Classifying Computing Education Papers: Process and Results

Simon
University of Newcastle
PO Box 127, Ourimbah
NSW 2258, Australia
+61 2 4348 4074
simon@newcastle.edu.au

Angela Carbone
Monash University Dandenong Rd, Caulfield East Victoria
3145, Australia
+61 3 9903 1911
angela.carbone@sims.monash.edu.au

Michael de Raadt
University of Southern Queensland
Toowoomba
Queensland 4350, Australia
+61 7 4631 5547
deraadt@usq.edu.au

Raymond Lister
University of Technology, Sydney
15 Broadway, Ultimo
NSW 2007, Australia
+61 2 9514 1850
raymond@it.uts.edu.au

Margaret Hamilton
RMIT University
GPO Box 2476V, Melbourne
Victoria 3001, Australia
+61 3 9925 2939
margaret.hamilton@rmit.edu.au

Judy Sheard
Monash University Dandenong Rd, Caulfield East Victoria
3145, Australia
+61 3 9903 2701
judy.sheard@infotech.monash.edu.au

ABSTRACT

We have applied Simon's system for classifying computing education publications to all three years of papers from ICER. We describe the process of assessing the inter-rater reliability of the system and fine-tuning it along the way. Our analysis of the ICER papers confirms that ICER is a research-intensive conference. It also indicates that the research is quite narrowly focused, with the majority of the papers set in the context of programming courses. In addition we find that ICER has a high proportion of papers involving more than one institution, and high proportions of papers on the themes of ability/aptitude and theories and models of teaching and learning.

Categories and Subject Descriptors

K.3.2 [**Computers and education**]: Computer and Information Science Education – *computer science education*; A.0 [**General**]: *conference proceedings*.

General Terms

Measurement, Reliability.

Keywords

Classifying publications, computing education, Delphi method.

Presented to *ICER'08*, September 6–7, 2008, Sydney, Australia.

1. INTRODUCTION

Having considered several different approaches to classifying publications in computing education [15, 22], Simon introduced a more comprehensive system, applying it to three years of papers from the Australasian Computing Education Conference and New Zealand's Conference of the National Advisory Committee on Computing Qualifications (NACCQ) [18] and to six years of papers from the Baltic Sea Conference on Computing Education [19].

To classify a large corpus of work in a short period of time is a non-trivial problem. Also, for academics to work as a team and agree with each other is difficult. The authors of this paper began their work with Simon's system when they met at a workshop to learn about the system, to test its reliability among multiple raters, and to apply it to further bodies of papers. We first used the system to analyse eight years of computing education papers at New Zealand's NACCQ Conference [20], and subsequently applied it to the three years of ICER papers. In this paper we report on the different processes we have used to test the reliability of the system and to achieve consensus on our classifications, and present our analysis of the ICER papers, which is effectively a profile of ICER. We have used various methods to reach consensus, culminating with the Delphi method [16, 26], which allows for several iterations through the classification exercise, with each iteration being informed by the justifications provided in the former iteration.

2. THE CLASSIFICATION SYSTEM

Simon's system classifies papers according to four dimensions, where a paper's classification in one dimension is independent of its classification in the others.

The *context* dimension identifies the subject in which a paper is set; the *theme* dimension identifies what the paper is actually about; the *scope* dimension gives some measure of the breadth of

the context; and the *nature* dimension extends the well-known notion of research papers and practice papers, without explicitly judging the values of different approaches.

Simon's system has been described and explained elsewhere [18, 19]. In particular, these earlier papers survey prior classification systems and explain the justification for a new system. It is therefore not appropriate to repeat that survey and explanation here. In the following subsections we give a brief explanation of each dimension, using references to papers from ICER that illustrate aspects of the dimension being described.

2.1 The Context Dimension

Most computing education research is set in the context of a particular subject. (In this paper we use the word 'subject' to indicate a unit of teaching for which students achieve a formal result. Such units are also called courses, papers, and indeed units.) The context dimension does not identify the subject by name and/or code, such as INFT3940 Information Technology Applications, but does identify the general thrust of the subject. Thus the ICER paper entitled *What novice programmers don't know* [13] has a context of programming, while the paper *Narrating data structures: the role of context in CS2* [25] has a context of data structures, and *Uncovering student values for hiring in the software industry* [2] has a context of professionalism and ethics.

The list of possible contexts is as broad as the list of possible subject areas in computing, and could be based on various existing compilations of these subject areas such as syllabus proposals. Simon's initial list was based simply on the subjects covered in the corpus that he first studied. In our analysis of NACCQ papers [20] we added a small number of subjects to that list; but we subsequently found that the ICER papers are fully covered by a greatly reduced list.

Not every computing education paper is set in the context of a particular subject. Some, such as this paper, are set in the literature, and others, such as *Through the eyes of instructors: a phenomenographic investigation of student success* [11] are not set in an identifiable subject, or are set across a range of subjects. To cater for such papers, the system supplements the list of possible contexts with *literature* and *broad-based*.

2.2 The Theme Dimension

The theme of a paper is what the paper is actually about. When we began to use Simon's system, it took some effort to distinguish this dimension from the context dimension. For example, while *What novice programmers don't know* [13] is set in the context of a first-year programming subject, the paper is actually about students' ability and/or aptitude; and while *Narrating data structures: the role of context in CS2* [25] is set in the context of a data structures subject, the paper is actually about a particular teaching technique used in that subject.

Most prior classifications of papers [15, 22] and of research [5] recognise the prevalence of papers describing tools that have been developed to assist with teaching or with assessment. Papers of this sort will have themes of teaching/learning tools [9] and assessment tools respectively.

In our prior study we identified nearly 20 distinct themes, including, for example, credit for prior learning, online/distance delivery, employment, gender issues, language/culture issues, tutors and tutoring; but as we shall see, only 10 themes were evident in the ICER papers.

2.3 The Scope Dimension

A paper's scope gives some sort of measure of the breadth of community involvement that it entailed. A paper set in a single subject could in principle be written in isolation from the computing education community, whereas a study involving researchers and participants across multiple institutions necessarily entails a broad community involvement.

The four substantive values of scope are

- subject (eg *Problems encountered by novice pair programmers* [10]);
- program/department (eg *Impact of alternative introductory courses on program concept understanding* [21]);
- institution (eg *Attitudes towards computer science – computing experiences as a starting point and barrier to computer science* [17]);
- many institutions (eg *Strategies that students use to trace code: an analysis based in grounded theory* [7]).

In addition, there are papers that have no recognisable scope, and for these there is a 'not applicable' scope. This paper would be an example, as would any paper within the context of literature. Another example would be *On models of and for teaching: toward theory-based computing education* [3], which is not based on work conducted in any of the specific scopes.

2.4 The Nature Dimension

There is a fairly well-known distinction between 'practice' and 'research' papers. The nature dimension recognises this distinction and takes it further.

An *experiment* paper is one that reports on a scientific-style experiment, with control and experiment groups and controllable variables. Papers of this sort are understandably rare in education, as it is not generally feasible (or ethically justifiable) to split a class into groups and treat each group differently. Even so, examples can be found, such as *Pattern oriented instruction and the enhancement of analogical reasoning* [14].

A *study* paper reports on an experiment in the looser sense of the word. A hypothesis is formed, a study is devised and conducted to explore the hypothesis, and data is gathered from the study and analysed. An example is *What does it take to learn 'programming thinking'?* [4].

An *analysis* paper reports on analysis performed on pre-existing data, such as students' results in a course over several years. A hypothesis is formed and the existing data is analysed to explore the hypothesis. An example of this would be *Warren's question* [6], which analyses postings to a mailing list.

A *report* paper focuses on informing the reader about something that was done, typically in the classroom. Report papers often describe innovations, and sometimes describe the adoptions of innovations already reported elsewhere. An example is *What do students know? An outcomes-based assessment system* [24].

Finally, a *position/proposal* paper presents a position that outlines the authors' beliefs on a particular matter, or describes a proposal

to carry out some work. In either case, no work has yet been carried out, so such papers do not report any results. *On models of and for teaching: toward theory-based computing education* [3] is an example of a position paper.

Following Simon's lead, we consider experiment, study, and analysis papers to be 'research' papers. The other categories are not so unequivocal, but we generally classify reports as practice papers, and position/proposal papers as perhaps not even that.

3. THE METHOD

The work reported in this paper stems from a two-day workshop conducted in association with the 2008 Australasian Computing Education Conference. The workshop began with an introduction to Simon's classification system, and proceeded to classify the papers from the eight most recent years of New Zealand's NACCQ Conference, first as an entire group, then in pairs, and finally individually. After reflecting on the method for NACCQ paper classification, the Delphi method was suggested as a means of achieving consensus during ICER paper classifications.

3.1 Classifying NACCQ Papers

The workshop introduced participants to Simon's classification system by way of the following steps.

3.1.1 Workshop discussion

The first set of papers was classified by consensus. Having familiarised ourselves with the papers prior to the workshop, we discussed each one in turn and decided on its context, theme, scope, and nature.

3.1.2 Individual classification then discussion

Once we had discussed enough papers to be familiar with the classification system, we classified a second set of papers individually, and then gathered to discuss the results. There were differences in our findings, but these differences were overcome in the discussion, resulting in a further set of agreed classifications.

3.1.3 Inter-rater reliability testing

For a third set of papers, we again classified individually, but instead of resolving our differences by discussion we noted and measured them, applying Fleiss's kappa, a standard measure of inter-rater reliability [8].

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 [1].

In this first test of inter-rater reliability, after only one day of preparation, we showed fair to good agreement on context (44%), theme (57%), and scope (54%), with poor agreement only on nature (32%).

3.1.4 Refinement of the system

As the workshop drew to a close we discussed possible improvements to the system. We added some new contexts; this is appropriate, as the list of contexts must be driven by the content of the papers being examined.

To overcome confusion on our part, we renamed as 'theme' the dimension Simon had originally called 'topic' as we found it too easy to confuse topic with context. For the same reason, we renamed the 'teaching and learning' category to 'teaching and learning theories and models'. Finally, we combined the original themes of 'recruitment' and 'employment' into a single broader theme of 'recruitment, retention, and pathways'.

Our biggest change to the system was the addition of a new category to the nature dimension. The original system had four natures: experiment, analysis, report, and position. We felt it important to distinguish between tightly controlled scientific experiments and the less controlled explorations that are more common in education, and so we split the 'experiment' nature into 'experiment' and 'study', as defined in 2.4 above. We also felt it important to recognise the distinction between position papers and proposals. Even though we left both sets of papers in the same category, we renamed it position/proposal to recognise this duality.

3.1.5 Further individual reliability testing

Following the workshop, we individually classified a further set of NACCQ papers, and again measured our agreement with Fleiss's kappa. Our agreement on context and nature was better than in the prior test, but on theme and scope it was worse.

3.1.6 Paired reliability testing

Immediately following the second set of individual classifications, we formed pairs to discuss our findings and try to agree on a common classification for each dimension of each paper. As the workshop was over by this time, and we had all returned to our respective institutions, these paired discussions were held over email, VOIP, and telephone. The intention of this phase was to see whether discussion improved the level of agreement. An underlying thought was that any careless classifications were likely to be weeded out through this discussion.

Paired discussions led to a marked improvement over the individual classifications. Regarding each pair as a single rater, we calculated Fleiss's kappa once again, with the results shown in Table 1.

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%

3.2 Classifying ICER Papers

Having familiarised ourselves with Simon's classification, and also having fine-tuned it, using NACCQ papers, we then proceeded to classify ICER in the following way.

3.2.1 Individual reliability testing

A total of 43 papers have been published at the three ICER workshops held to date. We began our exploration of these papers by classifying them individually, and once again measuring our agreement with Fleiss's kappa.

The results here were comparable with our prior individual classifications. All were in the fair to good range, but only just. The kappa values were 61% for context, 41% for theme, 54% for scope, and 42% for nature.

The kappa figure for nature was particularly intriguing. We had all observed that, unlike any of the other corpuses studied earlier, the vast majority of ICER papers are studies by nature. Intuitively we believed that by this point we had strong agreement on the nature dimension, but Fleiss's kappa accorded us fairly weak agreement. We suspect that this is because so many of the papers are classified as studies. The kappa formula, recognising that most of the population falls into this single category, gives each rater a high likelihood of picking that category through chance or laziness, and adjusts kappa accordingly.

3.2.2 The Delphi method

The Delphi method [16, 26] is a method of bringing about consensus in forecasting. In essence, it shows a group of forecasters a summary of their forecasts, along with brief justifications, then effectively invites them to reconsider their forecasts in the light of what the others have said. We decided to see whether this method might be effective in our classification.

One of us was designated as the facilitator, and gathered our individual independent classifications. If, for a given paper, four, five, or six of us had chosen the same category in a dimension, the classification of that paper in that dimension was considered to be agreed. Where we disagreed, the facilitator distributed a list of the categories that we had chosen for that dimension of that paper, along with our brief justifications. It is a principle of the Delphi method that the choices and justifications will be distributed anonymously, so that raters will not be influenced by the people they consider stronger or more knowledgeable. We then considered the list, noted other people's choices and justifications, and decided whether to change our own classifications.

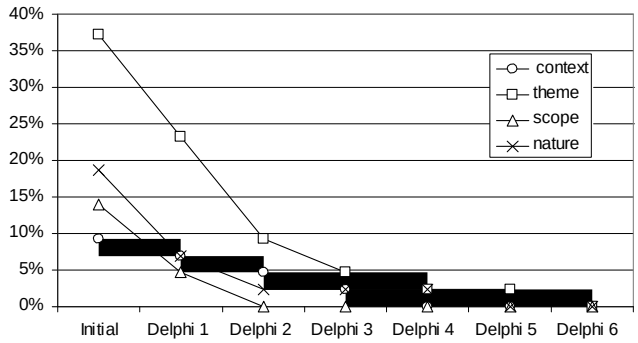

Figure 1: Proportion of classifications remaining unresolved after each round of classification

Table 2: Time for each round of classification, beginning with 132 classifications (4 each for 43 papers)

	classifications	days
initial classification	132	15
Delphi round 1	132	6
Delphi round 2	18	14
Delphi round 3	7	13
Delphi round 4	4	5
Delphi round 5	2	7
Delphi round 6	1	4

In the first Delphi round we included all 132 classifications (4 each for 43 papers), to allow for possible changes of mind. In subsequent rounds we considered only those classifications that remained unresolved, that is, for which no classification had been chosen by four or more of us.

When people did change their classifications, we asked why they had done this. Some indicated that they had made a mistake in their initial classification, or that they had been persuaded by the arguments of others. We see these as good reasons for changing, validating our choice to use this method. There were also one or two instances of people changing their classification in order to achieve consensus. This is of a little more concern, as it leaves open the possibility that a less well argued classification will prevail simply because it has more initial supporters.

There were two other minor weaknesses in our use of the Delphi system. First, some raters declined to give new justifications when sticking with the same classification, while others were taking pains to refute the arguments they disagreed with. In a related matter, some raters expressed frustration that their arguments were not being given due consideration by others. Second, while Simon's system requires that each paper be given a single classification in each category (choosing the best if there are several candidates), some raters chose to give two or more classifications to the same item, leaving the facilitator to assign the value that would be more likely to achieve consensus. While this certainly had the desired effect in one or two cases, if taken to an extreme it could leave all of the raters making multiply ambiguous classifications, thus leaving the final decision entirely in the hands of the facilitator.

As this was no longer an individual rating process, we did not measure our agreement levels with Fleiss's kappa. However, we were pleased to see how rapidly the proportion of non-agreed classifications converged to zero (Figure 1).

As with any other joint venture conducted remotely, the Delphi method is highly time-consuming. Table 2 shows how long we took to complete each round of classifications. Even though the number of classifications dropped sharply after the first round, the rounds took an average of 9 days. As the process can involve many rounds, researchers considering the use of this method should allow a great deal of time for it to run to completion.

It is clear from Figure 1 that our individual classifications were in substantial agreement on most dimensions of most papers, with theme being the exception. Even after the first pass of the Delphi

process, the themes of nearly 25% of the papers were defying agreement.

This finding leads to a fascinating conclusion: within Simon's classification system it is not always easy to determine a paper's theme. It is relatively easy to determine what sort of subject a paper is set in, how broad a community involvement it entails, and where it lies on the extended practice/research spectrum; but it's not necessarily easy to determine just what the paper is about. We find this fascinating because we imagine most authors would think it was obvious what their papers are about. A particular problem with regard to ICER papers was the difficulty in deciding whether the themes of certain papers were ability and aptitude or teaching and learning theories and models, because there can be a substantial overlap between these two themes.

4. THE RESULTS

Having been running for only three years, ICER has hosted only 43 papers. This is not a large number, so any findings must be regarded with some caution. Even so, we believe that we have discovered some features of interest, especially in comparison with other corpuses of papers that we [20] and Simon [18, 19] have analysed. While each study covered a different number of years in its corpus, for our comparisons between studies we consider only the most recent three years of papers in each study, to provide a firmer basis for comparison.

4.1 Context

The three prior studies have all shown the programming context dominating, but ICER takes this to a new extreme with 74% of all papers set in the context of programming courses. Outside this context, there was one paper in a data structures context, one in a context of professionalism and ethics, and the remaining 9 papers (21%) broad-based – that is, not set in the context of any particular subject. Table 3 shows how this sets ICER aside from the other corpuses studied.

Notwithstanding the somewhat lower number of papers in the ICER study, it is clear that ICER papers represent an extremely narrow range of contexts. It is not clear whether this reflects an ICER preference for the programming aspect of computing education, or whether only programming education gives rise to papers of the methodological quality expected by ICER.

4.2 Theme

The number of themes is not nearly so restricted as the number of contexts. The 43 papers range across 10 different agreed themes, compared with 17-19 in the other corpuses studied (Figure 2).

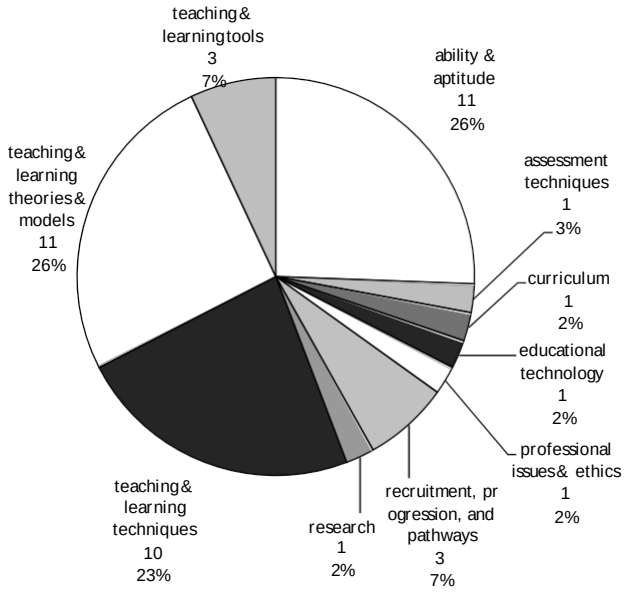

Figure 2: Themes of ICER papers

Figure 2 shows the number of papers in each theme.

Once again we see a concentration, this time in three categories. The strong position of teaching and learning techniques is something that ICER shares with all of the prior studies; educational research will surely always include explorations of the effects of different ways of teaching. But if we compare the proportions of ICER papers in these three dominant themes with the proportions in the same themes of papers from other conferences, we see once more what makes ICER distinct (Table 4).

What sets ICER apart is the proportion of its papers that deal with, on the one hand, ability and aptitude – what makes students good at computing – and, on the other, teaching and learning theories and models – how teachers teach and how students learn. The ICER workshops have far more papers in these two themes than do the other conferences studied.

4.3 Scope

Figure 3 shows the scopes of the 43 ICER papers. Comparing these proportions with those from the most recent three years of the other studies, there is again a stark difference. Thirty-three percent of the ICER papers are based on work carried out across multiple institutions, compared with 10% in the Australasian and Koli studies and 12% in the NACCQ study. If it is true that multi-

Table 3: Contexts of ICER papers compared with other corpuses studied (3 years of each corpus)

Conference	paper count	number of contexts	programming papers
Aust Comp Ed	81	12	42 (52%)
Koli Calling	63	15	26 (41%)
NACCQ	60	16	10 (17%)
ICER	43	4	32 (74%)

Table 4: ICER's dominant themes compared with other corpuses studied (3 years of each corpus)

Conference (papers)	ability/ aptitude	tch/lrn techniques	tch/lrn theories, models
Aust Comp Ed (81)	10%	22%	4%
Koli Calling (63)	13%	29%	6%
NACCQ (60)	3%	12%	3%
ICER (43)	26%	23%	26%

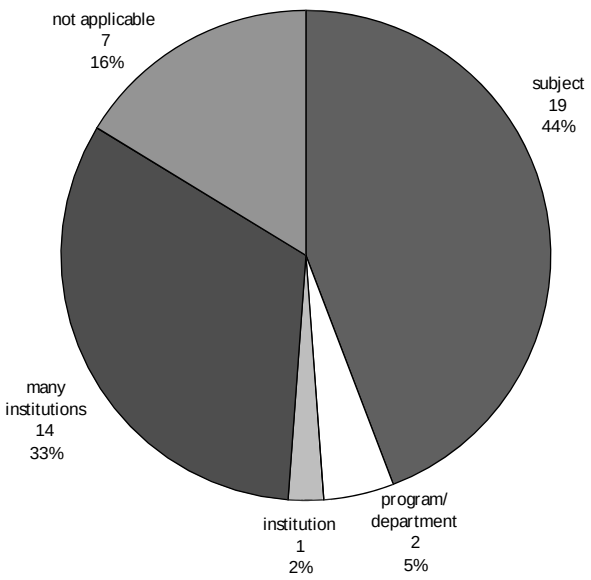

Figure 3: Scopes of ICER papers

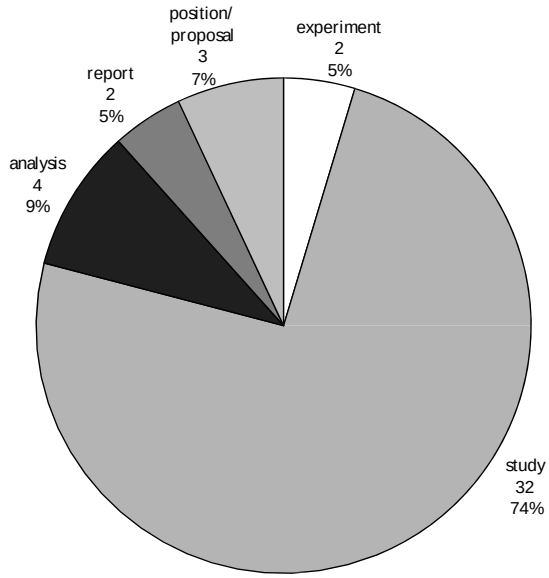

Figure 4: Natures of ICER papers

institutional papers reflect a greater involvement with the computing education community, the papers accepted for ICER show a high level of involvement with that community.

4.4 Nature

The natures of the ICER papers are shown in Figure 4. We suggested in section 2.4 that experiment, study, and analysis papers are clearly research papers, that reports are more likely to be practice papers, and that position/proposal papers are typically neither research nor practice. Following this suggestion, 88% of the papers accepted to the first three ICER workshops would generally be recognised as research papers.

In his analysis of the papers at Koli Calling [19], Simon presented a figure that showed the proportions of research papers over the previous three years at the Australasian Computing Education Conference, the NACCQ conference, and Koli Calling. In Figure 5 we extend this figure to include the same three years of ICER.

This comparison clearly shows that while each of the other conferences studied has a reasonable proportion of research papers, ICER is, as its name suggests, almost exclusively a vehicle for the presentation of research. Of course this is not a surprising result, but it is possibly a useful confirmation of what is generally assumed.

5. FURTHER ANALYSIS

Our analysis using Simon's system has aroused our curiosity about two other aspects of these papers: the types of research methods used, and where the authors come from and how readily they return to subsequent ICER workshops.

5.1 Types of Research

A total of 38 of the 43 ICER papers reported some form of research study. Our investigation of the types of research conducted in these studies involved several aspects. We considered the general methodology, data gathering methods and analysis.

We first considered whether the research was *qualitative* or *quantitative* in nature. Using a broad definition by Krathwohl [12] as cited in Wiersma [23] we defined qualitative research as research that describes phenomena in words and quantitative research as research that describes phenomena in numbers and measures.

We found that it was not always possible to make a clear distinction between qualitative and quantitative research, as some papers employed mixed methods. For example, a number of the studies were mainly quantitative but included some quotes from open-ended questions on a survey. Another study reported a statistical analysis of the frequencies of responses classified in groups but used qualitative data to form the initial categories. In

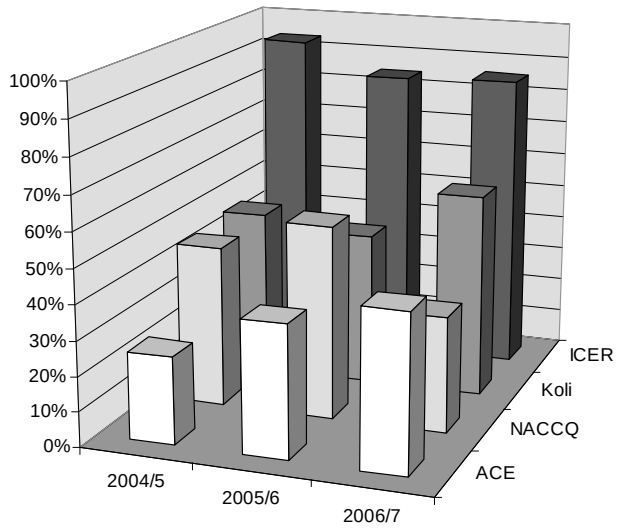

Figure 5: Proportion of papers from each conference that can be clearly classed as research papers (adapted from [19])

both these cases we used a classification of *mixed*. We found that all the papers could be classified into one of these three categories, as shown in Table 5. Some studies collected data in the form of drawings; however, these were described in words or numbers of features, so the papers fell into qualitative or quantitative classifications.

Next we considered the different types of data gathering methods used. Table 6 shows the variety of methods reported. In almost half the studies more than one method was used, with one study reporting six methods. In almost half the studies, data was gathered from set tasks carried out by students. Other common methods were questionnaires and interviews. Initially we intended to make a distinction between online and paper-based questionnaires; however, this was not possible as in most cases the medium was not specified. It is interesting to note that only two studies used an existing inventory. Lastly we considered the data analysis techniques used. Most quantitative studies included some statistical analysis beyond descriptives. But there were a few papers where only descriptive statistics were presented and the research findings would have been further supported or enhanced with some inferential statistical analysis. There were several qualitative analysis methods reported, including phenomenography, grounded theory and content analysis.

5.2 Provenance and persistence of authors

As we classified the papers we noticed how many papers had multiple authors. Defining an 'author contribution' as a single author's contribution to a single paper, we note that for the 43 papers there are 110 author contributions contributed by a total of 78 authors. Six of the papers (14%) have just one author each, 23 (53%) have two authors, 5 (12%) have three authors, 4 (9%) have four authors, 3 (7%) have five authors, and 2 (5%) have six authors.

The diversity of authorship is truly international, but with America dominating. One author contribution is from Australia, two are from New Zealand, three from Wales, four each from England, Germany, Sweden and Israel, six from Denmark, eight from Ireland, fourteen from Finland and sixty (55%) from the USA. A further breakdown to states within the USA reveals that the majority of authors come from Washington (11 papers with 8 individual authors), Georgia (11 papers with 7 individual authors) and California (8 papers with 5 individual authors). It is possibly significant that the first and third workshops were held in Washington and Georgia. We wanted to see if many authors have written several papers and continued to contribute to ICER over successive years. We found that the majority of authors (60) have contributed only the one paper to the collection of workshops. However, there were groups of authors who have continued their research over the three years. One author in particular contributed to five separate papers over the three-year period, and those five papers were written with 4, 5 and 6 authors. The work of this author shows what we perceive as a progression from investigating novice programmers, to incorporating 'commonsense computing', to an overview of a theory or model for teaching.

Of the 110 author contributions, 52 were in papers written with colleagues from the same institution, 52 in papers with authors from other countries or states, and 6 in papers written by single authors.

Table 5: Quantitative/qualitative classification (N=38)

	Qualitative	Quantitative	Mixed
2005	7 (47%)	5 (33%)	3 (20%)
2006	1 (9%)	2 (18%)	8 (73%)
2007	8 (67%)	1 (8%)	3 (25%)
Total	16 (42%)	8 (21%)	14 (37%)

6. CONCLUSIONS

We have taken Simon's system for the classification of computing education papers and explored its reliability by having multiple classifiers apply it to the same sets of papers in various ways. While we did not always agree easily, in general we conclude that the system is reasonably robust and reliable.

Applying the system to the 43 papers presented at the first three ICER workshops, we find clear evidence that

- an extremely high proportion of ICER papers (74%) are set in the context of programming subjects;
- ICER has far higher proportions of papers about ability and aptitude (26%) and teaching and learning theories and models (26%) than other conferences analysed with Simon's system;
- ICER has a far higher proportion (33%) than earlier conferences studied of papers involving multi-institutional work;
- ICER has a far higher proportion (88%) than the other conferences of papers that are unarguably research papers.

Of the research papers, we found that about 40% of papers reported qualitative analysis and about 40% reported mixed

Table 6: Data gathering methods

Data gathering method (74 methods reported in 38 papers)	Number of times used
Task	15 (20%)
Questionnaire (online or paper)	13 (18%)
Interview	12 (16%)
Formal course assessment	7 (9%)
Artefact	6 (8%)
Observation	5 (7%)
Video	2 (3%)
Inventory	2 (3%)
Test	2 (3%)
Other (focus groups, log files, screen shots, journals, online discussion, course materials)	10 (13%)

methods, while only about 20% reported purely quantitative analysis.

Of the authors, we discovered that a majority come from the USA, that nearly half collaborate with colleagues from the same institution, while as many again collaborate with authors from other states or countries.

We also discovered, along the way, the value of a good abstract. We would now be willing to define a good abstract as one that permits a reasonably reliable classification of a paper without the need to read the paper. By this criterion, we hope that the abstract of this paper would lead informed readers to conclude that it is set in the context of the literature, its theme is research, its scope is not applicable, and its nature is analysis.

7. ACKNOWLEDGMENTS

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.

8. REFERENCES

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

[2] D Chinn & T VanDeGrift (2007). Uncovering student values for hiring in the software industry. Proceedings of the 3rd International Computing Education Research Workshop (Atlanta, Georgia, USA, September 15-16 2007) ACM Press, New York, 145-157.

[3] JP East (2006). On models of and for teaching: toward theory-based computing education. Proceedings of the 2nd International Computing Education Research Workshop (Canterbury, UK, September 9-10 2006) ACM Press, New York, 41-50.

[4] A Eckerdal & A Berglund (2005). What does it take to learn 'programming thinking'? Proceedings of the 1st International Computing Education Research Workshop (Seattle, Washington, USA, October 1-2 2005) ACM Press, New York, 135-142.

[5] S Fincher & M Petre (2004). Computer science education research. London, Routledge Falmer.

[6] S Fincher & J Tenenberg (2007). Warren's question. Proceedings of the 3rd International Computing Education Research Workshop (Atlanta, Georgia, USA, September 15-16 2007) ACM Press, New York, 51-60.

[7] S Fitzgerald, B Simon, & L Thomas (2005). Strategies that students use to trace code: an analysis based in grounded theory. Proceedings of the 1st International Computing Education Research Workshop (Seattle, Washington, USA, October 1-2 2005) ACM Press, New York, 69-80.

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

[9] P Gross & K Powers (2005). Evaluating assessments of novice programming environments. Proceedings of the 1st International Computing Education Research Workshop (Seattle, Washington, USA, October 1-2 2005) ACM Press, New York, 99-110.

[10] B Hanks (2007). Problems encountered by novice pair programmers. Proceedings of the 3rd International Computing Education Research Workshop (Atlanta, Georgia, USA, September 15-16 2007) ACM Press, New York, 159-164.

[11] P Kinnunen, R McCartney, L Murphy, & L Thomas (2007). Through the eyes of instructors: a phenomenographic investigation of student success. Proceedings of the 3rd International Computing Education Research Workshop (Atlanta, Georgia, USA, September 15-16 2007) ACM Press, New York, 61-72.

[12] DR Krathwohl (1993). Methods of educational and social science research: An integrated approach. New York: Longman

[13] G Lewandowski, A Gutschow, R McCartney, K Sanders, & D Shinners-Kennedy (2005). What novice programmers don't know. Proceedings of the 1st International Computing Education Research Workshop (Seattle, Washington, USA, October 1-2 2005) ACM Press, New York, 1-12.

[14] O Muller (2005). Pattern oriented instruction and the enhancement of analogical reasoning. Proceedings of the 1st International Computing Education Research Workshop (Seattle, Washington, USA, October 1-2 2005) ACM Press, New York, 57-67.

[15] A Pears, S Seidman, C Eney, P Kinnunen, & L Malmi (2005). Constructing a core literature for computing education research. ACM SIGCSE Bulletin, **37**(4) 152-161.

[16] C Powell (2003). The Delphi technique: myths and realities. Journal of Advanced Nursing 41, 4, 376-382.

[17] C Schulte & M Knobelsdorf (2007). Attitudes towards computer science – computing experiences as a starting point and barrier to computer science. Proceedings of the 3rd International Computing Education Research Workshop (Atlanta, Georgia, USA, September 15-16 2007) ACM Press, New York, 27-38.

[18] Simon (2007). A classification of recent Australasian computing education publications. Computer Science Education 17, 3, 155-169.

[19] 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.

[20] Simon, J Sheard, A Carbone, M de Raadt, M Hamilton, R Lister, E Thompson (2008). Eight years of computing education papers at NACCQ. 21st Annual Conference of the National Advisory Committee on Computing Qualifications (NACCQ 2008), Auckland, New Zealand, 101-107.

[21] AE Tew, M McCracken, & M Guzdial (2005). Impact of alternative introductory courses on programming concept understanding. Proceedings of the 1st International Computing Education Research Workshop (Seattle, Washington, USA, October 1-2 2005) ACM Press, New York, 25-35.

[22] D Valentine (2004). CS Educational Research: A Meta-Analysis of SIGCSE Technical Symposium Proceedings. Proc. 35th SIGCSE Technical Symposium on Computer Science Education, ACM SIGCSE Bulletin, 36, 1, 255-259.

[23] W Wiersma (2005) Research Methods in Education: An Introduction. 8th ed., Needham Heights, Massachusetts: A Simon and Schuster Company.

[24] T Winters & T Payne (2005). What do students know? An outcomes-based assessment system. Proceedings of the 1st International Computing Education Research Workshop (Seattle, Washington, USA, October 1-2 2005) ACM Press, New York, 165-172.

[25] S Yarosh & M Guzdial (2007). Narrating data structures: the role of context in CS2. Proceedings of the 3rd International Computing Education Research Workshop (Atlanta, Georgia, USA, September 15-16 2007) ACM Press, New York, 87-97.

[26] F Yetim & M Turoff (2004). Structuring communication processes and enhancing public discourse: the Delphi method revisited. Proceedings of the 9th International Working Conference on the Language-Action Perspective on Communication Modeling (LAP 2004), Rutgers University, New Jersey, 235-251.

APPENDIX: Our full classifications of all 43 ICER papers

Author 1	Title	Theme	Context	Scope	Nature
ICER 2005					
Lewandowski	What novice programmers don't know	ability/aptitude	programming	many institutions	study
Weidenbeck	Factors affecting the success of non-majors in learning to program	ability/aptitude	programming	subject	study
Tew	Impact of alternative introductory courses on programming concept understanding	ability/aptitude	programming	program/ department	study
Ben-David Kolikant	Students' alternative standards for correctness	teaching/learning theories & models	programming	many institutions	study
Hundhausen	Personalizing and discussing algorithms within CS1 studio experiences: an observational study	teaching/learning techniques	programming	subject	study
Muller	Pattern oriented instruction and the enhancement of analogical reasoning	teaching/learning techniques	programming	subject	experiment
Fitzgerald	Strategies that students use to trace code: an analysis based in grounded theory	ability/aptitude	programming	many institutions	study
Bergin	Examining the role of self-regulated learning on introductory programming performance	ability/aptitude	programming	subject	study
Gibson	Software engineering as a model of understanding for learning and problem solving	teaching/learning theories & models	programming	not applicable	study
Gross	Evaluating assessments of novice programming environments	teaching/learning tools	programming	not applicable	analysis
Fincher	Multi-institutional, multi-national studies in CSEd research: some design considerations and trade-offs	research	broad-based	many institutions	analysis
Ihantola	Taxonomy of effortless creation of algorithm visualizations	teaching/learning tools	programming	not applicable	study
Eckerdal	What does it take to learn 'programming thinking'?	teaching/learning theories & models	programming	subject	study
Schulte	Novices' expectations and prior knowledge of software development – results of a study with high school students	ability/aptitude	broad-based	many institutions	study
Bennedsen	An investigation of potential success factors for an introductory model-driven programming course	ability/aptitude	programming	subject	study
Winters	What do students know? An outcomes-based assessment system	assessment techniques	broad-based	not applicable	report
ICER 2006					
Ebel	Affective effects of program visualization	teaching/learning techniques	programming	subject	study
Nevalainen	An experiment on short-term effects of animated versus static visualization of operations on program perception	teaching/learning techniques	programming	subject	experiment
Schulte	What do teachers teach in introductory programming?	curriculum	programming	many institutions	study
Simon	Commonsense computing: what students know before we teach (episode 1: sorting)	ability/aptitude	programming	many institutions	study
East	On models of and for teaching: toward theory-based computing education	teaching/learning theories & models	broad-based	not applicable	position/ proposal
Guzdial	Imagineering inauthentic legitimate peripheral participation: an instructional design approach for motivating computing education	teaching/learning techniques	broad-based	program/ department	report

Author 1	Title	Theme	Context	Scope	Nature
Hundhausen	A methodology for analyzing the temporal evolution of novice programs based on semantic components	teaching/learning tools	programming	subject	study
Jadud	Methods and tools for exploring novice compilation behaviour	teaching/learning theories & models	programming	subject	study
Byckling	A role-based analysis model for the evaluation of novices' programming knowledge development	teaching/learning theories & models	programming	subject	study
Kinnunen	Why students drop out CS1 course?	recruitment, progression, and pathways	programming	subject	study
Stamouli	Object oriented programming and program correctness: the students' perspective	teaching/learning theories & models	programming	subject	study
Koile	Improving learning in CS1 via tablet-PC-based in-class assessment	educational technology	programming	subject	study
Dorn	Graphic designers who program as informal computer science learners	ability/aptitude	programming	not applicable	study
ICER 2007					
Sajaniemi	A study of the development of students' visualizations of program state during an elementary object-oriented programming course	teaching/learning theories & models	programming	subject	study
Kaczmarczyk	Challenging the advanced first-year student's learning process through student presentations	teaching/learning techniques	programming	subject	study
Schulte	Attititudes towards computer science - computing experiences as a starting point and barrier to computer science	recruitment, progression, and pathways	broad-based	institution	study
Yardi	What is computing? Bridging the gap between teenagers' perceptions and graduate students' experiences	recruitment, progression, and pathways	broad-based	many institutions	study
Fincher	Warren's question	teaching/learning theories & models	broad-based	many institutions	analysis
Kinnunen	Through the eyes of instructors: a phenomenographic investigation of student success	ability/aptitude	broad-based	many institutions	study
Simon	First year students' impressions of pair programming in CS1	teaching/learning techniques	programming	many institutions	study
Yarosh	Narrating data structures: the role of context in CS2	teaching/learning techniques	data structures	subject	study
Gray	Suggestions for graduated exposure to programming concepts using fading worked examples	teaching/learning techniques	programming	not applicable	position/ proposal
Caspersen	Instructional design of a programming course — a learning theoretic approach	teaching/learning theories & models	programming	subject	position/ proposal
Eckerdal	From limen to lumen: computing students in liminal spaces	teaching/learning theories & models	programming	many institutions	study
Lewandowski	Commonsense computing (episode 3): concurrency and concert tickets	ability/aptitude	programming	many institutions	study
Chinn	Uncovering student values for hiring in the software industry	professional issues and ethics	ethics	many institutions	analysis
Hanks	Problems encountered by novice pair programmers	teaching/learning techniques	programming	subject	study