report (Marco Polo, tools, nifty, John Henry)	69%	55%	55%
analysis/experiment (experimental)	21%	35%	43%

The comparison suggests that both overall and in programming, the Australasian conferences have a higher proportion of analysis and experiment papers and a lower proportion of report papers than the SIGCSE papers analysed by Valentine. When it comes to position papers, the overall proportion in the Australasian conferences is similar to that found by Valentine, while in the programming context the proportion is rather lower than Valentine's.

However, these comparisons must be regarded with great caution, given the great differences in time frames, selection criteria, and classification systems between the two studies.

8. Conclusions

Having developed a comprehensive system for the classification of computing education papers, we have applied that system to all 175 computing education papers published in the two most relevant Australasian conferences between January 2004 and January 2007.

Analysis of the results illustrates that in the nature of the papers, the corpus is dominated by reports; in context, it is dominated by programming; in topic, by teaching/learning techniques; and in scope, by the single subject. Papers describing research, whether analytical or experimental, make up a reasonable proportion, but come nowhere near challenging the dominance of reports.

Valentine (2004) challenged the computing education community to reduce the proportion of subject-based reports and increase the proportion of papers describing analysis and experiment. We echo that challenge, and add our own wish list: a reduction in position papers and an increase in collaborations across and between institutions.

Along with a determination of the reliability of the system, future work must clearly include a thorough exploration of the corpus along paired dimensions, and a repeat analysis a few years hence, to determine the extent to which our wishes have come true.

Acknowledgements

The author gratefully acknowledges the valuable advice of the guest editors and the reviewers of this paper.

References

- Berglund, A., Daniels, M., & Pears, A. (2006). Qualitative research projects in computing education research: an overview. *Proceedings of the Eighth Australasian Computing Education Conference, Australian Computer Science Communications*, 28:5, 25-33.
- Bower, M. (2006). A learning system engineering approach to developing online courses. *Proceedings of the Eighth Australasian Computing Education Conference, Australian Computer Science Communications*, 28:5, 45-54.
- Clancy, M., Stasko, J., Guzdial, M., Fincher, S., & Dale, N. (2001). Models and areas for CS Education Research. *Computer Science Education*, 11:4, 323-341.
- Clear, T., & Young, A. (2006). An exploratory study into the impact of NACCQ research. *Proceedings of the 19th Annual Conference of the National Advisory Committee on Computing Qualifications*, 45-52.
- Dijkstra, E. (1968). Goto considered harmful. *Communications of the ACM*, 11:3, 147-148.
- Fincher, S., & Petre, M. (2004). *Computer science education research*. Routledge Falmer.
- Holt, A. (2006). An analysis of negative marking in multiple-choice assessment. *Proceedings of the 19th Annual Conference of the National Advisory Committee on Computing Qualifications*, 115-118.
- Howard, J., & Atkins, C. (2006). Where have all the students gone? IT secondary education in New Zealand. *Proceedings of the 19th Annual Conference of the National Advisory Committee on Computing Qualifications*, 119-126.

- Lister, R. (2004). Teaching Java first: experiments with a pigs-early pedagogy. *Proceedings of the Sixth Australasian Computing Education Conference, Australian Computer Science Communications*, 26:5, 177-183.
- Lister, R. (2005). Mixed methods: positivists are from Mars, constructivists are from Venus. *ACM SIGCSE Bulletin*, 37:4, 18-19.
- Pears, A., Seidman, S., Eney, C., Kinnunen, P., & Malmi, L. (2005). Constructing a core literature for computing education research. *ACM SIGCSE Bulletin*, 37:4, 152-161.
- Randolph, J., Bednarik, R., & Myller, N. (2005). A methodological review of the articles published in the proceedings of Koli Calling 2001-2004. *Proceedings of the 5th Annual Finnish / Baltic Sea Conference on Computer Science Education*, 11:4, 103-109.
- Simon, Fincher, S., Robins, A., Baker, B., Box, I., Cutts, Q., de Raadt, M., Haden, P., Hamer, J., Hamilton, M., Lister, R., Petre, M., Sutton, K., Tolhurst, D., & Tutty, J. (2006). Predictors of success in a first programming course. *Proceedings of the Eighth Australasian Computing Education Conference, Australian Computer Science Communications*, 28:5, 189-196.
- Valentine, D. (2004). CS Educational Research: A Meta-Analysis of SIGCSE Technical Symposium Proceedings. *Proceedings of the 35th SIGCSE Technical Symposium on Computer Science Education, ACM SIGCSE Bulletin*, 36:1, 255-259.
- Whalley, J., Lister, R., Thompson, E., Clear, T., Robbins, P., Kumar, P., & Prasad, C. (2006). An Australasian study of reading and comprehension skills in novice programmers, using the Bloom and SOLO taxonomies. *Proceedings of the Eighth Australasian Computing Education Conference, Australian Computer Science Communications*, 28:5, 243-251.