The second step: A constructivist approach to classroom computing.

The Second Step in classroom computing is a profound one, founded in learning theory, implemented in modern technological practice and, best of all, steeped in fun, constructionist activities for the learner. The constructionist approach to school based software development is now well-established see Harel (1991), Harel and Papert (1991) and especially Kafai (1996, 1997) for expositions. This paper, exploring a constructionist approach as enacted in one school, addresses itself to classroom teachers, curriculum coordinators and those in the academic community who are interested in a different way of teaching and a different way of learning.

On the surface we have has children as software designers producing computing solutions to curriculum demands, underneath, we have the construction of the learners’ own knowledge structures. The contents of the software are then distributed across the community of learners. Thus the learners are very much given a sense of control and ownership of their learning milieu. They also develop a strong feeling of belonging to a knowledge, skills and communication rich collaboration of those like-minded in the self-construction and multiple distribution of cognition.

Furthermore, the learners are actively encouraged by all of the elements of a driven community of learning to build into their software solutions fun elements of aspects of the learning environment in which they live; distributing their enjoyment to the enrichment of all. (see Minsky, 1980) More importantly this approach strengthens the community bond.

Figure 1 shows the overall scheme of things, providing, at the base, the developing learning milieu and, above that, the various elements which interact as many small agents. The elements include learning paradigms which inform and enrich the practice. Each box in Figure 1 contributes a vital ingredient to the learning environment and shows the extent of the theoretical influences on our Technologically Enriched Learning Environment projects (TELE projects – a name which arose playfully from a student suggestion that these activities were better than watching TV – a considerable claim for the particular child!). Included within the paper are a number of example projects.

There are a number of computer rich environments such as Computer Supported Intentional Learning Environments (CSILE) (Scardamalia 1990), Child Driven Learning Environments (Strommen 1992), the hypermedia systems of Hill (1998) and the programming environment Microworlds to name a few. A theoretical overview of these environments can be found in Vosniadou (1994).

The unique strength is the inhouse production of the entire system, for this we use the computer programming language, Visual Basic, and the database package, Access.

Distributed knowledge, intelligence and cognition

As Pea (1993) has observed, when we watch the practices of cognition in action it is noticeable that the learner’s mind rarely works in isolation. Rather intelligence is spread across peoples’ minds and the environment. This concept of distribution has become important in the literature (see Salomon, 1993 for a comprehensive round-up). Our purpose is best served here by way of a concrete example.

Some years ago a group of students and staff designed and wrote a software package of some complexity for a forensic science unit. In each subsequent term one, Year Seven students use the package to solve the “murder” mystery of their school Principal detailed below. This is an adventure, investigative style activity with many twists, which draws heavily on the students’ prior knowledge of forensic science. During the course of the activity the students find themselves, their teachers and other members of their school community as suspects within the mystery. This gives rise to considerable motivation for the students to find out where this content has come from.

During the previous year, students

Practical Support for TELE Design

Theoretical

SCHOOL IDEAS
Staff & Students

DISTRIBUTED INTELLIGENCE
Gavriel Salomom & Roy Pea

SCHOOL ACTION
Staff, Students and Curriculum Computing Support

CONSTRUCTIVISM
Jean Piaget

SOCIAL INTERACTION & SCAFFOLDING
Lev Vygotsky

KNOWLEDGE AS DESIGN
David Perkins

CONSTRUCTIONISM
Seymour Papert

INSTRUMENTAL ENRICHMENT
Reuven Feuerstein

THE SOCIETY OF MIND
Marvin Minsky

CHILD PROGRAMMING CULTURES
Idit Harel & Mitchel Resnick

CSILE
Marlene Scardamalia & Carl Bereiter

Figure 1: Technologically Enriched Learning Environments is a project involving Year 7 & 8 students and teachers at a Melbourne school & a researcher from the Faculty of Education, Monash University, Melbourne.

who had already solved the murder mystery rewrote the story for the next intake. These rewrite students gathered information about fellow students in the junior section of the school who will be called on to solve the mystery the following year. The structure of the software package is left intact while its content – particularly the identity of the murderer – changes annually. This rewrite is then posted onto the computing system to remain dormant until required. Thus we have the rewrite students distributing their knowledge about the school community to the next intake of students.

The story, however, does not end there. By doing the rewrite the students are exposed to the methods which the previous group have used; thus rewrite students are distributing their cognition, developed during the construction of their version of the mystery, over the system to be used by the next generation. The rewrite students of one year enjoy the benefits of the insights – the intelligences – of the students of the previous year and can put this distributed talent to telling use in the construction of their own cognitive structures. We have here a nice example of what Perkins (1993) calls person-plus. The person-plus concept has each of us as an individual learner plus the advantages to our cognitive development of the contributions of the community in which we find ourselves.

What advantages do we see in this distribution of knowledge? Hewitt (1996) describes the wealth of missed opportunities for learning in the traditional classroom.

We saw many examples of realized opportunities in the TELE projects as they developed. For example one student had considerable expertise on linking Visual Basic to Access databases which was made available to his fellow students from the start of the project. This provided a vital opportunity for the others, as such linking was to be a significant aspect of the project. Other students had well-honed internet searching and downloading skills. Others were skilled scanners. By the end of the project James had made generally available his system software knowledge for running video in Visual Basic. The teacher could delegate all of these responsibilities and concentrate on his area of expertise – content. We share Hewitt’s concern that traditionally oriented teachers would have difficulty knowing about such expertise or of taking advantage of it if they did. We advocate the building of knowledge maps from the start which display in graphic form the existing relevant skills and talents of the individual participants in the project. These maps can be drawn manually by the students in a brainstorming session at the start of a project.

A note of caution, though: our experience is that Hewitt’s missed opportunities can also be greatly increased in number when learners are placed in a computer rich environment which permits them to work in isolation. It is important to take advantage of distributed processes. We have found that when students work alone at their own machine for an extended period of time, a significant subset become inward looking and do not share their discoveries and inventions. Students need to be encouraged and also unskilled to work co-operatively.

We are not talking here only of collaborative student learning but also of apprenticeship learning of a special type. In TELE environments it is often the case that a student has the expertise over the other students and the teacher. Technological advances are producing students who are domain experts and their educational environment must learn how
to take advantage of this knowledge and its associated practical skills.

**Constructivism**

Constructivism and the work of Jean Piaget are central to TELE projects. Papert (1988) considers that despite the trend away from Piaget’s specific results in which two restructuring of thought. Whatever the knowing systems individual constructing new knowledge (often one external, the teacher, one internal, the learner) resolve a conflict with the individual constructing new knowledge structures. This is followed by self-reflection on the impact of the conflict on the individual’s view of reality and self-regulation to bring about a developmental restructuring of thought. Whatever the width of the theory the central force of self-construction of cognitive structures is integral to TELE projects.

**Constructionism - the N word as opposed to the V word**

Seymour Papert would have at a simplistic level the definition of constructionism as learning-by-making. However he would immediately counsel us to construct - a sense of constructionism much richer and more multifaceted, and very much deeper in its implications, than could be conveyed by any such formula.

We have constructionism as the kernel of TELE projects; the students make computing software with a view to constructing their own knowledge frames (Boyle, 1998). TELE projects acknowledge the need for environmental enrichment and variety of experience within the learning milieu. We would also side with Feuerstein (1980) in that cognition is modifiable and view TELE projects as examples of his Instrumental Enrichment Programmes in which such modification can be effected.

Papert (1991) considers that the construction of a public entity is an especially effective method of knowledge construction. We have found that developing a piece of computing software is a powerful realisation of this.

All of the participant learners in TELE projects construct computing based artefacts that they distribute across the community as a Perkins-like People-Plus. Research at MIT builds on this concept of distributed constructionism.

The...three main categories of distributed constructionist activities: discussing constructions, sharing constructions and collaborating on constructions. (Rosenick, 1996)

TELE projects emphasise and magnify these concepts and we see the construction of artefacts by each learner as the lynchpin of effective cognitive building.

**Social interaction and scaffolding**

Some will have noticed that our concept of distributed cognition distances us somewhat from the classical constructivism of Piaget. We move further away in homage to the work of the Russian psychologist, Lev Vygotsky (1986), in being strong advocates of the value to the learner of social interaction. Thus our philosophy, whilst acknowledging the importance of internal constructions of cognitions, equally advocates the value of the external learning environment and the significant contribution of social interaction.

Vygotsky sees the direction of the development of thinking as being from the social to the individual. We stress in TELE projects the active learner in the responsive environment with ideas bouncing between both. Our classroom practice constantly encourages interaction between learners, presentation of ideas by individuals to groups or the class and presentation by groups of work in progress.

Most important of all, we are with Vygotsky in exploiting zones of proximal development, that is, the areas where learners can proceed by themselves before requiring external help from more knowledgeable peers or teachers (Moll, 1992). We have found that learners are more knowledgeable than their peers about something. The exploitation of that something is a most potent force in the cognitive development of the individual; and the pursuit of such exploitation highlights the importance of the distributed cognition of the community of learners as they realise the significance of their own contribution.

Within our community of learners it is vital that scaffolding is provided to...
guide the novice learner as required. The
development of computing software is far
from a trivial activity and we always
provide starter-kits which individuals or
groups can use for a kick-start. A starter-
kit may consist of a Visual Basic Project of
a few forms, which are the basic building
blocks in Visual Basic, to be taken as a
starting point. The starter-kit is developed,
enhanced and then discarded when the
learners find their own way. We have
found that such safety nets are important
in confidence building, and, after a time,
the learners develop starter-kits of their
own in the light of their experience to be
distributed over the community.

Child programming cultures -
knowledge as design

We see the process of designing a
project to be a vital ingredient in the
construction of cognitive structures. For
theory we owe much to Perkins (1986) and
Gargarian (1996) but in practice we turn
for guidance to the Instructional Software
Design Project of MIT’s Idit Harel (1991) at
the Hennigan Elementary School in
Boston.

We have found that it is important
to involve the learner in all stages of TELE
projects and that no stage has proved
more important than design. There is no
better litany of value than Harel’s own five
reasons for design with illustrative
comments provided by our own students:
(Harel, 1991 p xviii-xx)

First reason: Design motivates
learning.
It was my job just to set up the database but I
got really interested in the animals in
Antarctica and took that on as well.

Second Reason: Designers make things
happen.
We got so much stuff on The Renaissance from the Net that we didn’t know what to do
next. Then John [the Project Designer]
showed us that we’d have to bring it down to
fifty characters for each event.

Third Reason: Design evokes self-
knowledge. Designers make personal
connections between the affective and
the cognitive.
It’s a bit scary [being in charge of a
project’s design stage]. You feel you
have to get it right but you just can’t first time -

Fourth reason: Designing a product
promotes consideration of intended
users, clients, customers - the
community of other that designers
serve.
We liked the first colour scheme but when we
ran a demo on the TV display no one could
read the writing!

Fifth Reason: Design is integrative
and holistic.
I enjoyed it [being the Project Designer]
because at the end you can see how everything
fits.

School based ideas

All of the ideas about software
enhancement of the curriculum to
increase learning should be generated in-
house by either students or their teachers.
That, of course, is not to say that those
contributors are not heavily influenced by
external factors in formulating ideas. We
want, however, every idea which we
implement to have an identifiable internal
owner as this raises the profile of both the
contributor and the contribution within
the learning milieu. Some protest that
students cannot devise ideas for
educational applications. That is a moot
point, which begs our issue: we are not
interested, per se, in the educational value
or significance of a student-constructed
software artefact. We see the act of
creation by the learner, in all its stages, to

be the vital cognitive structure-building activity
essential to all effective learning. The plus
with the TELE project is that the existing
cognitions which lead to the new
development are internal to the system,
either individual or pre-distributed
cognitions through the entire project. Any
software development which proves
useable within the curriculum is
considered a bonus not an objective.

This is a purposeful switch in the
direction of education away from instruc-
tionism and towards constructionism; it
represents a fundamental change in the
teacher-student relationship. As Seymour
Papert says in relation to the new type of
mathematical thinking introduced by
Logo’s Turtle:

... opposition to this kind of mathematics is
firmly rooted in prevailing epistemological
ideology: A shift here challenges more than
particular knowledge it challenges the very idea
of knowledge. Inevitably the resistance will be
fierce. (Papert, 1991 p 22)

The “prevailing epistemological
ideology” is now being challenged, not
only in mathematics education, but
through the curriculum, learner driven, in
computer rich environments taking the
Second Step.

School Based Action

The process of production for TELE
projects is a complex one that involves the
application of the full systems analysis life cycle for software production. (See Kendall, 1998 for details). In essence the cycle commences with academic staff and technical support staff making a detailed examination of curriculum statements to identify possible software applications which enhance learning. Technical support then design and implement the required scaffolding software with frequent reference back to the academic staff. The proposed project is given a final evaluation before being described within the curriculum statement whence it becomes available for classroom use.

In the implementation of TELE projects being described the software was written in Visual Basic and Access. It should be emphasised that hardware and software concerns are close to the bottom of priorities in TELE project systems analysis. Any computing facility capable of driving the school’s curriculum needs will suffice.

Technologically enriched learning environments: some examples.

The above outlines the theoretical considerations which permeate the philosophy of TELE project development. The final evaluation of a project before classroom use considers the implications of distributed knowledge through constructionist principles for the given project. Also we examine how the scaffolding and social interaction within the project may enhance the learning conditions for the students and how design principles will guide students to a successful outcome. Throughout the classroom-based phase of the project’s development we continue to emphasise the philosophy, as will be apparent from the following examples.

The bottom box in Figure 1, into which all of the others either indirectly or directly feed, is the product of the Technologically Enriched Learning Environment project. All of these software elements have been designed and produced by students in conjunction with teachers in response to a perceived curriculum need.

Forensic Science: As indicated above, at the first level, students, who are in the early stages of Year 7 (13+ years), use a software package written by a previous generation of students, in order to help them solve the mystery of the murder of their school principal. This is a small element in a unit of work in Forensic Science. At the second level, later in the same year, these students will re-work the software to be used by the next Year 7 intake. The new intake always contains students who are known in advance and they are built into the redesigned investigation, along with their teachers and other members of the learning environment. The previous generation of learners distribute their accrued knowledge over the system to the cognitive advantage of their successors. This is a nice example of fun learning as described above. The student designers are encouraged to make the story for the next year as humorous as they can at the expense of those who are going to use the software. We see in this example a paradigm of collaborative constructionist design which is very much a fun learning activity. Another plus is that by keeping the structure of the software package constant, with a rewrite of the user interface and of the contents of the database, the package has a wide range of applications. It is currently being developed to investigate the alleged murder of Tutankhamen.

Figure 2 shows the Main Menu for Forensic Science. There are complex software aspects within each option. For example Figure 3 shows a screen, from one of the animated sequences in the Technical Information section, which inform the users about various aspects of forensic science. Thus we have a non-trivial, collaborative software development product having utilitarian value for the users, which becomes a paradigm design process for the next group of developers.

Another series of projects involves the establishment of what we call collaborative databases. Currently there is a French Database, a Database of Scientists, a World Database and a Database of the Renaissance. All of these database activities require each learner to contribute some ten to twenty records of information to the database. Thus learners construct their own cognitive view of the topic in hand. Then the learners collaborate to combine the mini-databases thus giving in total a very substantial body of information. Any given final database is much too arduous in scope for an individual learner to establish. With this approach each learner
has available a research resource which can only be established by communal effort.

The following example of the use of such databases illustrates the cognitive power for the community learner, a power not available to the learner alone.

**Database of the Renaissance:** Students in the history unit on the Renaissance are required to go and research some aspect of the period. They may research art, science or politics; the selection of areas can be altered from year to year. They also have had to set up a small Access database whose structure is supplied as part of the software scaffolding. Into this database they enter up to twenty records on the topic of their choice. Figure 4 shows a typical database entry. The students need to precis and paraphrase the information to fit the deliberate constraints of the database design. Each of these mini-databases is then merged into the existing main database to produce a resource which no individual student could produce on their own. The students then make a class presentation on some aspect of the Renaissance set by the teacher using the main database as a research tool. Over just a few years this tool can grow into a significant asset which all the students know has been established by their peers and to which they have all contributed.

**Software Development Elective:** The above examples illustrate quite specific curriculum demands but we also offer a more flexible software development environment to Year 9 students who are skilled designer/programmers and who want to continue to learn to develop more sophisticated software units, perhaps for use in the curriculum.

Thus a body of curriculum software is amassed, with students producing units on endangered species, volcanoes, world or regional maps, etc., and these software units are built into the system. These units are interactive and usually contain a quiz of some sort to be used by other students for fun or, if the content is suitable, built into a teaching unit.

**Ancient Egyptian Mathematics:** Computer Programming as a Trojan Horse. This unit is a serious attempt to bring about cognitive development in mathematics underachievers. This is what Solomon (1991) would call an effect of computers rather than an effect with.

From the work of Minsky (1986) and Feuerstein (1980) we have developed a model of learner thinking which attempts to alter prior cognitive structures. See Boyle (1998) for a non-technical exposition. In particular we have found that many students, who are otherwise satisfactory school performers, consistently under-perform in mathematics. The cognitive model suggests that such students have a fixed view about mathematics, which inhibits further development within the domain.
This deflecting frame needs to be bypassed in the learner and we are investigating using computer programming to develop an explanation of the algorithm and display it in the panel. In order to achieve this they would need to understand the algorithm and then transmit that understanding to others.

Solomon first suggested the Trojan Horse aspect of programming where the experience is used to achieve another purpose and we have implemented those ideas in the following way.

Suitable subjects in Years Seven (n = 23) and Eight (n = 24) were invited to take part in a programming project in Visual Basic to develop educational software which would illustrate, for Year 6 students, various mathematical techniques used by the Ancient Egyptians. Suitable scaffolding was provided in the shape of a partially written software package that the students were to develop to a finished product.

All of the students (bar one Year Eight student new to the school) were experienced users of notebook computers across the curriculum and had had programming instruction in Visual Basic. This was expected to minimise any Hawthorne effect: achieving beyond the expected due to the novelty of the approach.

Figure 5 shows the scaffolding for the multiplication algorithm and Figure 6 the scaffolding for the section on coordinates.

In Figure 5 the students were to develop an explanation of the algorithm and display it in the panel. In order to achieve this they would need to understand the algorithm and then transmit that understanding to others.

All 47 students demonstrated a mastery of the ancient algorithm by providing a description of its workings and arriving at the correct result for two different multiplications, as demonstrated by the production of a working piece of software. The example multiplications were 14 X 23 and 19 X 27. All of the students were given a printed worksheet of the two multiplications and asked to verify the results of the calculations by the standard school maths pencil and paper algorithm. It was emphasised that the purpose of this exercise was the crosschecking the ancient algorithm.

The following table shows the results:

<table>
<thead>
<tr>
<th>Problems</th>
<th>Problem 1 &amp; 2</th>
<th>Problem 1 only</th>
<th>Problem 2 only</th>
</tr>
</thead>
<tbody>
<tr>
<td>Year 7</td>
<td>8</td>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>Year 8</td>
<td>7</td>
<td>2</td>
<td>5</td>
</tr>
</tbody>
</table>

The results:
- Number of students who failed on Problem 1 only
- A further six students in each year refused to perform the hand check and would only verify the ancient algorithm results by electronic calculator. All twelve were successful by this method.

Thus 100% of the students were successful in using the ancient algorithm and 6% were successful using the standard school algorithm. Successful was predefined to mean getting the correct results for both problems.

Success in one framework was denied in the other. From the view of Minsky (1986) a framework for the standard algorithm held default values that led to incorrect results. A newer "ancient algorithm" framework held default values that led to correct results. The frameworks appear independent. Lewis (1998) emphasises that unlearning may be necessary before cognitive restructuring can take place. There was no attempt in this instance to analyse the errors made in the application of the standard school algorithm.

In Problem 79 from The Rhind Mathematical Papyrus, shown in Figure 7, all of the students correctly identified the error made in the original by the scribe and developed an explanation of the geometric series, explicitly stated in Peet's hieroglyphic translation of the papyrus, which satisfied their peers. Note also how this hieroglyph error gives us a "fun" link back to the error in the scaffolding software in Figure 5.

In their final written evaluation of the project 68% of the students did not acknowledge that they had been taking part in mathematical activities. Detailed case studies on seven of the students showed that they maintained a very clear distinction between school mathematics and other mathematical activities. This dichotomy was supported by interview data, recorded observation and log sheets maintained by each student. For the sessions on the ancient multiplication algorithm 62% of the students did not tick the box to indicate that they had been involved in mathematical activity.

In terms of our mental model: the subjects are maintaining two separate frames, one for the school algorithm and one for the ancient. Whether we can circumvent the effects of the inhibitory
school maths frame requires further analysis. The Trojan Horse aspect of computer programming, whereby a subject only realises that they can do something successfully after the event, was in play. Through the vehicle of computer programming, children, who are identified failures in school maths, successfully manipulated mathematical algorithms as complex as those in the school frame, when working in the "Ancient" frame. What then posted back into the system for the next generation to avail themselves of the fruits of previous cognition in the pursuit of their own knowledge building. These posted notes are strong cognitive parcels! The students continue to build the parcels throughout their school career, forever posting and unposting to an ever more important, relevant, and friendly database - a journeyman resource known to be our creation and existing on the school’s computing network to which all the students have access. Thus we have a dynamic knowledge:

Awareness that knowledge can enable the acquisition of other knowledge, and can be conjoined in various ways to play various roles is critical to intentional learning. Naive learners seem instead to regard knowledge as basically inert stuff that can be accumulated and sorted and arranged. Without some more lively conception of what knowledge is about, students would seem to be poorly equipped to overcome the tendency of conventional schooling to produce what Whitehead (1929) condemned as “inert knowledge”.

(Scardamalia, 1989 p. 54)

So stands the power of community for the individual learner in TELE projects.

Conclusion.

We spoke earlier of the influence upon our Project of Artificial Intelligence, in particular the work of Minsky (1986). We now see coming out of MIT (Porter, 1996) applications of The Society of Mind using databases over the Internet. From the musing of these theoreticians we expect a reworking, a realignment and an expansion of Technologically Enriched Learning Environment projects - the individual, the local community the international community.

The Second Step in classroom computing has taken our students away from the apparent safety of instructionism, which had been their lot before, and into the unnerving uncertainty of constructionism. One staff member new to the approach was working with a seasoned group on the Antarctica Project and had to wait over a hundred and fifty minutes before a student spoke to him. They were intent upon the task and he was a thwarted interventionist! However like all good natural Vygotskians he held off and instinctively recognised the moment for a zone of proximal development intervention. His sense of relief was obvious!

The Second Step has a long way to go in practice and in its research, but the opening possibilities of expanding internationally into the person-world construct has given us a glimpse of the Third Step.
REFERENCES


