INTRODUCTION

The thoughts expressed in this paper arise from the constructivist assertion that learners have good reason to do what they do, and that there is merit in seeking to understand the knowledge on which they draw when going about their work or talking about what they have done. It will helpful, however, to commence by presenting some anecdotes that describe how this applies to the use of computers.

1. A pre-school child knows that to open the car door, the remote control may be used. One morning she gets out of the car, and proceeds to use the same remote control to try to open the front door of the house. The child has learnt something about locks and remote controls, but has still more to learn.

2. At an introductory computer course, a teacher was heard to say that she ‘knew a bit about computers’. A few minutes later, she complained to the instructor that the pedal isn’t working. Puzzled, the instructor looked at her workstation to observe that the mouse had been taken off the table and placed on the floor, as if it were a pedal. The teacher was a sewer, used to a sewing machine and its pedal, and had drawn on this knowledge to make sense of the environment in which she now was working.

3. Another adult beginner was shown the process of creating, saving and printing a document. Each lesson he focused on placing his disk in the drive correctly, and then proceeded to create and print his work, only to come in the following lesson to re-type and re-print his work. After about four lessons of this, he realized what his disk was for, and how wonderful it was to be able to save his work, and how this would improve his efficiency each lesson. He had not, at first, grasped the accepted and definitive model of computer architecture as consisting of primary and secondary memory and the purposes of each.
4. On a popular morning television show, a segment was presented on the topic “the internet’s getting full” (Seven Sunrise, 2009). What it meant to have a “full internet” was something that the show’s journalist found hard to articulate, but what emerges from this is that some people, at least, have a conception of the internet as something fillable. Maybe it can be conceptualized as being like a network of pipes, maybe like an extremely large bucket. Ironically, in such cases where the availability of water is considered, “being full” is a good thing. Even to have full library shelves is not a bad thing unless you want to store more books. Maybe the intention was to draw attention an analogy between the degradation of performance which is sometimes experienced when a computers user has a full (or nearing full) hard disk. Regardless, it seems that some users have some understanding of the internet (or hard disks) to which the notion of “fullness” might (or might not) be apt.

These anecdotes point to the constructivist position (So, 2002) that all people have some kind of knowledge which guides how they work with them and how they perceive the limitations and possibilities. My particular interest is with the ‘common-place’ computing activities which may be found in classrooms or homes around Australia (if not the world), for example: word processing, e-mail, graphics, file management, virus checking, accessing pages on the Web, facebook, youtube, google or engaging in e-commerce. In this paper, I develop an argument for recognizing and understanding the underlying and often implicit knowledge which guides how people work with such technology. For simplicity, I will refer to this as learners conceptions of ‘common place’ computing.

The discussion is presented in two sections. The first section is centred around a literature review, preceded by a discussion of learner understandings in more general terms, and followed by a discussion of knowledge in relation to computing. The second section is more speculative, identifying potential conceptions and data gathering techniques that may be valuable in future research. The purpose of this paper, then, is to be thought-provoking – to identify a deficiency in our thinking and research about computing to date and then to encourage the development of work in that field.

PART A: BACKGROUND, REVIEW AND THEORY

Constructivism

It is important to commence the detailed discussion of ‘learners conceptions’ by clearly articulating the epistemology in which this work is situated. As already flagged, the author espouses a constructivist view of how knowledge is acquired and integrated with other knowledge. Constructivism has been described, reworked and summarized countless times throughout the educational literature, a venture that will not be re-attempted here. I have found useful perspectives in Ben-Ari (2001), Matthews (1994) and Skamp (2008), and adopt a generalist stance in relation to it, that:

- we come to know our world by interacting with it (Brookes, 1987, p. 64); and
- people construct ideas as they learn, and they use prior knowledge, experiences, and beliefs, as well as interpretations they generate in the moment, as the stuff out of which to build those ideas (Knowles & Holt-Reynolds, 1994, p. 6 [quoted by Carter & Doyle 1996, p. 122])

From this, it can be said that knowledge is robust (insofar as it proves to be viable), idiosyncratic, sensitive to the particular holder, incomplete, familiar and sufficiently pragmatic to have taken the learner to where he or she is today, and to acknowledge that knowledge is not constrained to the learning of propositions and rules at identifiable moments in time, but includes the rather more ad-hoc accumulation of experiences, beliefs and interpretations.

Constructivism has been enormously important in computer education, particularly through the work of Seymour Papert (eg Papert, 1980) and its derivatives. In such work, use of computers in schools was not valued for its own sake, but as a means of developing knowledge in another domain, such as mathematics. The computer thus became a somewhat invisible partner in the learning enterprise, which was based in constructivism.
This paper seeks to make the knowledge of the computer as the object of study. That is not to say that use of computers in schools should be valued for their own sake rather than as tools to promote learning. Rather, it is to acknowledge that students in our schools (and the community more generally) interacts more-or-less successfully with computers on a regular basis, and it is therefore important to garner some understanding of what they ‘know’ of computers and related technology. That is, to grasp the learner’s conceptions of the technology which have developed through interacting with the technology (amongst other things) through psychological processes which I assert are broadly constructivist.

**Learner understandings**

Increasing attention has been paid to learner conceptions in recent decades. In the subject area of science, for instance, Osborne and Freyberg (1985, p. 12) drew attention to “children’s science” by observing that:

- from a young age, and prior to any teaching and learning of formal science, children develop meanings for many words used in science teaching and views of the world which relate to ideas taught in science;
- children’s ideas are usually strongly held, even if not well known to teachers, and are often significantly different to the views of scientists; and
- these ideas are sensible and coherent views from the children’s point of view, and they often remain uninfluenced or can be influenced in unanticipated ways by science teaching.

The identification of realities of learning, such as these has led to nothing short of a paradigm shift in science curriculum over the last 20 years or so (So, 2002), attributable to science educators taking constructivist ideas seriously. In earlier times, science teaching methods may have emphasized the learning of answers more than the exploration of questions, memory at the expense of critical thoughts, bits and pieces of information instead of understanding in context, recitation over argument, reading in lieu of doing. If there was a time when science teaching was dominated by the transmission of disconnected facts (though arguably this was never an appropriate goal or even universal approach), there was also a time when students came to computing classes with essentially no exposure or life experience with computers whatsoever. Well-meaning teachers (such as the author) would try to make connections between life experiences and teaching computer use. For instance, to introduce concepts such as ‘desktop’, ‘copy’, ‘paste’ and ‘cut’ (which hard for the novice user to imagine with pre-GUI word processing software such as Wordstar and Zardax), one could clear a desk space, place a newspaper on it and created a toolbox of scissors and glue and proceeded to act out what the students were doing with the software.

Countless numbers of investigations and publications in science education have addressed the idea of strongly held ideas which are hard to influence and often not apparent to teachers (eg Duit, 2009, whose bibliographic work numbers in excess of 8400 entries), and texts on how, from this perspective, science teaching should be different today to what it was typically a few decades ago abound (eg Skamp, 2008). In the field of computing, by contrast, students of the present era are frequently described as “ICT savvy” (Christopherson, 2006) or “digital natives” (Prensky, 2007). Despite the observation by Yan and Fischer (2004) that insufficient attention has been given to how people learn to use computers from the perspective of cognitive development, some of our practices have continued to be about the learning of a range of a catalogue of skills (see both the popular ICDL and INGOT programs), or have focussed on the purpose, design and context of computer-produced artifacts (eg Victorian Curriculum and Assessment Authority, 2008). Even large scale projects, such as the netbook trial in Victorian government schools, seem quite content that students can (and should) ‘learn all about the technology themselves’ (Ellul & McGarry, 2009).

---

1 See http://en.wikipedia.org/wiki/WordStar
3 The International Computer Driving Licence, http://www.icdl.com
4 An abbreviation of “International Grades - Open Technologies”, http://www.theingots.org
Some (e.g., Galloway, 1999; Urban-Lurain, 2003) have lamented an attention to skill rather than developing a deeper understanding, and it seems generally true that the underlying ideas learners may have, how close they are to authoritative explanations, or what inappropriate ways of working, thinking or talking that may ensue do not seem to be on the radar of the typical teacher or researcher. Even if learners are “ICT savvy” or if being “digital natives” has seen them develop fundamental understandings *au naturale* in ways not typical of earlier generations, this has not been investigated or documented in any deep, conceptual sense akin to the work in science education (see literature review below). Therefore, there is nothing systematic to distinguish between learners who are “conceptually good” and those who have become adept at the computing equivalent of “barking at text”, and little acknowledgement – in stark contrast to science – that a sound conceptual understanding (whatever that may be) would actually be a preferable goal for programs which induct learners (of any age) to use computers and related technology.

An example from Ben-Ari (2001) helps reinforce this point. He asks the reader to consider the paste operation: the word whose original meaning is “form a permanent chemical bond between one item and another” must be related to the operation “insert a copy of the material held in an internal buffer into the current working document at the place pointed to by the cursor” (p. 59).

Ben-Ari writes from the point of view of a computer science specialist, which would inform us that there is exactly one authoritative model for copying and pasting, and this is what should be conveyed to learners. I would contend that there are people who regularly use “copy and paste” who have never heard of the term ‘internal buffer’, and it might be instructive for the researcher or teacher to gain some real insight into understandings they do have of what is going on.

This purpose of this paper, then, is to contend that “science” may reasonably be replaced with “computing” in the above-mentioned findings from Osborne and Freyberg, and that important implications for how the computing may be taught may then follow. The first statement, that meanings for words and views of the world develop prior to any formal knowledge of a discipline, can certainly be applied to the anecdotes shared earlier. If it is also true that ideas are strongly held, often contrary to expert opinion and uninfluenced by teaching, then there is much to be learnt by those of us who induct learners (of any age) to use of computers and related technology. The corollary is that quite inadequate understandings of the technology may continue to guide use well beyond the rudimentary stages. Even though some researchers in computing education have recognized that students do not come to computing classes *tabula rasa*, there is at least a hint that instilling ‘correct’ conceptions is rather easy to do, and can be achieved through teacher-directed instruction rather than through structured student experience (e.g., Ben-Ari, 2002; Yeshno & Ben-Ari, 2001). Science educators have increasingly adopted the view that knowledge cannot be transferred intact from the head of the teacher to the head of the student (Louden & Wallace, 2002, p. 191).

Another way of saying that Osborne and Freyberg’s observations apply to computing is to propose that there is an analogue in the area of ‘common place’ computing to what is called “alternative conceptions” in science education (Skamp, 2008, p. 4). It was once more common to refer to this as “children’s science” (but it is a phenomenon true not only of children), and other terms such as ‘misconceptions’, ‘everyday conceptions’, ‘intuitive conceptions’ and ‘naïve ideas’ also used. I have settled on “learners conceptions” as the title of this paper for three reasons: it relates to the preferred term in science education; it generalizes to ‘learners’ rather than referencing a particular age-group; and it side-steps the presumption that the conceptions will necessarily be contrary to an authoritative position. In time, and following investigation, it may become appropriate to talk of “alternative conceptions” of ‘common place’ computing or even of “children’s computing”, but that would be presumptive for the present.

It is useful to observe the difference articulated by Skamp (2008) between conceptions and knowledge structures.

[Learners] … may have coherent … ways of looking at the world which could account for alternative conceptions … These ways of looking at the world, it can
by hypothesized, are underpinned by the manner in which various ideas are linked in each learner’s mind (called cognitive structures, framework(s), etc) (pp. 4-5). In focusing on “learners conceptions”, one is more concerned with “what” ideas learners have which guide their practice than deriving an underlying explanation. For example, Hubber and Tytler (2005), who list dozens of alternative conceptions encompassing 14 topics in junior levels of science, observe in relation to bodies and skeletons that young children:

- imagine their bodies as hollow skin bags that are all ‘stomach’ (a reservoir in which blood, food and wastes are somehow contained);
- think of food as vanishing after it is eaten;
- don’t associate the ear with hearing;
- don’t think of the brain being needed for overt behaviour; and
- think that we see because we look; it has nothing to do with light.

It is conceptions of this type, but related to ‘common place’ computing which I am suggesting would be interesting to investigate: What are they? How close they are to authoritative explanations? How easily influenced are they? To what extent are they indeed influenced by (existing) teaching practices.

Having now introduced the field of study, I now more to a more thorough review of the literature relating to it. This is followed by a consideration of some aspects of knowledge relating to computing, prior to a more speculative discussion relating to some techniques and considerations which might further study in this field.

**Literature Review**

Whilst the particular interest in this paper is with ‘common-place’ computing activities which may be typically found in classrooms or homes, it is helpful to review the literature somewhat more generally. Indeed, there has been some attention to understandings of technologies related to ‘common-place’ computing such as Video Cassette Recorders (VCRs), materials technology, autonomous robots, internet searching and mobile devices such cellular phones. The following review is based on searches developed from key words such as ‘understanding’ and ‘computing’ or ‘technology’.

Over 15 years ago, Krendl, Clark, Dawson, and Troiano (1993) considered preschoolers and their use and conceptions of Video Cassette Recorders (VCRs). They found that

Children do not think of television as a medium but as a delivery system. It delivers movies on tape, television programs on tape and other programs not on tape. The distinction in children’s minds between tape viewing and broadcast/cable viewing is not a critical one – television to them is a delivery system only (p. 309).

Arguably, this kind of understanding might apply to computer users – the difference between data resident on a hard drive, DVD or the internet might not be considered to be consequential.

In another early work, Lloyd (1996) made a study which is interesting both in terms of methodology and findings. She studied the mental models students have of computers as expressed through both literal and figurative speech, through responses to sentence stems such as “a computer is like ...” or “programming is like ...”. The outcome of the study was that “it allowed me to ‘see’ the computer through my students’ eyes - and most alarmingly, to realise that what we were all seeing were very different things” (p. 23). The detailed findings are potentially not relevant over a decade later, but a replication study would certainly be interesting. The work of Williams and McKeown (1996) would also be interesting to repeat. They studied teacher’s views of the Internet from a social constructivist perspective, investigating their understanding of the Internet (in particular, its value applicability to their life and work) was shaped through the communities with which they interact (both online and face-to-face).
In other related fields, but more recently, Davis, Ginns and McRobbie (2002) have described students’ understandings of concepts associated with materials technology. Young student's perspectives in explaining the ‘behaviour’ of a mechanical, autonomous robots have been studied by van Duuren and Scaife (1996) and Levy and Mioduser (2008). Zammit (2000) considered the iconic literacy of computer users and found that the meaning of icons was not always immediately apparent, a sentiment echoed by Ben-Ari (2001, p. 56). Kafai (2008) explored students’ conceptions of a computer virus, and found that that the large majority of online users have little understanding of a computer virus, and that explanations concentrated on ‘behaviour’ features of virus, not on their underlying structure. It was also found that visual representations might work well for exploring many concepts in science learning, they did not work well in this context. Efthimiadis, Hendry, Savage-Knephshield, Tenopir and Wang (2004) also report on work which asked to draw: in order to identify how people conceptualise internet search engines and how they work, university students were asked to draw sketches of how they work.

Closer to the idea of ‘common place computing’, it needs to be observed that constructivism is steadily being embraced by computer science educators (that is, there is a growing realization that their students are not empty vessels waiting to be filled). Ben-Ari (2001) and Nordström (2002) have discussed the importance of this perspective in computer science. More specifically, Chesñevar, Maguitman, Gonzáles and Cobo (2004) have employed such ideas in the teaching of highly abstract topics associated with the theoretical fundamentals of computing. Chen (2003) has used constructivist principles teaching computer networking including some innovative teaching procedures using items such as rope, key rings and post-it notes to represent elements of a computer network and allow students to put their understanding on display. Conceptual understandings have had some attention, too. There is a suite of papers published in the field of the psychology of computer science which address school student understandings of programming and computer science concepts (eg. Ben-Ari & Yeshno, 2006; Pea, 1986; Powers & Powers, 2000).

Notwithstanding this background, there is little work done with ‘common place computing’. Ben-Ari (1999), Yeshno and Ben-Ari (2001) and Ben-Ari (2001) studied the mental models of word processing of academic staff in a university, and whilst Ben-Ari and Yeshno (2006) extended this work to school students, the data available at http://portal.acm.org/ illustrates that Ben-Ari’s work has been predominantly cited in research into programming and computer systems, not at school-level computing. Titled Constructivism in computer science education, taking word processing as an example, and published in a refereed journal, the paper by Ben-Ari (2001) is as close as any to being a seminal paper in the field.

Influenced by Ben-Ari’s work, Powers and Powers (2000) conducted some research into student preconceptions related to Computer Science and Information Systems (CSIS). They observe that little work has focused on identifying the initial ideas that students bring with them to the door of their first computing class. Some related work on student conceptions has been done, but it differs from ours in the following important ways. First, these works primarily consider the conceptions that students construct once in the CSIS classroom, not the conceptions that they bring with them to the door. Also, most of this work is limited in scope to programming per se, as opposed to CSIS generally (p. 1).

And furthermore:

considerable research has focused on the erroneous ideas that beginners develop in the process of learning to program. But to our knowledge, the idea of considering the intellectual framework that existed before the learner engaged the subject has not been explored. To what extent might their erroneous ideas be the result of general knowledge, formed in the general social setting, which is either inaccurate or misapplied? CSIS educators must confront the erroneous preconceptions that students bring to the discipline from general society, and these preconceptions cannot be confronted until they are identified (p. 4)
In other words, important as the work is, we should not expect to find too much written; indeed, very little has apparently been published in the last 9 years. Powers and Powers findings are discussed later (as they provide as good a yardstick as any of what sort of ‘conceptual understandings’ might be found), but unfortunately their method for gathering data is not discussed, and may in fact be a reflection on what a teacher has casually observed after much time in the classroom.

In describing the “Fluency with Information Technology” (FIT) initiative, Urban-Lurain (2003) explained that some argue that it is not going to be necessary to teach FITness because, as computers become ubiquitous, students will arrive at college already FIT. This argument goes back to the 1980s … However, we have found that students are not arriving at our course any more FIT than they were in the 1980s. Students have a great deal of exposure to using computers, but have little conceptual understanding (p. 69). In response, the FIT initiative designed instruction around an analysis of critical computing concepts, their inter-relationship and the types of problems that epitomise these concepts at each stage of the learning process. Unfortunately, the bank of resources developed through this program seems to be no longer available, and the concepts addressed seem more to be those derived from computer science than those emergent though investigation.

The one field which impacts on the concerns expressed in this paper which is quite extensive is that concerning mental and conceptual models. A conceptual model is one which is invented to provide an appropriate representation of a target system; the mental model is what the user ‘presently has in his/her head’ about the target system (Cardinale, 1991). From this emerges several strands, but a consistent theme is about ‘engineering’ the optimal performance of an individual. Whilst this approach recognises that users are not empty vessels waiting to be filled, it tends to emphasise effective ‘transmission’ of the conceptual or target model, and not be particularly concerned with the mental model (eg. Hagmann, Mayer and Nemlinger, 1998; Yeshno and Ben-Ari, 2001). Nor does it pay much heed to the possibility that the mental model might be strongly held, sensible and coherent views from the learner’s point of view, or difficult to influence (or influenced in unanticipated ways).

Notwithstanding these concerns, there are some papers of particular interest. Scott Brandt has discussed the mental models of learners as relevant to the role of the reference librarian (Brandt, 1997; Brandt, 2001), including a study entitled “Insight into mental models of novice Internet searchers” (Brandt & Uden, 2003). There have been numerous studies into student’s hypermedia and web navigation (eg. Fenley, 1998, Chiu and Wang, 2000). Ziefle and Bay (2004) have considered the mental models of a cellular phone menu. In the work by Brandt and Uden (2003), the mental/conceptual model literature adds another research method to the repertoire – that of cognitive task analysis.

In summary of this review, several things can be said. Firstly, there is no major corpus of work in the field of ‘common place’ computing, and whilst there is some in related fields, there is little published in academic journals. Secondly, a range of research methods have been identified, and these include: analysis of literal and figurative speech, learner articulation of their basis for working as they do (eg think aloud protocols), cognitive task analysis, and drawing and explanation (although a warning is sounded from Kafai [2008] that asking learners to create a drawing of ideas which are abstract, such as computer viruses, may not be productive).

Thirdly, some small hint of the nature of learner’s knowledge about computing is given in a few papers. Ben-Ari (1999, 2002) and Yeshno and Ben-Ari (2001) have found that, in the absence of an explicitly-taught conceptual framework, students were unable to articulate their bases for working as they do. If knowledge about computing is incoherent, then that would be in marked contrast to the “sensible and coherent views” which learners are said to present of science as a result of their life and experience. At least, this possibility deserves some attention.
What is clear is that many computing educators recognize that learners come to computing (or related) activities with understandings, and that it is important to value these. Several have had reason to question whether students of the present day are any more knowledgeable of computing matters, in any deep sense, than students of earlier times. What emerges, then is an opportunity to further develop data gathering techniques and to systematically collate views of learners understanding of computers. Some potential approaches are developed in Part B.

Knowledge in computing

Before moving to that discussion, it is important to acknowledge an ontological difference between knowledge in computing compared with that in science. Theories in science – whether they be those of Newton, Einstein, the alchemists, or a student – are part of a quest towards ever more productive explanations for how the world works; there are no absolutes in terms of explanation. As Louden and Wallace (2002, p. 191) observe, cognitive scientists would view student knowledge as imperfect and in need of perfection; philosophers would view science itself as imperfect and imperfectable. In contrast, computing – at least at the level of software or macroscopic hardware function – offers absolutes: what a task bar is, how saving a file should work, for instance, are precisely defined. As Ben-Ari (2001, p. 49) explains:

... since computer science deals with artifacts — programming languages and software — the creator of the artifact employed a very detailed model and the learner must construct a similar, though not necessarily identical, model. Second, knowledge is not open to social negotiation. Given that the word processor is an extant artifact, you cannot argue that its method of using fonts is incorrect, discriminatory, demeaning, or whatever. You may be able to choose another software package, or to request modifications in an existing one, but meanwhile you must learn the existing reality.

Matthews (1994, p. 133) observes (in relation to science) that it is with respect to contemporary science that students might have misconceptions, it is not with respect to the behaviour of the natural world. The situation with computing is slightly different, as any misconception will be with respect to an authoritative and absolute (rather than contemporary) model. By way of example, one can live in a society which has constructed a view about the earth’s shape (cf digital native) which is productive in its explanations of the phenomena around you most of the time (cf ICT savvy) which is actually counter to an expert opinion. Just because your ideas are productive for you, your ideas may still be at considerable variance to an expert position. The discipline of computing, therefore, is concerned with of knowledge which is perfectable. (It is little wonder, then, that the ‘mental model approach’ has appeal, as it is concerned with the transmission of the mental model developed by the designer of the software ‘into’ the head of the learner).

Ben-Ari (2001, p. 56) asserts that there are two characteristics of constructivist views of knowledge in computing that do not appear in natural sciences: that the beginning computer science student has no effective model of a computer, and that the computer forms an accessible ontological reality. The ontological reality is what has been discussed in the previous paragraphs; the accessibility of that refers to the fact how the system responds to particular input is readily available to the user and thus provides near-instantaneous feedback which helps the user formulate a mental model. These observations about the ontological reality seem just as true of ‘common place’ computing as it does with computer science, as was Ben-Ari’s concern.

The assertion that the student has no effective model of a computer may be true for the beginning computer science student, but seems most unlikely to be true for ‘common place’ computing. Indeed, this paper asserts that since most computer users are actually able to go about the business of using a computer they must have constructed knowledge or model of some sort; or, to amplify the meaning of ‘effective’ as given by Ben-Ari, the learner has a cognitive structure that the student can use to make viable constructions of knowledge based on their experiences. In this way, I would assert the validity
of drawing on the science education literature to inform a discussion of learners conceptions in the domain of ‘common place’ computing.

**Framing research into learner’s conceptions**

There are two ideas which are important to identify as they may well have bearing on how research into learner’s conceptions might proceed, indeed they may well help to frame such research.

Firstly, there is the distinction between ‘data structure’ compared with ‘operations on that data structure’ (Ben-Ari, 2001, pp. 47-48). This is a fundamental distinction in computer science, but unlike other ‘essentials’ of computer science (such as a von Neumann architecture and the universal Turing machine) which are largely invisible to anyone but the engineer or low-level programmer, this distinction impacts on day to day computer use. For instance, letters and their formatting in a word processing document comprise the data, and actions such as ‘copy’ or ‘paste’ are operations. Ben-Ari suggests that in order to learn how to use a word processor (for instance) one must create a mental model of the data structure and the effect of each operation. The issue, then, is whether users are aware of the distinction or not, whether such awareness actually matters in terms of how they conceptualise (and then work with) computers, and whether a more fine-grained awareness of different data structures in the environment in which they work (cf paragraphs and tables in a word processor) is desirable.

The second idea is the distinction between ‘ostensive’ and ‘non-ostensive’ objects (Artigue, 2002, pp. 249, 270). Mathematical objects, for instance, are non-ostensive because they are not directly accessible to our senses. What the earlier quote from Ben-Ari (2001) about ‘pasting’ indicates is that computing is fundamentally about non-ostensive objects (eg internal buffer). Artigue observes that we work with non-ostensive objects through ostensive representations. The familiar user interface, with its mouse, menus and icons, is an ostensive reality and Artigue goes on to assert that our work with an ostensive representation shapes our understanding of the non-ostensive objects. This ostensive/non-ostensive distinction is vital as it clarifies that what is of real interest is learner’s understanding of a non-ostensive ‘world’.

**PART B: SUGGESTIONS FOR ADVANCING THE RESEARCH**

Unlike the previous section which was centred around the literature review, this section is more speculative, identifying potential conceptions and data gathering techniques that may be valuable in future research. Firstly, there can be discerned from the literature numerous conceptions of computing which might guide our *modus operandi* with respect to computers and other technology. I refer to these as ‘high level conceptions. An exploration of learner’s conceptions may well be enhanced by attending to these, but they are not particular ideas which guide practice of the sort such as “food vanishes after it is eaten” or “the ear not being associated with hearing”. Following a presentation of the high-level conceptions, some techniques which may be useful in probing for particular ideas and conceptions are presented.

**High-level conceptions**

There are several sources that discuss a range of ‘views of computing’ which allows us to obtain a foothold onto some of the high-level conceptions which learners may have. Lawler (1999) asserts that there is language which is typical of each view, and each has its problems, each its opportunities. For instance, the idea of the computer as servant is reinforced by “software that uses first person pronouns, by interfaces that make the user learn a recondite and unchangeable set of terms for what the software can do, and by overly cute documentation that personalizes the name of the program”. Some high level conceptions of computing are as follows:
The computers as an analogue devices (Powers & Powers, 2000)
Students are accustomed to devices that respond linearly with variances in their input. This
is not always the case with computing: a single-click will not suffice for a double-click;
changing a single line of code, can drastically alter the output. But note that tools such as
volume controls and gesture-based interfaces, such as in vlc⁵, work by responding to
incremental changes in mouse function, so the computer user can sometimes work
analogically, and other times digitally.

Complexity and levels of abstraction (Powers & Powers, 2000)
Few students ever truly grasp the degree of complexity of computing systems and the
absolute necessity for levels of abstraction throughout computing. Indeed, students often
equate the complexity of computers with other much simpler digital devices, such as land-
line telephones and cassette players.

The computer as a giant brain (Powers & Powers, 2000)
From the screen images which concurrently present information from multiple
applications, one might be inclined to think of the computer as a “giant brain”, already
“aware” of all the information available. The reality that this information is distinct and not
necessarily shared within the computer is not apparent to students. This could manifest
itself in anything from puzzlement over one application failing to print even though
another prints perfectly, through to an attitude of ‘I’ll just get on the internet and look up
the family tree’.

The computer as an appliance (D’Ignazio, 1995)
A computer is not like an appliance such as a toaster or an electric screwdriver. Those are
somewhat static technologies which exist for a particular purpose. In contrast, computers
are in constant mutation, have instant obsolescence and are linked to every human
endeavor. People of the 1940s or 1950s may have dreamt of technologies which would
make our lives simper and easier but that is not the week-in-week-out experience of
computer users when networks fail, backups get lost or software malfunctions.

The computer as a solution (D’Ignazio, 1995)
A realistic view of computers is that they create as many problems as they solve. As
D’Ignazio observes, as long as we look at technology as a solution, all we will ever be is
cranky, and that we should face the reality head-on that technology is actually a gigantic
problem. Our modus operandi with respect to the computer being a solution (or not) is
very important.

The computer as a servant (Lawler, 1999)⁶
Everybody would love a program that was a good servant; but like even the best of
servants, such a program must be instructed on what to do and how to do it. And if you’re
still not satisfied with how the program or the servant does it, too bad. There is only so
much any servant (or any program) can be expected to know or to learn. Computer
programs are much more limited than humans, and can typically only do one kind of thing.
To do another, you need a different program (or specialized servant), and that one can’t
speak the same language as your other one(s), and can’t learn anything they don’t already
know.

The computer as a race (Lawler, 1999)
The term “run” is used in relation to making computer programs “go”. It sounds like we’re
being a speed freak, trying to make everything go faster. As computer users, we’ve
become addicted to instantaneity. (One might wonder about the connotations of the
computer-synonym of ’run” – execute!)

⁵ See http://wiki.videolan.org/Mouse_Gestures
⁶ Lawler (1999) draws on work by Lakoff and Johnson (1980) who asserted that our though is shaped by the
metaphors which surround us, and furthermore, the metaphors found in the language that we use give insight
into our understanding. Lawler presents a range of views of computing that are current in current (American)
culture and some of the implications of those.
• Computer as a tool (Lawler, 1999)
  The idea of ‘tool’ implies both extending our manipulative ability and acknowledging that
  the tool must ‘work on’ something; that distinction is the fundamental one in computing,
  of the difference between data and information. For a technician, this is probably the most
  useful and productive metaphor available for computing. For anyone else, however, there
  are problems: understanding the distinction, and identifying what the valid data might be.

• Computer as a machine (Lawler, 1999)
  The car is a machine that you don't need to understand in order to use. While there are
  plenty of people who enjoy tinkering with cars, many more want nothing to do with such
  activity. They want to use the car. This they can do, because functionally a car is simply an
  extension of a natural activity (movement) whose operation can be transferred from other
  learned activities. In short, you learn a high-level skill associated with the hardware -- you
  learn to drive a car. What you do with it is then open to your own intentions. Computers
  don't really work like this -- you can never really escape from needing to know something
  of what happens ‘under the hood’, and anyone who thinks otherwise has actually
  misunderstood the possibilities and limitations of computers.

• Computer as a workplace (Lawler, 1999)
  The computer is the ‘place’ where we work, yet unlike the physical workspace, there are
  real limitations on how this ‘place’ can be customized to how users work.

• Computer as filing cabinet (Lawler, 1999)
  Almost universally, the computer uses the metaphor of a business office, with a filing
  cabinet full of folders, each containing some kind of information, each with some kind of
  label. This is actually very misleading. The problem is that real files all hold the same kind
  of thing (legible papers), while computer files can hold anything at all, much of which isn't
  legible at all, at least by humans. So eventually, the filing cabinet metaphor gets stretched
  too thin to be of use here; the learner really needs to develop a concept of ‘computer files’
  which is really quite different to ‘filing cabinets’.

• Computer as toy (Lawler, 1999)
  Game playing is quite ubiquitous; the computer is a site for fun.

An exploration of learner’s conceptions may well be enhanced by attending to high-level conceptions
such as these. But what of more detailed lower-level conceptions? From experience and research no
more than of having been a practicing teacher of computing, I would offer that the following
ideas/concepts would be worthy of exploration – to explore what understanding learners have of
these:

• Physical understandings which connect the orientation of the mouse and pointer; the
difference between insertion point and pointer
• A ‘windows’ interface’: what it means to have active window; background tasks etc
• What it means to click; click-drag, double-click, etc
• The filing system, drive letters, folders and the idea of saving and ‘save as’; absolute and
  relative paths
• The idea of an object and the idea of ‘selecting’ text or objects
• The idea of layers
• The distinction between viruses, adware and hardware/software faults
• The internet, networks and connectivity

These are only suggestion, and at this time there is no research to present in relation to these.
However, what we can attend to are some techniques that may be useful in identifying and exploring
particular ideas and conceptions.

**Identifying learner’s conceptions**

White and Gunstone’s (1992) *Probing Understanding* discusses a range of techniques and
approaches, including concept maps, predict-observe-explain (POE) tasks and interviews about
instances (IAI). Some of these have particular application to the scientific knowledge (eg POE and
IAI) but their work is applicable to the full range of disciplines. It could be argued that POE and IAI encapsulate what is central to a science education from a constructivist perspective. The literature review above has shown that there is no such ‘seminal method’ relating to computing. Notwithstanding the possibilities of techniques such as analysis of literal and figurative speech, think-aloud protocols, and cognitive task analysis, there are several procedures which can be readily adapted to probing understanding of ‘common place’ computing. In this section four of these are listed and briefly discussed.

**Primed clinical interviews**

Fensham and Lui (1999) have developed the technique of ‘primed clinical interviews’ originally in chemistry education. Using this technique, there are two groups of questions. First of all, there are questions which orientate the learners to their ideas about the topic. Secondly, learners are asked a second group of questions which do not ask for their knowledge of the topic to be described directly, but in terms of analogies; none of the analogies are a precise representation of the subject, but each of them has at least a grain of truth about them. A schedule developed in relation to learner’s understanding of internet search engines is as follows:

<table>
<thead>
<tr>
<th>Are you familiar with:</th>
</tr>
</thead>
<tbody>
<tr>
<td>• a library</td>
</tr>
<tr>
<td>• a library catalogue</td>
</tr>
<tr>
<td>• a librarian and how s/he helps you research</td>
</tr>
<tr>
<td>• the game of Chinese whispers</td>
</tr>
<tr>
<td>• the index page to a book</td>
</tr>
<tr>
<td>• the idea of speed reading</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>In what ways might a search engine might be like:</th>
</tr>
</thead>
<tbody>
<tr>
<td>• a library</td>
</tr>
<tr>
<td>• a library catalogue</td>
</tr>
<tr>
<td>• a librarian</td>
</tr>
<tr>
<td>• a game of Chinese whispers</td>
</tr>
<tr>
<td>• the index page to a book</td>
</tr>
<tr>
<td>• someone who is a speed reader</td>
</tr>
</tbody>
</table>

**Figure 1: Sample schedule for ‘primed clinical interview’ in an ICT topic**

**Mapping for conceptual change**

Concept maps have been well described elsewhere (eg White & Gunstone, 1992, ch. 2), and they should be noted as a powerful way of ascertaining both students understanding of the range of concepts and the interconnections between them. In the field of ‘common place’ computing, there is, as yet, very much an open question as to what sorts of words might be expected that learners would include in such a diagram.

With this in mind, we can refer to Kern and Crippen (2008) who have presented a novel way of using concept maps to explore how knowledge changes over the course of instruction. The process commences with a jointly constructed (ie teacher with students’ help) list of terms which can be used. This generates a list of concepts/words which students can use to form an initial concept map. Only those terms agreed to by a whole class can be used, and then if the class also recognizes that there is a linking statement which would connect the concept to the others.
So, each learner produces a concept map using the terms agreed on during this process. Learners then engage with further learning, and then evaluate the (initial) concept map and proceed to revise it as necessary by working in small groups. The process of mapping-learning-reflection-refinement can be repeated several times. It would thus provide a means of both facilitating conceptual change and charting how it proceeds. In a field where the terms used are ill-defined, this technique may make the user identification of that central to the method.

**Double-bubble diagram**

Simpler than a Venn Diagram (see White & Gunstone, ch. 8), a Single Bubble diagram is a simple way of identifying the parts, or characteristics, of a topic under review (see Model Learning, 2009). Its main purpose is to initiate thinking on a topic and to capture it easily without any undue organisation.

A Double bubble allows you to make a direct comparison between two topics, people or events.

**Figure 2: Sample single-bubble diagram**

**Figure 3: Sample double-bubble diagram**

A double-bubble diagram is constructed by:

1. Constructing two Single bubbles on the two items to be compared
2. Placing the two Single bubble side by side
3. Identifying the characteristics that are similar and those that differ
4. Redrawing the two central bubbles, placing the similar characteristics between them. Draw lines joining these shared bubbles with the two central bubbles.
5. Arrange the dissimilar bubbles around their associated central bubbles, drawing lines to link them to either of these two main bubbles.

The double-bubble method can be useful in identifying shared perspectives, agendas or interests of two opposing people or views, comparing and contrasting or examining change over time. I have tentatively tried using this approach to ask middle years students to compare Word and Powerpoint; what I found was that students either compared trivialities (ie ‘you can make things bold’) or the
purpose of the software (ie ‘you could make a greeting card in either’). Therefore, it is possible that a fair amount of scaffolding maybe required for students to lead students toward crucial differences such as variation in data structure.

**Predict-observe-explain**

Predict-observe-explain (see White & Gunstone, ch. 3), is a strategy to uncover individual learner’s predictions, and their reasons for making these, about a specific event. The essence of science is observation and prediction so it is very well known in that field. It also works well in mathematics, particularly in statistics, and I would suggest that there are similar opportunities in computing.

Here is a possible example⁷. Consider a small section of text in relation to an image in a word processor.

---

**Figure 4: Sample paragraph and graphic from a word processor**

There are a number of attributes of both the paragraph of text and the image which are important to achieve the layout the author requires, for instance, hard or soft character returns, image in front or behind text, image anchored to paragraph, character or page. Learners could be provided with several examples of this paragraph which appear very similar but are in fact implemented in different ways – how does the text or graphics behave when additional material is inserted or deleted? A “prediction” is possibly trivial, but the explanation may reveal a range of understandings of the data structures and the applicable operations.

**Think boards**

A Think Board is a format that allows learners to represent their concept knowledge and understandings in multiple ways.

---

**Figure 5: Sample thinkboard**

Using a think board technique, the concept or question is placed in the center box and the student develops appropriate representations of that concept. The above example is pertinent to mathematics, but the representations could also be: stories, real things, pictures or symbols. There is no reason to think that the number of alternative representations needs to be exactly four, but certainly ‘several’ representations that have some comparability would suggest that the learner has an established concept.

---

⁷ I would like to thank Roland Gesthuizen for his help in developing this particular example.
Referring the word-processing example above, a think board could be used to help learners represent what they perceive as going on in the different paragraphs. They could build an example with a string of beads and some wooden blocks (or other equipment); they could explain it in words; they could draw a picture of it, etc.

CONCLUSION

The concern which I have shared in this paper is that rather than accepting learners as being “IT savvy” at face value and without enquiry, there should be a more earnest effort to identify the knowledge that learners have about computers and computing. That is, to presume that rather than being authoritative, that such knowledge may well be robust, idiosyncratic, familiar and pragmatic, but unique to the individual learner.

Literature relevant to the field has been considered, as has issues of knowledge in the domain of computing. Some suggestions which may guide future research have been presented. The field of “learner’s conceptions of ‘common place’ computing” has been identified as one in which there are numerous possibilities for research ventures. So (2002) has observe that there has been a paradigm shift in science curriculum over the last 20 years or so, attributable to Science educators taking constructivism more seriously. Maybe attention to the same issues may set computing education on the same trajectory.

REFERENCES


