Informatics in Education, 2007, Vol. 6, No. 1, 215–230

Variation in Approaches to Lab Practical Classes among Computing Academics

SIMON
School of Design, Communication, and Information Technology, University of Newcastle
PO Box 127, Ourimbah, NSW 2258, Australia
e-mail: simon@newcastle.edu.au

Michael de RAADT
Department of Mathematics and Computing, University of Southern Queensland
Toowoomba, Qld 4350, Australia
e-mail: deraadt@usq.edu.au

Anne VENABLES
School of Computer Science and Mathematics, Victoria University
PO Box 14428, Melbourne, Vic 8001, Australia
e-mail: anne.venables@vu.edu.au

Received: August 2006

Abstract. As part of a wide-ranging phenomenographic study of computing teachers, we explored their varying understandings of the lab practical class and discovered four distinct categories of description of lab practicals. We consider which of these categories appear comparable with non-lecture classes in other disciplines, and which appear distinctive to computing. An awareness of this range of approaches to conducting practical lab classes will better enable academics to consider which is best suited to their own purposes when designing courses.

Key words: computing education, phenomenography, lab practical class.

1. Introduction

Across academic disciplines, lecturing remains the prevailing teaching method undertaken in Australian universities (Ballantyne *et al.*, 1999). The academic stands before a large group of students and talks to them, aided perhaps by a board, a slide presentation, or a choice of other props. The effectiveness or otherwise of this type of instruction continues to be subject to debate (Bligh, 1971; McInnis, 1999; Carter and Boyle, 2002). Views vary from those of McInnis (1999) *"Lecturing is about as effective at delivering information as other methods"* to that of Carter and Boyle (2002), who seriously question whether students learn *"by simply being present in a room whilst someone stands at the front and attempts to transmit information to them, particularly with a practical subject such as IT"*. In their study of exemplary teaching practice of 708 practising university teachers, Ballantyne *et al*. (1999) found that much good teaching and learning goes on in classes other than lectures. They found small group discussions and tutorials to be very common in the humanities, law and social sciences, and the use of practical sessions prominent in health, agriculture and other science disciplines.

It is our understanding that in a discipline such as history, a tutorial is typically the venue for students to discuss aspects of the topics that were covered in recent lectures. In a discipline such as mathematics, a tutorial is where students practise the techniques that they have been shown in recent lectures. In a discipline such as chemistry, a practical session is where students learn laboratory techniques and conduct experiments that supplement, rather than recapitulate, lecture material.

Many computing courses have lab sessions, although some academics call them workshops and others call them tutorials. What is the role of these classes? Are they like history tutorials, like mathematics tutorials, like chemistry labs? Or are they something different, unique to computing education?

As one aspect of a wide-ranging phenomenographic study of computing academics, their understandings of lab practicals were isolated and analysed. This analysis sheds new light on these classes, suggesting strongly that they have varying roles in computing education, some of which are unlike the roles of tutorials or lab classes in other academic disciplines.

1.1. *Computing Lab Practicals Defined*

The terminology of classes is not consistent among computing academics, so the term *lab practical classes* or *practicals* is used here to mean classes in a computing lab in which students work at computers to learn the use of a software tool, device, programming language, or similar, with tutors at hand to assist them in learning to use that tool. This is quite distinct from lectures, in which students sit and watch while a lecturer explains or demonstrates the material to be taught. It is also quite different from the use of computers for teaching in other disciplines. There are many examples of computer-assisted learning (CAL) such as the use of computer simulations to replace wet laboratory sessions in medical and pharmacological training, or where software acts as an electronic tutor for physics and mathematics students (Sewell *et al.*, 1996; Reif and Scott, 1999; Hughes, 2000; Ward *et al.*, 2001; Ruthven and Hennessy, 2002). In these cases, the computer is simply a vehicle being used to assist the student achieve a non-computing learning outcome.

Azemi (1995) describes an approach where lab practicals and lectures are combined within a computing course. This approach yielded positive feedback from students and faster learning was observed, albeit at the cost of significantly greater effort from instructors. Simon (2003) describes a similar approach using VET (Vocational Education and Training) teaching: “While a university subject will typically be taught with lectures to the full class followed by labs or tutorials for groups of 20 or so students, all VET teaching takes place in classes of 20 or so students. Each class is like a combined lecture and tutorial, and there is no analogue of the university lecture.” Approaches such as these, while clearly of interest, do not fall within the scope of this study.

1.2. *The Phenomenographic Process*

A phenomenographic study begins with interviews of a number of subjects. The interview transcripts are then analysed to discover different ways that the subjects understand the same phenomenon. It is the contention of phenomenography that for any phenomenon there is only a small number of possible understandings, which are called *categories of description*, and that the understanding of any individual will fit into one or more of these categories. It tends also to be the case that for a given phenomenon, the categories of description are hierarchical. Commonly, the understanding of the novice will generally fit into the simplest category. As people become more familiar with the phenomenon, they will often progress to higher-level understandings, which will generally still encompass those at the lower levels. In such a case, the highest level of understanding, which encompasses all of the lower levels, will be in some sense a true and complete understanding of the phenomenon.

As important as categories of description are *dimensions of variation*, individual aspects of the phenomenon in which a range of values are found. These values are not in themselves different ways of understanding the phenomenon, but it is generally the case that a category will be associated with a set of comparable values across a number of dimensions.

One approach to a phenomenographic analysis is to look for dimensions of variation and the distinct values within each dimension; then to see what different apparent understandings of the phenomenon emerge when the researchers combine, say, the low-level values of each dimension, then the medium-level values of each dimension, then the high-level values of each dimension.

Another approach is to start by eliciting the different categories of description, perhaps somewhat holistically, and then to observe which values of each dimension appear to correspond with each category.

A third approach, as described by Åkerlind (2005) in her excellent walk-through of the phenomenographic process, is to cycle between considering the categories of description and the dimensions of variation.

Regardless of which approach is taken, it will involve many iterations, and its outcome can often be expressed in a table whose rows are the categories of description that have emerged and whose columns are the dimensions of variation, showing which value each dimension displays for each category.

1.3. *A Phenomenographic Study of Computing Academics*

As expounded by Marton (1986), phenomenography is a valued tool for qualitative research in the social sciences, but it is not yet widely used in computing education research. Carbone and Kaasbøll (1998) point out that such studies are very time-consuming and most computer science teachers do not have the necessary training to conduct them.

In early 2006, Raymond Lister, Anders Berglund, Ilona Box, Chris Cope, and Arnold Pears conducted a workshop on Phenomenography in Computing Education Research

(PhICER). The workshop was conducted immediately prior to the Eighth Australasian Computing Education Conference, and is described in overview in Lister *et al.* (2007).

Prior to the workshop, each participant was required to read a number of papers on phenomenography in practice and its application in computing education, to interview at least one computing academic, following a fairly general and wide-ranging script, and to transcribe the interview.

Interviewees were asked to speak about just one course, perhaps the one that they most enjoyed teaching, and were encouraged to speak freely and at length. The first questions, intended to elicit their approach to learning, covered such things as what they want the students to learn in the course, whether they explicitly discuss links between these things and the profession they expect the students to take up, and what problems students have with the course.

Next they were asked what distinct ways they present learning material to students, such as lectures, tutorials, website, email, etc. For each **X** of these ways, they were then asked:

- Is there a typical structure to your **X**'s? Why do you do it that way?
- Is there something distinctive about your **X**'s, compared with other **X**'s in the department/school?
- Do you expect students to do any preparation prior to **X**'s? How do you encourage this? Why do you think it is important that students do this preparation?
- Can you give an example of an **X** which was more effective than most? Why was it more effective?
- Can you give an example of an **X** which was less effective than most? Why was it less effective?
- Can you imagine an alternative approach to make your least effective **X** better? For example, you might restructure it or present it in a different format such as a lab or a tute.
- Do you think it is appropriate for students to talk among themselves as they do an **X**? Why? What opportunity do you provide to support this?
- What sorts of thing do you expect your students to be able to do when they finish an **X**?
- What are the main problems students have with your **X**'s?
- How do your **X**'s link with your other (non-**X**) presentations of learning material?

The interview went on to ask about distinct ways of assessing the students, followed by a comparable bank of questions for each assessment method.

The goal of phenomenography is to elicit as broad as possible a range of understandings, not to categorise differences between subsets of the population, so no demographic details were collected. We do know that our interviewees included younger and older academics, male and female, from universities and technical institutes, from at least five countries (Australia, New Zealand, Finland, Ireland, USA); but nothing in our collected data indicates which is which.

The interview script was based closely on one used by Kutay and Lister (2006) in an earlier study. Although there is some difference between the two scripts, there is also substantial overlap, and the interviews from that study were included with those specifically gathered for the PhICER workshop. In all, 25 transcripts, anonymised and identified by a code, were brought to the workshop as data.

The body of the workshop, which ran for two days, consisted of some formal instruction in phenomenography and a great deal of analysis of the transcripts. By the end of the first day, participants had formed four groups, each working on a different phenomenon to be found in the transcripts. Analysis continued for some time after the end of the workshop, and indeed still continues. The results are described in overview in the previously mentioned paper (Lister *et al.*, 2007).

1.4. *Exploring Lab Practical Classes*

This paper presents in detail the results of one group which concentrated on the specific parts of the transcripts that deal with computing lab practical classes, as defined in Section 1.1 above.

Of the 25 transcripts, only 10 made any reference to what we have called a lab practical class. Some referred to clearly non-practical classes such as classroom tutorials without computers, and some made little or no mention of any non-lecture classes.

The question that we asked as we began our exploration of the transcripts is "What are the variations in lecturers' experience of laboratory practical sessions in IT?"

2. Dimensions of Variation

We opted to start our analysis by looking for dimensions of variation, feeling that this might be easier than trying immediately to elicit categories of description. Following our examination of those parts of the transcripts that deal with practical classes, three clear dimensions of variation emerged: the level of preparation expected of the students; the links with lectures or other means of presentation; and the extent to which students are responsible for their learning.

Several other candidate dimensions of variation were discarded, either because we could find too few interview excerpts to give them credence, or because there was little or no variation, with most or all of the excerpts illustrating the same value.

It is usual when presenting phenomenographic results to illustrate each value of dimension of variation with quotations from the transcripts. We believe that the dimensions and their values are reasonably self-explanatory, and have chosen to keep the illustrative quotations for Section 3, where the different values of each dimension are combined to explain the more holistic categories of description.

2.1. *Preparation Expected of the Student*

One of the questions in the interview script asked how much preparation the academic expected students to do prior to any type of class. The responses to this question showed distinct variation in the amount of preparation that academics expect their students to undertake prior to a practical class; this dimension of variation had four values, as seen in Fig. 1.

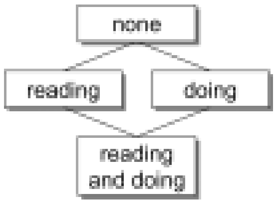

Fig. 1. Levels of preparation expected of students.

2.2. *Links to Lecture or other Facets of the Course*

Another interview question asked how each type of class linked to each other type of class. The responses gave rise to a second dimension of variation, the relationship between lab practicals and lectures or other means of teaching; again we found four values (see Fig. 2).

2.3. *Student Responsibility for Learning*

A third dimension of variation was not related to any specific interview question, but was teased out from everything that the respondents had to say about their lab practical classes. This dimension, with three values, perceives the level of student responsibility for learning in a practical class as one of the three values shown in Fig. 3.

It has been suggested to us that this dimension of variation is a phenomenon in its own right. While this might indeed be the case, that does not prevent it from being a dimension of variation of the phenomenon that we are studying. To illustrate with a simple example, rain is undoubtedly a phenomenon, and one of which there are many different understandings. Yet if one were to conduct a phenomenograhic study of holidays, rain might well arise as a dimension of variation, ranging from ‘none’ to ‘almost constant’.

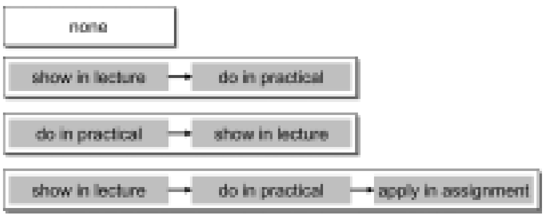

Fig. 2. Links between facets of the course.

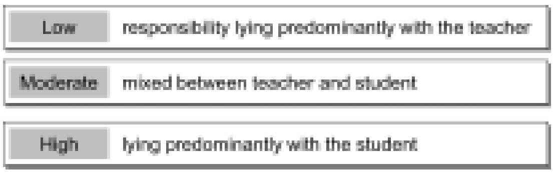

Fig. 3. Student responsibility for learning.

3. Categories of Description

Armed with these dimensions of variation, it was possible to identify four categories of description of computing practical classes. As novice phenomenographers, we initially thought of these as distinct understandings of lab practicals. As our own understanding has matured, assisted by feedback from the PhICER leaders, we have come to appreciate that our categories might more accurately be described as four different *approaches* to lab practicals in computing education. We address this distinction in the conclusion, and crave the indulgence of experienced phenomenographers if we use the phenomenographic lexicon a little too loosely between now and then.

Fig. 4 summarises our findings and illustrates how the dimensions of variation combine to produce the categories of description.

The categories of description are explained in more detail in the remainder of this section, illustrated by interview excerpts that are identified by the codes of the transcripts from which they are taken.

3.1. *The Lab Practical as a Class where Students Acquire and Practise Skills Independent of Concepts Covered in Lectures and Assignments*

In the first category of description, academics perceive the practical class as somewhat independent of lectures. While the lectures will deal with the theory component of the course, the practicals are where the students learn about, acquire, and practise specific skills that form a distinct and independent practical component.

Because of this independence from the lectures or textbooks, little or no preparation is required for these practicals. Students are not even required to do prior reading.

"There is no textbook that tells them what DreamWeaver is about. How do you learn about DreamWeaver unless you actually put your hands on and do it? They learn very quickly without reference to textbooks." [I1].

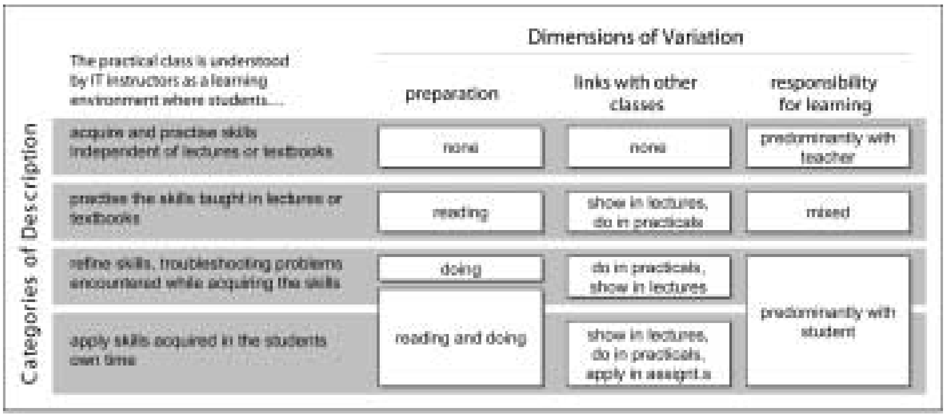

Fig. 4. Categories of description of IT instructors' experience of practical classes.

For obvious reasons, the links between practicals and other classes are essentially non-existent.

"There's nothing particular in the labs that reflects back on general lecture material. Because the labs are primarily focused on the Haskell language, it's obviously related to any Haskell lectures I give, which is early in the semester, so there's a kind of one-to-one correspondence there. But there's not a great deal of correspondence to the general material or conceptual material that's spread widely in [the course] because the labs are really focused on mainly learning a brand new programming paradigm, which is only one part of the whole course. So there's not a great deal of cross-linking." [L1].

In this category, the teacher tends to assume the primary responsibility for the learning experience, from which it often follows that the class is highly structured.

"I try to always have an amount of questions that will fill their lab sufficiently... Some concepts I'd make them do loads of different examples to really hammer home what's going on... The lab sheets start off with a couple of examples to get them going, and then a couple of easy questions to get things started... If you give them little problems initially it helps overcome the "I can't do this" fear that some students have... I check in on every lab, and if there is anything causing difficulty, I'll do my best to banish it straight away... I try to get in early and make sure there are no obstacles to learning." [M1].

3.2. *The Lab Practical as a Class where Students Acquire and Practise Skills Taught in Lectures or Textbooks*

In the second category of description, academics view the practical class as the means for students to put into practice the skills that they have been taught in the lecture or the textbook. The lectures, for example, will be used to teach and demonstrate a particular skill; then in the practical class, students will be given exercises in the application of that skill.

In this category, the academic tends to expect the student to spend some time preparing for the practical – at the very least, attending the lecture or reading the relevant tasks before arriving.

"I suggest that they read the descriptive part of each of the labs. . . " [L2].

The link with other facets of the course is stronger in this category.

"And the tutorials, as I said previously, are the theory, and show us how to create a normalised design, and in the labs we implement that design and then we carry it further, with forms and reports and code and stuff." [E4].

Responsibility for learning is no longer primarily the teacher's; instead the students take up some of that responsibility.

"It's possible that if they're under-prepared they don't get that much out of it. In other words, if they under-prepare they don't complete all of the exercises . . . I've got this set of exercises for each lab, and they should be able to complete them in a two-hour period, I believe. If they don't, if they're under-prepared then they <u>may</u> finish them..." [L1].

3.3. *The Lab Practical as a Class where Students Refine and Troubleshoot Skills Acquired in their own Time*

In the third category of description, academics expect the students to do the bulk of the skill acquisition in their own time, and perceive the practical as a class in which students are provided with help on aspects of the work that they have found problematic.

In this category the student is expected to do significant preparation for the practical; or rather, to spend significant time working to acquire the skills in question, so that the practical can be a productive troubleshooting session.

"Well, I really like it if they do some themselves. Two things I expect beforehand. First of all... I encourage them... to work through the whole of the textbook so that when they come to the tutorials [note that E4 uses the wowd 'tutorials' for classes we have categorised as lab practicals], they're just doing the exercises that I've set them. And, if possible, they can do the exercises before the tutorial; then they only need to come to the tutorial and ask about anything they had trouble with, and they can perhaps go home early." [E4].

While the link between lectures and practicals is essentially the same as in the previously defined category, there is sometimes an additional inverse link, where problems that arise in the practical are resolved in the lecture.

"It was a mutiny. I had demonstrators coming back to me saying 'You have to change this lab, they are going nuts in there... It was as if the very use of the word 'recursion' terrified them...' I had to salvage this case in the lecture. I dug up a few of the solutions that I had been provided with by students and showed them... and then a student would go 'That's recursion!' When they saw that, they seemed to realise 'Hang on, this is actually easier than we thought.' " [M1].

Responsibility for learning is now predominantly the student's.

"Some students will have done all the questions, and come in ready with their questions, the ones they had trouble with. Other students won't have done anything, and they'll start working... Everyone's working at their own speed, covering the material. Some students will do all the questions, some won't. It depends how much they're willing to do beforehand at home." [E4].

3.4. *The Lab Practical as a Class where Students Apply Skills Acquired in their own Time*

In the fourth category of description, the emphasis moves from acquiring the skills to applying those skills. The troubleshooting assistance is still provided, but in the context of applying the skills to a particular task such as a project or a major assignment.

As with the previous category, students are expected to acquire the skills in their own time (or perhaps in earlier practical sessions) so that this practical can be devoted to work on a project. The practical is now of less importance than the prior work, and can indeed become optional.

"They have to have completed the previous week's work. They have either a worksheet or a manual but they had to complete the week's work before the next week.." [O1].

The link between lectures and practicals is the same as in the previous category, but the link between practicals and assignments now becomes explicit.

"*Well they're parallelled, completely. Each lecture refers directly to a tutorial [lab practical], which refers directly to an assignment. So they're all linked, and it's very obvious what the links are... The tutorial is the glue, if you like, between the assignment and the lecture material. It relates directly to the implementation or transfer of the material presented in lectures to an assignment situation.*" [E3].

Responsibility for learning is now almost entirely the student's, with the academic providing few or no instructions.

"*. . . The following four [classes] are, as I say, basically one-liners, saying implement the philosophies and material from the [lectures]; for example, it might have been on help, it might have been on how to implement pop-up help in a web [page], so the tutorial might just say 'implement pop-up help in your assignment'. And that's it; that's what the tutorial says... I'm trying to wean them off, as much as possible, specific instructions on how to do a particular job, and get them to think about how it should be done.*" [E3].

4. Conclusions

Three dimensions of variation in IT academics' understandings of practical classes have emerged through phenomenographic study. Through analysis of those dimensions of variation, four categories of description of the practical class have been identified.

4.1. *Similarity with Prior Work*

By way of validation, similarities with prior work were sought and found with ease. There is a good deal of recent research focusing on university academics' conceptions of and approaches to teaching, along with the impact upon student learning of these conceptions and approaches. In summarising several studies from multiple disciplines across differing institutions, Åkerlind (2004) noted two striking commonalities in the key dimensions of meaning that teaching has for university teachers. The first dimension focused on the "*transmission of information to students or the development of conceptual understanding in students*" and the second focus was towards "*the teachers and their teaching strategies or the students and their learning and development.*" Building upon earlier work by Kember (1997), Åkerlind proposed four descriptions of experiences that resonate strongly with the 'responsibility for learning' dimension of variation in our study. She posited a four-valued hierarchical shift in focus:

- focus on knowledge transmission by the teacher;
- focus on teacher-student relations;
- focus on student engagement; and
- focus on student learning.

Two years earlier, McKenzie (2002) had reflected that university teachers should aspire to using approaches focused on student learning since these experiences encourage students to take deeper approaches to their learning, approaches that are often associated with higher-quality learning outcomes.

In an earlier qualitative study of teaching and learning, Fox (1983) delineated four personal theories of teaching, which have been paralleled by more recent studies such as that by Prosser *et al.* (1994).

At Fox's lowest level, which he calls *transfer*, the student is seen as a container into which the discipline knowledge is to be poured. Our first category of description, in which the students are taught new skills that are independent of lecture material, seems reasonably consonant with this theory.

At his next level, *shaping*, the student is viewed as a raw material to be shaped into a finished product whose specification is couched in terms of the discipline knowledge. It is tempting to relate this to our second category, in which the lab practical is where students acquire and practise skills they have been shown in other classes such as lectures; but the link is perhaps a little tenuous.

In the third level Fox moves the focus from the content to the student. At this level, *travelling*, the discipline knowledge moves somewhat into the background, as the countryside of a journey; the teacher's task is to guide the student through this countryside, pointing out features of interest along the way. This ties in well with our third category, in which students already have much of the knowledge, but are still being guided in its correct use.

At the fourth level, *growing*, the student is seen as already full of knowledge, the teacher's task being to cultivate that knowledge in the student, weeding and fertilising as appropriate. This ties in well with our fourth category, in which students already have the knowledge and skills, and seek help only in occasional aspects of their application.

The categorisation emergent from our study clearly ties in quite well with earlier theories, thus giving some validation. At the same time, if this work is to be anything more than a replication of earlier studies, its distinct and novel aspects need to be exposed.

4.2. *Differences from Prior Work*

The novel outcome from this work is the indication that, while computing lab practical classes are generally thought of as somewhat uniform, there are in fact a number of diverse approaches that appear to tie in with equally diverse educational purposes. In addition, some of these approaches appear distinctive to computing education. In the first instance, this can perhaps be best explained by referring back to the non-lecture classes of other disciplines, as mentioned briefly in the introduction.

Tutorials in, say, a mathematics course are generally intended for students to practice skills and methods acquired in lectures and/or textbooks. They thus fall neatly into category 2.

Tutorials in many humanities courses are for discussion of the topics that have been presented in lectures and/or texts. If they are for practice at anything, it would be at analysis and argumentation. With that interpretation, they probably fall into category 1, a class where students acquire and practice skills independent of concepts covered in lectures. Alternatively, if analysis and argumentation are explicitly taught in lectures, these tutorials too would fall into category 2.

Lab classes in some of the physical sciences appear to fall into category 1. In chemistry, for example, theoretical aspects of the discipline are taught in lectures. In the labs, students follow tightly defined procedures to learn new techniques, and either discover properties that have not been addressed in lectures (category 1) or confirm properties that have been covered in lectures (category 2).

We have had to search rather harder to find tutorial or lab classes that fall into category 3 or category 4. We have not yet been able to verify this in the literature of other disciplines, but pending a thorough investigation, it appears to us that these categories are more or less exclusive to the creative disciplines. In art, design, music, and architecture, for example, we would expect to find classes where students apply their creative skills, with a tutor on hand to guide and assist rather than to show the way. It seems, therefore, that the presence in computing education of classes where students refine and troubleshoot (category 3) or simply apply (category 4) skills that they have already acquired confirms the often-argued position that computing is as much a creative discipline as it is an analytic one.

So is computer science different, and why are there so many approaches in the practical sessions? The discipline of computer science can be likened to a craft that must be acquired and practised. For instance, programming novices must begin with introductory exercises allowing them to acquire basic skills (Lister and Leaney, 2003) before they can become proficient enough to move onto independent problem solving. Indeed, much has been written on the difficulties faced by students in learning programming and on how best to teach the art (Hagan and Sheard, 1998; Davy and Jenkins, 1999; Jenkins, 2002). According to Carter and Boyle (2002), this is just one of the discipline-specific delivery problems that is unique to computer science education. Other issues to address include managing a curriculum that is diverse and ever-changing. This aspect is particularly challenging when many computer science departments are historically composed of staff who have migrated to the discipline. Coupled with these problems are wildly fluctuating student numbers and varying academic interests and abilities within the student population (Venables *et al.*, 2006). So perhaps it is not so surprising to find differing dimensions of variation and categories of description amongst the transcripts analysed for this paper.

4.3. *Is it Phenomenography?*

The findings presented here are both interesting and significant. There remains a question, though, as to whether they (yet) represent phenomenography.

Phenomenography is clearly and explicitly designed to elicit different ways of experiencing a particular phenomenon. Cope (2000) conducted a phenomenographic survey to discover students' understandings of an information system. The 'information system' is the same thing throughout the study; all that changes is how students understand it. Berglund (2005) studied students' understandings of various network protocols. The protocols remain fixed, but different students have different conceptions of what they are and how they are used.

By contrast, the findings presented here have elicited variation in *approaches* to conducting lab practical classes. It is not that different academics have different understandings of the lab practical: as the people who create the classes for their course, they can be assumed to have a fairly complete understanding of those classes. Rather, different academics have different uses for the lab practicals, and thus run them in different ways with different sorts of goal.

The phenomenographic method was used to analyse interview transcripts gathered for a phenomenographic study, but did not result in a categorisation of different understandings of the static concept of a lab practical. What has emerged instead is different approaches to lab practical classes, each suited to different purposes. In this sense, what has been achieved is not pure phenomenography. Fortunately, we do not believe that this makes it any less valuable.

4.4. *The Value of this Work*

How can the computing education community benefit from this work? McKenzie (2002) showed that when university teachers were able to discern critical aspects of variation within differing teaching strategies, they moved to more student-focused ways of experiencing their teaching. Therefore an awareness of the different categories of practical class will better inform academics who are designing courses; they will be able to consider the categories and decide just where they intend their own work to lie.

This leads to another aspect of the difference between these findings and the standard expectations of phenomenography. Aligned with the hierarchical arrangement of phenomenography's categories of description is an understanding that the more inclusive categories are in some sense better, that they are an ideal to be aimed for. Cope (2000) would presumably be happy if all of his students expressed the most inclusive understanding of information systems, and Berglund (2005) would likewise rejoice if his students all expressed the most inclusive understanding of network protocols. If this were the case with computing practicals, all teachers should be aiming to design their practicals in accord with the fourth category presented here. To the contrary, it is important to recognise that the different categories represent different approaches used for different purposes, and to appreciate the value of this distinction. Faced with a hierarchical categorisation of approaches to lab practicals, it is the responsibility of academics to decide which category or combination of categories is most appropriate for their courses.

The notion of combining categories is finely illustrated by E3, who in a single 12-week course progresses deliberately in approach from category 2 to category 4:

"the tutorials ... the first, about four, are actually structured formal tutorials: do this, do this, monkey see, monkey do ... you know, do this, open this up, use this tool, right-click this, type this in the box, in the wizard, enter this data ... very, very specific instructions. The next four are less specific, and are mainly concerned with integrating the concepts of the lectures into their assignment. And the final four are basically one-liners: 'Integrate the material in the lectures into your assignment, full stop."' [E3].

Although it is the job of phenomenography to describe rather than to recommend, in response to comments from colleagues we suggest that some academics might perceive the lower levels of our categorisation as being more suited to beginning students and the upper levels as better suiting advanced students. We do not have enough pertinent transcripts to determine whether they support this suggestion, so at this point it must remain completely hypothetical.

4.5. *Future Directions*

Future work could include an investigation of academics' differing perceptions of students that lead them to adopt the different approaches delineated by this study. Further exploration of the current transcripts might provide a first step in this direction; but it is possible that these transcripts are not sufficiently rich with regard to this particular question, and that to answer it properly will require a fresh study with questions designed for the purpose.

Further study, of both academics and students, might also result in firmer or clearer guidelines as to which approach to lab practicals is better suited to which circumstances.

5. Acknowledgments

This study was supported by a Special Projects Grant from the ACM Special Interest Group in Computer Science Education (SIGCSE). The authors thank Raymond Lister, Anders Berglund, Ilona Box, Chris Cope, and Arnold Pears for thinking of and running the PhICER workshop; and fellow PhICER participants Chris Avram, Mat Bower, Angela Carbone, Bill Davey, Bernard Doyle, Sue Fitzgerald, Linda Mannila, Cat Kutay, Mia Peltomäki, Judy Sheard, Des Traynor, and Jodi Tutty for their transcripts and collaboration.

References

Åkerlind, G.S. (2004). A new dimension to understanding university teaching. *Teaching in Higher Education*, **9**(3), 363–375.

Åkerlind, G.S. (2005). Phenomenographic methods: a case illustration. In J. Bowden and P. Green (Eds.), *Doing Ddevelopmental Phenomonography*. Melbourne, Victoria, Australia, RMIT University Press. pp. 103–127.

Azemi, A. (1995). *Teaching Computer Programming Courses in a Computer Laboratory Environment*. `http://fie.engrng.pitt.edu/fie95/2a5/2a55/2a55.htm` (retrieved 3 March 2006).

Ballantyne, R., J.D. Bain and J. Packer (1999). Researching university teaching in Australia: themes and issues in academics' reflections. *Studies of Higher Education*, **24**(2), 237–257.

Berglund, A. (2005). *Learning Computer Systems in a Distributed Project Course: the what, why, how and where*. Uppsala, Sweden, Acta Universitatis Upsaliensis.

Bligh, D.A. (1971). *What's the Use of Lectures?* Penguin, Harmondsworth.

Carbone, A., and J.J. Kaasbøll (1998). A survey of methods used to evaluate computer science teaching. In *Proceedings of the 3rd Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE '98)*. Dublin, Ireland.

Carter, J., and R. Boyle (2002). Teaching delivery issues – lessons from computer science. *Journal of Information Technology Education*, **1**(2), 77–89.

Cope, C. (2000). Educationally critical aspects of a deep understanding of the concept of an information system. In *Proceedings of The Fourth Australasian Computing Education Conference (ACE2000)*, Melbourne, Australia.

Davy, J., and T. Jenkins (1999). Research-led innovation in teaching and learning. In *Proceedings of the 4th Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE '99)*. Krakow, Poland.

Fox, D. (1983). Personal theories of teaching. *Studies in Higher Education*, **8**(2), 151–163.

Hagan, D., and J. Sheard (1998). The value of discussion classes for teaching introductory programming. In *Proceedings of the 3rd Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE '98)*. Dublin, Ireland.

Hughes, I.E. (2000). Alternatives to laboratory practicals – do they meet the needs? *Innovations in Education and Teaching International*, **38**(1), 3–7.

Jenkins, T. (2002). On the difficulty of learning to program. In *Proceedings of 3rd LTSN-ICS Conference*. Loughborough, USA.

Kember, D. (1997). A reconceptualisation of the research into university academics' conceptions of teaching. *Learning and Instruction*, **7**(3), 255–275.

Kutay, C., and R. Lister (2006). Up close and pedagogical: computing academics talk about teaching. *Australian Computer Science Communications*, **28**(5), 125–134.

Lister, R., A. Berglund, I. Box, C. Cope, A. Pears, C. Avram, M. Bower, A. Carbone, B. Davey, M. d. Raadt, B. Doyle, S. Fitzgerald, L. Mannila, C. Kutay, M. Peltomäki, J. Sheard, Simon, K. Sutton, D. Traynor, J. Tutty and A. Venables (2007). Differing ways that computing academics understand teaching. *Australian Computer Science Communications*, **29**(5), 97–106.

Lister, R., and J. Leaney (2003). First year programming: let all the flowers Bloom. *Australian Computer Science Communications*, **25**(5), 221–230.

Marton, F. (1986). Phenomenography – a research approach to investigating different understandings of reality. *Journal of Thought*, **21**, 28–49.

McInnis, C. (1999). *Lecturing*. Centre for the Study of Higher Education Booklet Series, University of Melbourne.

McKenzie, J. (2002). Variation and relevance structures for university teachers' learning: bringing about change in ways of experiencing teaching. *Research and Development in Higher Education*, *25*, 434–441.

Prosser, M., K. Trigwell and P. Taylor (1994). A phenomenographic study of academics' conceptions of science learning and teaching. *Learning and Instruction*, **4**, 217–231.

Reif, F., and L. A. Scott (1999). Teaching scientific thinking skills: students and computers coaching each other. *American Journal of Physics*, **67**(9), 819–831.

Ruthven, K., and S. Hennessy (2002). A practitioner model of the use of computer-based tools and resources to support mathematics teaching and learning. *Educational Studies in Mathematics*, **49**, 47–88.

Sewell, R.D.E., R.G. Stevens and D.J.A. Lewis (1996). Pharmacology experimental benefits from the use of computer-assisted learning. *American Journal of Pharmaceutical Education*, **60**, 303–307.

Simon (2003). An IT degree combining the strengths of university and TAFE. *Australian Computer Science Communications*, **25**(5), 35–40.

Venables, A., G. Tan, S. Devi Nagappan and A. Ghous (2006). Everything we wanted to know about our course, but were afraid to ask? Views from a students' perspective. In *Proceedings of 2006 Information Resources Management Association International Conference*. Washington. D.C. USA.

Ward, J.P.T., J. Gordon, M.J. Field and H.P. Lehmann (2001). Communication and information technology in medical education. *Medical Education Quartet: The Lancet*, **357**, 792–796.

Simon has a bachelor of science and a bachelor of arts from James Cook University of North Queensland, and a diploma of computer science and a master of mathematics from the University of Newcastle, Australia. He is currently a senior lecturer in information technology at the University of Newcastle, Australia. He has also taught at James Cook University of North Queensland, Exeter University in England, and Griffith University in Australia. In a career spanning more than 30 years he has taught many aspects of computing. Most of his recent research is on aspects of computing education.

M. de Raadt is a lecturer at the University of Southern Queensland, Australia. Michael's research focuses on teaching problem solving skills to novice programmers. Michael is also interested in online teaching and assessment including electronic peer review.

A. Venables is a lecturer in computer science at Victoria University, Melbourne, Australia. She has research and teaching interests in the application of artificial intelligence in biological systems. As a former secondary science and mathematics teacher who has migrated into tertiary education, Anne is particularly interested in innovations in education and she has previously published in this field.

Prieigų prie akademinių kompiuterinių praktinių laboratorijų įvairovė

SIMON, Michael de RAADT, Anne VENABLES

Šio straipsnio autoriai, būdami plataus profilio fenomenaliais kompiuterijos mokytojais, pastebėjo besiskiriantį jų supratimą apie praktinius kompiuterijos darbus ir nustatė keturias skirtingas kompiuterijos praktinių darbų kategorijas. Straipsnyje svarstoma, kuri iš kategorijų gali būti lyginama su kitų dalykų ne paskaitos tipo užsiėmimais, pasižyminčiais panašiomis kaip ir kompiuterija savybėmis. Šio tipo praktinių užsiėmimų supratimas padės dėstytojams geriau atsirinkti, kas kuriant kursus geriausiai išreiškia jų tikslus.