Conceptual Understandings of Computing

Paul D. Chandler
Yarra Valley Grammar

Introduction

Over the past 20 years or so, increasing attention has been paid to constructivist understandings of student learning. In the subject area of Science the concept of "children's science" has received considerable attention. Osborne & Freyberg (1985, p. 12) comment 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.

This paper is not a fully developed treatise on the learning of computing or the teaching methods related to the teaching of computing concepts The purpose of this paper is more *to wonder* - to wonder whether there is a analogue in the field of computing to what has become known in the field of science education as "children's science"; to wonder what would happen if computing educators too seriously "naïve understandings" of computing[1] in the same fashion as "children's science" influenced areas of science education; to wonder what techniques there might be for effectively documenting and understanding learner's naïve understandings; and for wondering whether there might be some important expert/novice differences which may be revealed through insights into naïve understandings.

An Historical Reflection

It has always been the case that students would have entered a science classroom having had exposure to the world around them throughout their lives; children's science has taken seriously the idea that the understandings that they may have acquired about the way the world works should be taken seriously as a departure point for future teaching and learning in Science.

Twenty years ago, life experience with computers could not be assumed. More or less the only experience students had of a computer was that which they were

1 In other words, some kind of "children's computing". "Children's science" doesn't apply exclusively to children, but to learners prior to any teaching and learning of formal science, and is therefore misleading; the use of the terminology "naïve understandings" or "naïve conceptions" is also known in the science education literature. Later in the paper, I clarify why I have chosen the term "naïve understandings" in the computing context.

given in the laboratory. Very few students had a computer at home. Well-meaning teachers (including 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' 'cut' (hard for the novice user to imagine with pre-GUI word processing software such as Wordstar and Xardax), one could clear a desk space, place a newspaper on it and create a toolbox of scissors and glue and proceed to act out what the students were doing with the software.

These days, a great deal of work with computers in schools exists in environments where it can be safely assumed that students have ready access to computer resources outside of school time. There is a great deal of information about what computers are and how they work presented both formally and informally in the mass media. In other words, before students step inside our classes, knowledge about computers has been both personally and socially constructed. If it was ever true that students in computing classes were blank slates, coming in bearing to relevant knowledge, then that ceased to be true some time ago and is certainly not true in the present day and age. Students are frequently described as "ICT savvy" (Christopherson, 2006) or "digital natives" (Christopherson, quoting Prensky).

By not re-teaching skills which students bring to the classroom and by not using glib examples to illustrate all-too-familiar operations, teachers take this reality seriously. But there is a deeper level to this, and that is that the copy-and-paste example was, in its time and place, a genuine and appropriate way to build on students' prior knowledge. In these days, we need to have a deeper and more exacting perception of the understandings our students have of computing so that we might allow our teaching to build on this prior knowledge.

Before entering into a discussion of how we might glimpse into the understandings our students have of computing, it is necessary to discuss the nature of that knowledge and how it is developed and constructed.

Epistemology

An epistemology is a theory of knowledge – informing us of how knowledge might be said to be acquired and integrated with other knowledge. The particular epistemology which I wish to draw on is constructivism. Though dating back at as early as the 16th century (von Glasersfeld, 1988), constructivism has received considerable attention in recent decades and is as central to the works of theorists such as Piaget and Vygotsky (Ridgway & Passey, 1991). It is a constructivist perspective which has informed the children's science movement, and is central to the notion of naïve understandings, and the importance of probing understandings, expressed in this paper.

The central tenant of constructivism is that we come to know our world by interacting with it (Brookes, 1987, p. 64). In the constructivist paradigm, knowledge is not a representation of an observer-independent world. Rather, individual learners construct understandings of concepts, situations, people, etc.

which are viable in the experiential world of the knower (von Glasersfeld, 1988). Knowledge arises neither from the subject nor the object, but rather through their interaction. It can be said that there is no pre-defined body of knowledge.

To adopt a constructivist epistemology is to assert 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.

Taking a constructivist perspective, I would contend that we Osborne & Freyberg (1985, p. 12) could be re-worked for learning in relation to computing:

- from a young age, and prior to any teaching and learning of formal computing, children develop meanings for many words used in computer use and views of the world which related to computing;
- children's ideas are usually strongly held, even if not well known to teachers, and are often significantly different to the views of experts; 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 teaching

At least, I would suggest, there ought to be a concerted attempt to put these contentions to the test.

Furthermore, we may make the mistake of confusing "ICT savvy" or "digital native" with "expert". To explain this point of view, consider an example from the world of science education, and that is the understanding of the world as either flat or spherical. For many centuries, that the world was flat was established commonsense. Even to the average person today, to think of the world as flat is quite productive in terms of measuring distances. It's only when you go beyond the everyday experience and wonder why the stars move above the earth as they do or why the horizon is curved and our approach to it is endless that 'the earth is round' becomes a more productive explanation. The point is that you can live in a society which has a socially constructed 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 99% of the time, your ideas may still be at considerable variance to an expert position.

In the spirit of wondering, as expressed in the opening paragraphs of this paper, *I wonder* what would happen if we, as computing educators, rather than presuming that knowledge that our students is authoritative, took the view that such knowledge may well be robust, idiosyncratic, familiar and pragmatic, and unique to the individual learner.

An Example: Understandings of Internet Search Engines

Having discussed the nature of knowledge, it seems desirable to move from theory into a practical example of naïve understandings of computing. My interest in this field was sparked by White & Gunstone's (1992) *Probing Understanding* which discusses a range of probes which include concept maps, predict-observe-explain (POE) tasks and interviews about instances. Some of these have particular application to the scientific knowledge; many of them are time consuming, requiring detailed interview schedules. Fensham & Lui (1999) have developed the technique of 'primed clinical interviews', which was still focused on scientific knowledge, has then advantage of being a much more efficient technique to administer, and I sought to try to find a what to use this to effect with computing concepts.

The chance came soon enough, after I made note of some bizarre ideas students told me about search engines, such as "they search the world to find something for you". Whilst there is, at a technical level, some 'absolute fact' as to what search engines are and how they work, I'd prefer to take the view that learners have good reason to be believe what they say. In other words, they have constructed some knowledge, based on experience in life or in the classroom, to think in certain ways about search engines (for example). It is therefore interesting to probe what, in fact, they do believe.

Using the 'primed clinical interview', there are two groups of questions. First of all, there are questions which orientate the student to their ideas about the topic. Secondly, students 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. Here is the question schedule which I developed:

To what extent do you believe that the following are true statements about internet search engines *(orientating questions):*

- *They search for data on every page of the Internet*
- *They locate key words by searching the world*
- *They are just like a library*
- *They search a certain part of the Internet for you*
- *A search engines searches the pages which are connected to it*

Write a few sentences to describe why a search engine might be like *(the probe):*

- *a library*
- *a library catalogue*
- *a librarian*
- *a game of Chinese whispers*
- *the index page to a book*
- *someone who is a speed reader*

Can you thinking of any other analogies for search engines? *(more probing)*

Which do you think is the best analogy? Why? *(more probing)*

I've only tried this question schedule once, and that was a few years ago. I'd asked students to work in groups, and the major finding was the students tended to defer to the member of the group who had the best reputation for being knowledgeable about computing. It is also possible that all the analogies were not actually well understood by all students.

Whilst much more work needs to be done on this particular probe, I hope it is now clearer what I mean by seeking to understand the knowledge which our students have about certain ideas in computing.

Some Naïve Understandings

In this section, I would like to give several examples where the learner has developed some level of understanding which robust, idiosyncratic, familiar and pragmatic, unique to the individual learner, but not necessarily commensurate with expert knowledge in the field. As a result, particularly when encountering situations different to which they have previously encountered, students seem to find computing difficult.

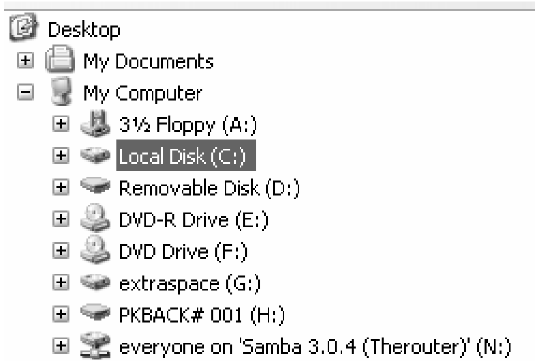
For a semester, I taught an adult learner of non-English speaking background for his first introduction to computers. Despite being shown the process of creating, saving and printing, each lesson he focused on placing his disk in the drive correctly, and then proceeded to create and print his work, only 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 all about, and how wonderful it was to be able to save his work, and how this would improve his efficiency each lesson. Whilst at the very early stages of this student's computer use, this anecdote suggests the possibility that **the accepted and definitive model of computer architecture (ie primary and secondary memory) is neither intuitively understood nor easily grasped**.

Another example concerns another adult learner had reason to save a document 'to the floppy disk'. This person had been using computers in her teaching, but was very conscious of a need to learn more. A telephone conversation to suggest that she save the file to the floppy disk so that she could work on it elsewhere did not faze her. When, in fact, the work was not to be found on the floppy disk, this was a cause for distress. To diagnose the problem, she was asked how she went about saving it, to which she replied that she had clicked on the 'save to disk' icon on the tool bar. It transpired that, thinking that a picture of a floppy disk meant save it to a floppy disk, she had clicked on  rather than 

The discussion that followed, however, revealed that this learner had no conception that floppy disks are often identified as 'drive A:' or that the other drive letters have any meaning useful to the novice. This simple anecdote reinforces the possibility that the accepted and definitive model of computer architecture is neither intuitively understood nor easily grasped. It also suggests that **the association between drive letters and secondary storage devices** as implemented in Microsoft Windows **should not be**

assumed to be intuitively grasped, nor should the 'iconic vocabulary' expected of computer users.

A similar issue can be identified from test responses from junior secondary students to an item on a multiple choice test. Students were shown the following diagram:



The question asked the students to identify the network drive and the local hard drive. This diagram was somewhat different to the network that they are used to (they are familiar with drive H: being the network drive), and they were not given the option of reporting drive C: as the local hard drive. The correct responses – G: for the local hard drive and N: for the network drive – should have been easily identifiable by attending to the icons, but a considerable diversity of responses was received. Clearly, there is something interesting about how students approach such a question, and potentially their understanding of **relationship between secondary storage devices and drive letters**.

Another idea which are sometimes presented by students which cause me some consternation is that **anything that goes wrong with my computer is a virus**. The view is often so well entrenched that an attempt an explanation of the difference between viruses, adware and a hardware fault often lands on deaf ears. It is clear that student understandings (particularly misunderstandings) are a powerful influence on current learning.

The above examples are hardly an exhaustive list of "incomplete understandings" or "misunderstandings" which the teacher of computing might encounter in his/her students. It is clear that, unless challenged, these are indeed robust, idiosyncratic, familiar, pragmatic, and unique to the individual learner. The learner has already learnt something, through some means or other (eg friends, family, the media, computer manuals, the internet, experimentation or even formal classes) and has good reason to believe what they do and act in the way that they do. I would suggest that the term **'naïve understandings'** is therefore better than 'misunderstandings' because it **draws attention to the learner having good reason to believe what they do and act in they way that they do**, and that this represents the commencement of a journey which must be fashioned by experience to a set of understandings which may be radically different to those presently held.

An exhaustive list of areas of "naïve understanding" relating to computing would be an extensive work in its own right, although is clearly of interest. I will return to a broader discussion of this issue later, but firstly it is necessary to describe what else has been written in this field.

Literature Review

As this is an opinion piece, it won't be claimed that an exhaustive literature search has been undertaken. However, an ERIC search has been conducted and several useful papers have been identified. It is clear, however, that there is no major published corpus of work in this field.

Powers and Powers (2000) have conducted some research in to 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).

Later in their paper, Powers and Powers observe that

> 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)

This is exactly the sentiment which I have been expressing throughout this paper and if Powers & Powers' consider that their work in CSIS is broader than the discussion of the intellectual frameworks of beginner programmers, then what I am suggesting in this paper is broader still.

Powers & Powers identify several preconceptions which effect student learning in CSIS, which are interesting though fairly narrowly focused on CSIS. However, their research method is not discussed. It is not clear whether their paper is primarily a reflection on what a teacher has casually observed after much time in the classroom, or whether a data gathering probe analogous to the ‘interview about instances’ common in children’s science research (Osborne & Freyberg, 1985) has been employed.

Scott Brandt has written a number of papers discussing the mental models of learners as applied to the role of the reference librarian (Brandt, 1997; Brandt, 2001; Brandt & Uden, 2003), again drawing on a constructivist understanding of student learning. The most recent of these papers, “Insight into mental models of novice Internet searchers” is interesting for three reasons. Firstly, Brandt and Uden use the ‘mental model’ terminology of cognitive psychology to distinguish between “conceptual models[2]” (an authoritative, idealized understanding of how a system works) and “mental models” (a user-specific collection of knowledge which builds a foundation of understanding and provides the tools for problem solving in a given domain). Brandt & Uden have also clearly presented their research methodology and approach to data collection – that of applied cognitive task analysis. Thirdly, the work is interesting because it is directed at applying an understanding of student’s mental models to the differences between experts and novices.

Primarily, the literature review conducted this far has revealed that there is no major body of work in the field of naïve understandings of computing concepts, although there is consolidated research in some related fields such as computer programming. There is much scope for work in identification of research methodology and suitable and efficient data gathering techniques.

What concepts to investigate?

The intention of this paper is to argue for a concerted effort to probe learners about why they do what they do – to strive to document the thinking that accompanies action and user explanation. I refer specifically to computing – not to IT or ICT which would considerably broaden the discussion; *my concern is with being able to drive the software and hardware*. Given that the literature doesn’t have much to offer, progress will only be made through a process of bootstrapping. There are two sources of information which can seed this process: personal experience and the literature.

Personal Experience

One needs to start somewhere, and the reader also possibly needs some suggestions to more readily understand the sort of thing that I am referring to. So, based on nothing more than anecdotal observation and intelligent guesswork, I am wondering whether there might be important computing concepts which are associated with the following:

2 This terminology is another reason to use the term “naïve understandings” rather than “naïve conceptions” which are sometimes used synonymously in the science education literature. From the children’s science perspective a naïve conception would be the constructs developed by the learner; from the cognitive science perspective ‘conceptual’ refers to an authoritative, idealized understanding. This confusion can be minimized by keeping the work ‘concept’ out the discussion as much as possible by referring to ‘understandings’ instead.

- 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 notion of ‘selecting’ text or objects
- The idea of layers
- The distinction between viruses, adware and hardware/software faults
- The internet, networks and connectivity

All of these are fairly fine-grained. Consistent with the discussion earlier in the paper, broader concepts, may be important, such as:

- computer architecture (with its primary and secondary memory), maybe important; and
- the computer as an “n-dimensional highly interconnected system” rather than a “2 dimensional simplistic device” are also important.

Research Literature

Powers & Powers (2000) identify some (mis)understandings which may have broader implication. There is insufficient space to discuss these in detail, but they can be found on `http://isedj.org/isecon/2000/408/ISECON.2000.Powers.txt`

- Computers as analogue devices
  Students are accustomed to devices that respond linearly with variances in their input. This is not 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.
- Complexity and levels of abstraction
  When confronted with the differential in complexity, a computer architecture student remarked "I used to get [angry] when my computer crashed. But, the more I find out about what's going on inside, the more amazed I am that it doesn't just crash all the time."
- The computer as a giant brain
  That is, the computer is already aware of everything: “Why do I need to write code in my program to determine the date? The computer already knows that!”

Lawler (1999) has discussed a range of views of computing that are current in (American) culture. There is language which is typical of each view, and each has its problems, each its opportunities. There is insufficient space to discuss these in detail, but they can be found on `http://www-personal.umich.edu/~jlawler/meta4compute.html`

- Computer as a servant
- Computer as a race
- Computer as a tool
- Computer as a machine

- Computer as a workplace
- Computer as filing cabinet
- Computer as toy

The work on metaphor, often stemming from Lakoff and Johnson (1980), is important because it asserts that our though is shaped by the metaphors which surround us, and furthermore further more, the metaphors found in the language that we use give insight into our understanding. For instance, Lawler asserts, that 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". One's mental position in relation to the above metaphors (and others) is an important influence on the understanding of computing.

Building on Personal Experience

Given that there are hardly any clear areas of conceptual understanding of computing which are identified in the literature as particularly worthy of examining (and those which are suggested are very broad in scope), a generative process needs to be adopted. That is, the wisdom of teachers needs to be used in conjunction with what we have and undergo several phases of refinement.

In endeavoring to help teachers think deeply about their method of teaching, Mulhall, Berry and Loughran (2003) asked to teachers to consider a science topic using a grid. The heading of the page was the name of the topic. Rows were headed:

- What you intend the students to learn about this idea
- Why it is important for the student to know this?
- What else you might know about this idea (that you don't intend students to know yet)?
- Difficulties/ limitations connected with teaching this idea
- Knowledge about students' thinking that influences your teaching of this idea
- Other factors that influence your teaching of this idea
- Teaching procedures (and particular reasons for using these to engage with this idea)
- Specific ways of ascertaining students' understanding or confusion around this idea

Columns were used for the various "big ideas". So for instance, for a topic on the circulatory system at middle secondary level, the 'big ideas' might be: 'model of a closed continuous system', 'functions to service the needs to individual cells', 'body systems are very dependent on each other for the proper functioning', 'blood is a complex substance', etc.

Whilst not the focus of Mulhall, Berry and Loughran's work at all, this approach would seem to offer possibilities helping teachers identifying the "big ideas" which we are endeavoring to foster in our teaching. It would seem that it would be worth at least trying to ask groups of computing teachers to work on such a grid for a computing topic, referring to the research literature and each others' experience as necessary. It may then be productive to create probes into student's understandings in relation to each of these ideas.

[Participants in the seminar will work in groups on such a grid]

Another way to continue to explore what the 'big ideas' are, and student's understanding in relation to these is through a wiki framework. One has been established at http://pdchandler.wikispaces.com All teachers are invited to contribute their thoughts.

Improving Teaching about Computing

My interest in big ideas and naïve understandings of computer is to raise the possibility that there is more than one way of looking at using a computer than amassing a large catalogue of skills. I have heard more than one teacher complain something to the effect that, *Teachers are often heard complaining that "10% of my students have good computer good computer skills, eg can effectively use Word, can logically organise their home drive and understand the basics of naming files and folders etc. The other 90% range from very basic computer skills to knowing how to turn it on and open a program (5-10%).*

Anecdotally, there seems to be some concepts or ideas which some students 'get' and others don't, which allows them to more freely among software and systems without much difficulty. One thing that we can do is to look carefully at whether there are some understandings which 'the better' computer users in our classes have that 'the strugglers' don't yet have, and that we might be best off to make a point of teaching these conceptual understanding *explicitly*, taking into account the understanding that they have now.

It could be argued that in our efforts to contextualise the use of computers throughout the curriculum insufficient attention has been paid to developing computing concepts. There is a case that in addition to integrating computer use across the curriculum, and ensuring that students have a range of skill to that they are functional with the technology that we should deliberately develop teaching strategies to develop broad conceptual understandings of computing. The computer is increasingly used as an excellent tool in the development of concepts in other subject areas, and this serves for computer use to be relevant and interesting; it is equally important to foster the conceptual understanding of the computer itself.

Fensham & Lui (1999) have identified some questions which direct a teacher's thinking away from the teaching of skills to the teaching of broader conceptual foundations. Among their suggestions are:

- What wider or general sense or reality does 'this' content exemplify and open up to the learner?

- What basic phenomenon or fundamental principle, what law, criterion, problem, method, technique, or attitude can be grasped by dealing with 'this' content as an 'example'?
- What situations and tasks are appropriate for helping students grasp the principle of the content by means of the example of an elementary 'case', and to apply and practice it so that it will be of real benefit to them?

It is not impossible to believe, for instance, that knowledge of frames in a word processor, might aid an understanding of master slides in PowerPoint, and layers in Paint Shop Pro or Dreamweaver. My experience of IT teaching and curriculum has not been to cast it in these terms, but to be concerned with the acquisition of application-specific knowledge. (I would not suggest that this particular example is, definitively, an important broad-based concept; that remains to be identified through relevant research.)

There are some relevant practical implications of such an approach. In circumstances where students are thought of a "IT savvy", and where it is reported that students are learning more of their ICT skills outside of school than in the classroom (Meredyth et al, 1999), **it may make more sense for the purpose of curriculum to be directed around developing a good conceptual framework rather than teach more and more skill. It may even be that certain conceptual understandings are best developed by engaging at activities away from the computer.**

Casual observation of students over many years suggests that some students who have had a fair degree of exposure to computers have picked up some 'enabling' skills, knowledge and concepts – skills, knowledge and concepts that in some way enable them to more quickly 'catch on' and adapt to new types of software. This is very much the expert/novice differences as considered by Brandt & Uden (2003). **It is, after all, very much the role of computer education to help all students develop understandings that are increasingly productive and viable.**

Conclusion: Suggestions for the Future

This article has identified that scant attention has been paid to naïve understanding of computer concepts. The following should be explored:

- the extent to which children, from a young age, and prior to any teaching and learning of formal computing, children develop meanings for many words used in computer use and views of the world which related to computing;
- the extent to which children's ideas are usually strongly held, even if not well known to teachers, and are often significantly different to the views of experts; and
- the extent to which these ideas are sensible and coherent views from the children's point of view, but often remaining uninfluenced or influenced in unanticipated ways by teaching.

Anecdotally, it is thought that there are a range of computing concepts, the individual's understanding of which would be interesting to investigate. Expert/novice differences in understanding of these may be helpful in supporting weaker students or devising teaching strategies for all. Indeed, from this perspective it may make more sense for the purpose of curriculum to be directed around developing a good conceptual framework rather than teach more and more skill. It may even be that certain conceptual understandings are best developed by engaging at activities away from the computer.

However, exactly what computing concepts should be investigated remains unclear, and is most likely to be clarified by an iterative process involving teachers' considering their work (and the 'big ideas' encapsulated therein) carefully, and a framework for working through this has been suggested. A wiki has been established for continuing discussion of these issues.

http://pdchandler.wikispaces.com

References

Brandt, D. S. (1997). Constructivism: Teaching for Understanding of the Internet *Communications of the ACM, 40*(10), 112-117.

Brandt, D. S. (2001). Reference, Mental Models and Teaching Technology. *Reference Librarian 74*, 37-47.

Brandt, D. S, & Uden, L. (2003) Insight into mental models of novice Internet searchers: Adaptation of Task Knowledge Structures. *Communications of the ACM, 46*(7), 133-136.

Brookes, M. (1987). Curriculum development from a constructivist perspective. *Educational Leadership, 44*(4), 63-67.

Christopherson, P. (2006). From the VCAA Corner. *INFONET (The Journal of the Information Technology Teachers' Association), 16*(2), pp. 2-3

Fensham, P. J., & Lui, J. (1999). *Teachers' Capability to Transpose the Content of Science Topics in the Curriculum*. Paper presented at the Contemporary Approaches to Research in Mathematics, Science, Health and Environmental Education, Deakin University, Melbourne, Australia.

Lawler, J. M. (1999). Metaphors we compute by. In D. J. Hickey "Figures of Thought: For College Writers", Mayfield Publishing. [http://www-personal.umich.edu/~jlawler/meta4compute.html, accessed 18 Nov 2006]

Lakoff , G.& Johnson, M. (1980). Metaphors we Live by. The University of Chicago Press.

Meredyth, D., Russell, N., Blackwood, L., Thomas, J., & Wise, P. (1999). *Real time. Computers, change and schooling. National sample study of the information technology skills of Australian school students*. Griffith University, Brisbane: Australian Key Centre for Cultural and Media

Mulhall, P., Berry, A., & Loughran, J. (2003). Frameworks for representing science teachers' pedagogical content knowledge. *Asia-Pacific Forum on Science Learning and Teaching, 4*(2). [http://www.ied.edu.hk/apfslt/v4_issue2/mulhall/mulhall8.htm, accessed 18 Nov 2006]

Osborne, R., & Freyberg, P. (1985). *Learning in Science: The Implications of Children's Science.* New Zealand: Aukland: Heinemann.

Powers, K. D. and Powers, D. T. (2000, Nov). *Constructivist Implications of Preconceptions in Computing*, Proceedings of ISECON '00 (Annual Conference for Information Systems Educators)

Ridgway, J., & Passey, D. (1991). A constructivist approach for educational computing. *Australian Educational Computing, 6*(2), 4-9.

von Glasersfeld, E. (1988). *Cognition, construction, knowledge and teaching.* Washington, D.C.: National Science Foundation. (ERIC Document Reproduction Service ED 294 754)

White, R., & Gunstone, R. (1992). *Probing Understanding*. London: The Falmer Press.