Constructivist Learning Using Simulation and Programming Environments
MIE2002H Readings in Industrial Engineering I
Calum Tsang
May 5th, 2004

1.0 Introduction

The original direction of this reading course was to prepare for a thesis project involving the creation of a curriculum for introducing high school students to the field of engineering. Like many such initiatives, the original request from the University was technology-oriented: Take a popular computer education product, the Lego Mindstorms system, and turn it into a course for student enrichment, encouraging the best and the brightest to apply to the engineering program.

Perhaps like Seymour Papert described, “the phrase ‘technology and education’ usually means inventing new gadgets to teach the same old stuff in a thinly disguised version of the same old way” (Papert 1971, 1-1). As a result, the focus of this set of readings was directed to making such an experience less about technology but more about learning methods and how to design a curriculum that would be redeeming for the students involved. As the Mindstorms product was based on Papert’s work, it made sense to investigate the principles of constructivist/constructionist learning.

This paper begins with an exploration of the concepts of constructivist learning and the early work of developing the LOGO computer language, then delves into some of the practical applications of these methods, such as the development of interactive construction toys and the use of software development in teaching.

Some of the questions kept in mind were:

- How does one pick meaningful projects which are interesting for students to investigate?
- How do you build an environment for such a course?
- How does one get above the initial learning curve of the technology at hand?

As a secondary goal, the aggregation of practical design principles for the creation of technological learning products was examined:

- How do you design a product to encourage constructivist learning?
- How do you overcome mental models and technical learning curves?

2.0 Definitions of Constructivism/Constructionism

Constructivism is a theory about learning, one where the learner has a “a self-regulated process of resolving inner cognitive conflicts that often become apparent through concrete experience, collaborative discourse and reflection” (Brooks and Brooks 1993: vii). Simply put, when someone doesn’t understand something, it bothers them internally. This nagging is resolved when one has the chance to experiment by doing, share the experience with others, and have time to think about the confusion.

Jean Piaget suggested that assimilation and accommodation of a concept resulted in a temporary cognitive stability. New structures of knowledge are formed when the learner fits their existing model to accommodate the new information. (Brooks and Brooks 1993: 26).

While Seymour Papert of the MIT Media Lab has done significant work in the area, there are several authors who also have perspectives on this topic and their discourse is also reviewed here.

MIT AI Lab Memo 247: Teaching Children Thinking (Papert, 1971)

Teaching Children Thinking is an early precursor to *Mindstorms*, introducing some of the questions and ideas found in the later, more comprehensive text. Early on, Seymour Papert asks "(w)hy don't we teach them to think, to learn, and to play?" (Papert 1971, 2-1) instead of the traditional instruction we give students. He encourages the emphasis on how children think and acquire knowledge from experiences instead of attempts at blindly teaching facts.

This early text describes educational methodologies Papert terms "Pop-Ed Thinking" which include "blank mind theories" where the student is encouraged to repetitively focus on memorization, "getting it theories" that typify a person's knowledge as a binary model of either understanding or not in one shot, and "faculty theories" which encourage children to believe there are capabilities they do not have, such as being "mathematical minded".

Because the child is "constantly engaged in inventing theories about everything" (Papert 1971, 2-2), the author instead proposes "creating an environment in which the child will become highly involved in experiences of a kind to provide the rich soil for the growth of intuitions and concepts for dealing with thinking, learning, playing and so on" (Papert 1971, 4-1). This environment would grow to become the LOGO computer language and later referred to as a "microworld".

A good part of the paper serves as a prologue to *Mindstorms*: A section of the paper is devoted to explaining the mechanics of LOGO, and many subtle concepts he would later explain are briefly touched on as well. For example, the concept of syntonic learning, where the child is encouraged "to study himself in the situation and try to simulate his own behaviour" (Papert 1971, 6-1) is quietly introduced through the description of a mechanical turtle robot used to demonstrate LOGO geometry.

The paper has a delightful idealism to it. Papert offers that "evidence is accumulating for the thesis that there is scarcely any child who cannot be involved in some computation project" (Papert 1971, 5-5). He suggests the experience of LOGO as intellectually and emotionally involving students, which later is expounded on in *Mindstorms*.

Mindstorms: Children, Computers and Powerful Ideas (Papert, 1980)

Seymour Papert's *Mindstorms* is considered a seminal work in the use of computers in educating children. This text also forms the foundation and direction of numerous educational technology products, most notably the LEGO/Logo and Lego Mindstorms toys. Hence, it was an apt starting point for this reading project.

Mindstorms encases the development of the LOGO programming language in a number of ideas about the process of education and examples of how computers could be used to teach increasingly complex concepts. At the time, presumably, these concepts were quite radical as affordable computers were only beginning to become available in the marketplace.

Papert begins with the concept of Piagetian learning, in his words, "learning without being taught" (Papert 1980: 7) based on the work of French psychologist Jean Piaget. The child is described as a builder who needs materials from his/her world to learn from. These materials are not necessarily traditional textbooks, but instead the real world experiences the child encounters. Children innately build theories of the world on their own, constructing their own foundations for knowledge. Papert cites Piaget's work investigating children's ideas on such topics as nature and conservation, and puts a focus on using those innate learning abilities as a method for teaching. The author offers a personal example of discovering gears and wheels as learning materials that later were used as a foundation for understanding algebra and mathematics.

An entire chapter is devoted to a topic Papert refers to as "Mathophobia" which builds on the "faculty theories" concept he introduced earlier. Formal mathematics education is described to be separate from the exploratory everyday learning we experience from birth, and while a basic tenet is that we begin our "lives as eager and competent learners" (Papert 1980: 40), it quickly devolves into a fear of not being math-minded. Much of this fear stems from traditional methods used in schools. He challenges us to create an environment, which he terms "Mathland" in which mathematics experience can be garnered. Of particular note is a discussion on the use of laboratories for teaching physics (Papert 1980: 139) which criticizes traditional science teaching as using a series of experiments where the answer is already known.

The book describes the LOGO language and begins with the tenets of Turtle Geometry, an on screen triangle that children would instruct to draw lines. At the most basic level, LOGO and the Turtle are described as educational "objects to think with" (Papert 1980, 11), used to learn a language of movement and geometry.

A key concept which Papert suggests is syntonic learning, which describes the process of learning linked both to mental and physical self. One aspect is ego syntonicity, which stems from the interest and satisfaction derived from using LOGO. As people, students have "intentions, goals, desires, likes and dislikes" (Papert 1980: 63) and a positive feedback cycle is generated from the learning process. The child wants to work with the computer instead of being forced to. Body syntonicity refers to the child's "sense and knowledge about their own bodies" (Papert 1980: 63). In working with the Turtle Geometry, Papert has observed the student understands geometry by way of his/her own spatial awarness, often pretending to be the on-screen or cybernetic Turtle.

Another concept in *Mindstorms* is that of the "microworld", a specialized environment for learning with appropriate materials for a given subject. Papert writes, the "design of the microworld makes it a 'growing place' for a specific species of powerful ideas or intellectual structures" (Papert 1980: 125) The author uses the example of a Newtonian microworld, where the LOGO Turtle Geometry system has been modified to support dynamics with objects called DynaTurtles. Instead of exposing the student to the Newton's laws by statement, the child is encouraged to interact and learn by using objects inside an environment programmed to abide by those laws.

The problem with using formal statements of knowledge only asserts Papert, is that it is difficult to develop the inherent intuitions children have and nurture links between those informal ideas and the laws being presented. Students already "think a great deal about their thinking" (Papert 1980: 145) and constantly "debug" them, trying to make they know or intuit fit with what they are being told or perceive.

One of the key effects of LOGO is that it encourages the student to debug. Traditionally the realm of programmers looking for mistakes in their code, the author suggests debugging as an important part of the learning process where children discover mistakes in their conceptual understanding of a problem and correct them. "Typically in math class, a child's reaction to a wrong answer is to try and forget it as fast as possible" (Papert 1980: 61) while in the LOGO environment, debugging is inherently encouraged as the child wants to correct the onscreen image drawn by the Turtle. This reinforces the right answer by the process of discovering the flaw, instead of blind repetition.

LOGO was designed with an emphasis on modularity with procedures. While introducing the language, the author explains how students are deliberately encouraged to break down processes into steps, and then debug those steps to achieve goals. Apart from the short term goal of becoming better programmers and finishing their assignments with the computer, it encourages a method of deconstructing problems into smaller pieces, which later can be used to reconstruct methods of tackling larger problems. Papert considers this "a powerful addition to a person's stock of mental tools" (Papert 1980: 155), one of many epistemologies available to students, as it fits not only the way computers process instructions but also how people live their lives day to day.

The use of LOGO is not isolated to an individual and a personal computer. The text describes how in Brazil, children participate in a samba school, where members prepare for the upcoming street carnival. The school is populated with parents and grandparents, making the environment trans-generational as well as a social affair. During the process of preparing for the event, children learn from elders on the various styles and techniques of dancing. In this community environment, "(l)earning is not separate from reality…(n)ovice is not separate from expert, and experts are also learning" (Papert 1980: 179). While there are significant differences, Papert believes a LOGO teaching environment should be similar to these schools, creating a LOGO culture. His view of education appears to be holistic, both social and educational.

Knowledge as Design (Perkins, 1986)

David Perkins, writes about an alternative though similar approach for learning through the perspective of design. While he does not outright declare a constructivist viewpoint, early in the book the author proposes we view "knowledge as structures adapted to a purpose, just as a screwdriver or a sieve are structures adapted to a purpose" (Perkins 1986: 3). Like Piaget's concepts of building mental structures to gain cognitive equilibrium, the use of a mental design approach explaining this knowledge is offered.

Perkins enumerates four design questions for us to ask when using the theme of design on knowledge. These are to ask the purpose and structure of a design, then to discover examples and arguments for it. A number of situations where the design theme can be used are offered as examples, some obvious and others rather obtuse. For learners, he suggests applying these rules and keeping a watch for "disconnected knowledge", the cognitive dissonance when something does not make sense. When these questions cannot be answered, it signifies the need to continue searching for the reasons for a given design. For teachers, material should be presented in a structure than answers the four design questions while watching for disconnections in the learners understanding of the matter (Perkins 1986: 33).

The author refines his concept of design in ensuing chapters and gives additional approaches for learners to gain deeper understandings of designs. Intentional design, labelled "deliberate" is contrasted against evolutionary or "natural" design, while both are opposite "non design", which are phenomena which happen. (Perkins 1986: 35) Given the number of different design analyses one can use, the text eventually offers strategies for dealing with and choosing which mental model to use. The parallels between this text and others continue when Perkins recommends learners should "(c)hange designs as often as circumstances require but announce it" (Perkins 1986: 41). This is similar to the debugging concept introduced by Papert or the guidelines suggested by Brooks and Brooks.

The viewpoint of examining processes and act of breaking down procedures, encouraged in *Mindstorms* as a tool for learning, are also covered in *Knowledge as Design.* Papert used this model to encourage the deconstruction and reconstruction of mental models to promote understanding, Perkins similarly offers it as a strategy for understanding the true purposes of procedural designs.

The text offers some interesting activities on the organization of objects through differing design perspectives. By looking at designs at different angles (characteristics), the learner will gain a better understanding. (Perkins 1986: 57)

Papert touched upon the use of constructivist methods for writing and poetry in *Mindstorms* but did not elaborate fully, preferring to stay with geometry and physics. Perkins offers an entire chapter on the use of design in reading and writing, especially in writing organization and critical analysis. Realistically, the author grounds his enthusiasm noting that while "(u)nderstanding the design of a form gets your writing off to a good start" (Perkins 1986: 92) but only serves to ease the structure, not the content of the paper.

Knowledge as Design turns the model around from analysis to creation in later chapters. This is perhaps the most relevant part of the book for this project, as the author presents some design challenges that teachers and learners can participate in, such as writing jokes and developing and testing study methods for schoolwork. However, the most interesting is the idea of studying everyday gadgets and designing better replacements.

Most constructivist texts highlight the need for motivation and personal acceptance of the learning task. Instead of offering ways to encourage personal buy-in of learning as others have written, Perkins offers strategies to mitigate the damage to this "rather delicate flower" that is "easily bruised" (Perkins 1986: 116) such as minimizing negative feedback and optimizing the challenge to suit learners. Perkins poses the importance of appropriate "problem finding" to discover ideal challenges for students. Instead of letting learners determine problems to design for free form, he instead suggests we help narrow down the choices by offering areas which are rich with problems to work from. (Perkins 1986: 121)

The final chapters of *Knowledge As Design* introduce the use of models and argument as objects to use in exploring designs. Perkins closes out his text with some critique of existing educational environments. Specifically, he acknowledges the current classroom approaches knowledge as information and uses closed loop models which isolate the student through exercises and lecturing.

Situating Constructionism *in Constructionism* (Harel and Papert, 1991)

Papert begins the book *Constructionism* with context for the title subject. While the general term is constructivism, the author has chosen the term constructionism. His explanation is that while both share a "connotation of learning as 'building learning structures' irrespective of the circumstances of learning", constructionism also includes the concept that this "happens especially felicitously in a context where the learner is consciously engaged in constructing a public entity" (Harel and Papert 1991: 1). This works particularly well for LOGO and Lego projects when designed cooperatively and shared in a classroom environment.

Papert expands the discussion on constructionism by questioning the issue of style and approach in creation. He notes that some LOGO programmers use a "bricolage" approach, which is much more art-like than the traditional computer science programming paradigm of planning and procedure breakdown. This difference in approach calls for an epistemological pluralism later discussed in further chapters.

He also approaches the issue of relevance and emotional attachment with an observation that by adding new objects such as “cybernetic construction kits” for LEGO/Logo, children might “want to learn it because they would use it in building” (Harel and Papert 1991: 7). This is similar to the utility concern Pratt and Ainley would later discover in their 1997 research with Cabri Geometry.

The Case for Constructivist Classrooms (Brooks and Brooks, 1993)

The work of Papert and colleagues was primarily research oriented, describing experiments at schools with LOGO and other research projects. Jacqueline and Martin Brooks take a more pragmatic stance in *The Case for Constructivist Classrooms*, a guide for practising teachers. Current teaching methods, they argue, are “dominated by teacher talk” and “expect students to identify and replicate the fields of knowledge disseminated” (Brooks and Brooks 1993: 6) from textbooks, lectures and isolated work. Much of this theory is based on “the stimulus-response relationship coupled with positive reinforcement” (Brooks and Brooks 1993: 25). Student thinking is devalued, and when questions are asked, the only answer suitable is the correct answer, discouraging further input from the pupil. While the obvious solution in the authors’ minds is constructivist methods, they quickly acknowledge that this is a difficult task in itself and offer recommendations for effectively using this approach in schools.

The first is to identify problems that students that have relevance to the learner and make them desire to resolve them. Brooks and Brooks note that the “line between cognitive dissonance, which can provoke a student’s desire to persevere, and intrapersonal frustration, which interferes with the student’s desire to resolve dissonance, is a fine one” (Brooks and Brooks 1993: 29). The teacher’s role is to help identify problems that resonate with the learner.

This text is overwhelming practical. The authors, both schoolteachers themselves, recognize that current educational methods are overwhelmingly curriculum based and stress coverage over time. (Brooks and Brooks 1993: 39) To counter, they believe that constructivist approaches may take longer, but actually foster better learning, an up front investment that is not lost later. Students, in their view, are supposed to change their minds, and the teacher should support and query their reasons for doing so. This is akin to the Papert concept of debugging. Not just the literal sense of debugging LOGO, but the concept of revising one’s own mental model.

The second recommendation is to support the learner by giving a whole picture of the learning quest. The reasons for learning must be made clear, illuminated by the steps in the learning process. Yet, just the presence of a “big” idea is not enough: It is the responsibility of the teacher to “mediate the emergence of relevance” and “match curricular questions to the students’ present suppositions” (Brooks and Brooks 1993: 58) to support the learning.

Third, the student's responses must be valued. Teachers, the authors argue, not only have to ask questions of the learner, they must listen to the responses and carefully elucidate additional understanding by following up appropriately. Suppositions each learner makes need to be strengthened or disproved by the learner themselves through the teacher solidly prompting without overbearing or lecturing.

Perhaps most intriguing is a discussion in *The Case for Constructivist Classrooms* about evaluation. The text challenges the concepts of right or wrong, but instead, asks teachers to draw out the reasoning behind suppositions and make it part of the learning process. Brooks and Brooks write "Differentiating between teaching and assessment is both unnecessary and counterproductive" (Brooks and Brooks 1993: 97). It is not a particularly satisfying answer, but telling in that it quietly demands wholesale reform of education instead of offering any concrete advice.

The book closes with practical advice for constructivist teachers. Most of them are highly pragmatic and well reasoned, though the imperative of "eliminate letter and number grades" (Brooks and Brooks 1993: 124), is a particularly tough pill to swallow.

Epistemological Pluralism *in Constructionism* (Turkle and Papert, 1991)

The question of intellectual style is posed by Sherry Turkle and Seymour Papert in Chapter 9 of *Constructionism*. As an example, two university students taking a programming course illustrate that the structured and planned approach favoured by traditional computer science is not for everyone: These two programmers find they must twist their way of thinking or fake the process in order to perform in these classes. Instead, they feel more comfortable working with the computer in their own styles. This difference is the foundation of Turkle and Papert's call for pluralism the representation and construction of knowledge, in this case, programming software.

One style of programmer, termed "bricoleurs", "prefer negotiation and rearrangement of their material" (Harel and Papert 1991: 169), correcting and changing their project over time. One young LOGO user explains his lack of loop structures because it gives him a sense of the pattern and rhythm of the program. The authors describe them as having more of a "conversation than a monologue" with their code (Harel and Papert 1991: 169) and assert that while Piaget or other developmental psychologists would consider their concreteness to be backward, "there can be different but equal voices even where the formal has traditionally appeared as almost definitionally supreme" (Harel and Papert 1991: 175).

Traditional science has maintained a formal distance from the object being studied. However, both the authors, drawing from feminist teachings, consider that some students may form reasoning and a relationship close to the studied objects, imagining their interactions and behaviour. Turkle and Papert list graphical user interfaces as one possibility for tapping into those styles. This cultural change in computer design may make systems easier to use and adapt to than the more formal models computer science has offered so far.

Constructivism: A Psychological Theory of Learning *in Constructivism* (Fosnot, 1996)

Catherine Fosnot gives an overview of constructivism in the second chapter of her book by first comparing it against other models of learning. The most common, behaviourism, is based on the assumption knowledge can be broken down into component skills, which are taught, reinforced and evaluated in a repeating process. Learners are "expected to progress in a continuous, quantitative fashion as long as clear communication and appropriate reinforcement are provided" (Fosnot 1996: 9). The mastery learning model similarly demands each skill is taught until mastery is achieved. Once all skills are achieved at that level, it is assumed the overall concept has been learned.

In contrast, the constructivist model is based on Piaget's idea of equilibration, both biological and cognitive. Equilibration was formed of "self regulated behaviour" between assimilation, "the organization of experience with one's own logical structures or understandings" and accommodation, the "reflective, integrative behaviour that serves to change one's own self and explicate the object" (Fosnot 1996: 13) so we can return to cognitive equilibrium. During this time, we examine our own contradictions to elicit new investigations of knowledge and build structures of cognitive mental systems to help organize them.

The chapter delves into some of the reasoning behind constructivism, including language, society and symbolism. Development, Fosnot writes, has some foundation in language and the dialogue of learning. Piaget studied young children and their internal dialogue with oneself at an early age, while others examined their dialogue with parents. Talking to yourself or others is a way of questioning, explaining and negotiating meaning (Fosnot 1996: 20). Similarly, semiotics, the study of signs and symbols, also plays a role in development, because we need representations to understand shared ideas with our social settings.

Fosnot writes, "(f)rom this perspective, learning is a constructive building process of meaning-making that results in reflective abstractions, producing symbols within a medium" (Fosnot 1996: 27) which become part of our coping strategy for dealing with new information. To summarize, she concludes that disequilibrium facilitates learning, that we need the process of reflection where we find meaning, and that we require dialogue with others to encourage further thinking. These principles are later summarized in practical terms by Gagnon and Collay as a guide for classroom teachers.

3.0 Applications of Constructivism/Constructionism

Software Design as a Learning Environment *in Constructionism* (Harel and Papert, 1991)

Idit Harel introduces a classroom experiment called the Instructional Software Design Project (IDSP) where a class of fourth grade students spent fifteen weeks designing software to teach other children fractions. Though they would learn programming as a side effect, this was a means to an end. Using the constructivist adage "you learn better by doing", Harel adds "best of all by thinking and talking about what you do". (Harel and Papert 1991: 42) The students were asked to design programs in LOGO with little direction other than to teach fractions. They were encouraged to help each other, but each had to complete their own project.

Harel writes, "the actual teaching was not as important as the fourth graders' feeling that they were working on a real product that could be used and enjoyed by real people." (Harel and Papert 1991: 46). Even though the topic of fractions may be intrinsically boring, the design project itself "mobilized creativity, personal knowledge and a sense of doing something more important than just getting a correct answer" (Harel and Papert 1991: 46).

In the classroom, Harel observes that the students can often vary between intense focus and wandering around socializing with other students. Though this could be labelled inefficient, she believes that the students pursued the project with an integrity towards getting the job done creatively (Harel and Papert 1991: 67).

Many of the students grew in their understanding by connecting the fractions topic with the rest of their lives because of their exposure to the subject in the LOGO microworld. Harel stresses the importance of this "situated knowledge", which offers a foundation for mental linkages between the virtual and physical environment. (Harel and Papert 1991: 72).

Harel acknowledges Perkins philosophy of a design environment for learning. In allowing the children to decide on and revise their goals for the fractions project, she observes his assertion that the "problem's meaning is not given by the problem itself" but instead, the designing student will impose his own meanings and goals throughout the project (Harel and Papert 1991: 77).

Formally, the students in the IDSP program did improve on their knowledge of fractions based on standardized testing before and after the experiment period. (Harel and Papert 1991: 51) Perhaps more importantly, by learning how to write educational materials and thinking of insights on how to help the user, the students learned the subject thoroughly, especially through illuminating areas where prospective mistakes would be made. At the same time, the integrated nature of both learning math through designing a software game encouraged the facilitation of other knowledge simultaneously. (Harel and Papert 1991: 75).

Learning Through Design and Teaching *in Constructionism* (Kafai and Harel, 1991)

A review of the social aspects of the IDSP experiment by Yasmin Kafai and Idit Harel is presented in *Constructionism*. The authors observe two collaborative aspects of the IDSP environment: Optional collaboration, where students are allowed the choice to work isolated or with someone else; and flexible partnership, where students can choose themselves who to work with (Harel and Papert 1991: 87). The physical arrangement of the classroom environment was designed to support this, encouraging a studio atmosphere where ideas could be overheard and groups formed quickly for collaboration.

The researchers observed that each student will discover and understand concepts on their own schedule. Some will hear about an idea and immediately pick up on it, while others may be ready for the idea later on. The collaboration environment should allow flexibility in both space and time for students in this manner.

LEGO/Logo *in Constructionism* (Resnick and Ocko, 1991)

Mitchel Resnick and Stephen Ocko describe LEGO/Logo as a "computer based robotics system designed to support a variety of design activities" (Harel and Papert 1991: 141). The evolution of the technical side of this product is better documented in Resnick's 1993 paper, *Behaviour Construction Kits*. This chapter on *Constructionism* speaks to the various activities LEGO/Logo has been used in and its educational outcome.

As with other constructivist approaches, LEGO/Logo attempts to encourage meaningful knowledge creation through the hands on activities. In the case of the Lego product, this includes building robots and contraptions that help solve self-chosen problems that each student ideally relates to personally. Like Harel's IDSP project, a design studio atmosphere is created in the class environment. Resnick and Ocko write of giving design patents and requiring students to document their work in notebooks.

The authors have observed that the LEGO/Logo environment allows students who have "strong design and mechanical skills, but may have been frustrated by the analysis-centered approach of traditional math and science classes" (Harel and Papert 1991: 145). Labelled as poor students, their appreciation for learning often changes because of LEGO/Logo.

There are several learning outcomes from these activities. First, most evident is the fact students learn a number of scientific and mathematical principles while building these Lego models. Concepts such as friction, gears, and even/odd numbers are some of the ideas explained in the process of design with LEGO/Logo. (Harel and Papert 1991: 146) The second is that the students learn about designing, the process itself. A number of design heuristics were extracted by students in the project, such as taking advantage unexpected discoveries, using materials in new ways, and collaborating with others (Harel and Papert 1991: 149).

Resnick and Ocko believe that the system is successful not just because it is hands-on and involving but because it puts students in control of their own projects and learning; offers different ways to get started; and encourages a sense of community. It is interesting to note that two of these success factors are not related to the technology at all, but are organizational constructs set by the teacher.

Behaviour Construction Kits (Resnick, 1993)

In an ongoing partnership between MIT and the Lego Company, Resnick and colleagues began creating interactive toys using Lego bricks and computer technology. *Behaviour Construction Kits* describes the genesis of the "Programmable Brick", an intelligent construction block used to aid constructivist learning. It begins with a description of LEGO/Logo, a building kit for school children, in which Lego building parts are controlled by a personal computer.

The author opens with the idea that construction toys have evolved from building structures (bricks and beams forming arches and spans), to building mechanisms (gears and pulleys forming transmissions and machines), to building behaviours (software code and sensor inputs forming actions and reactions). Not only are beams and gears available, the new construction toy kits now "include computation in the bin of building parts" (Resnick 1993).

With access to such advanced technology in the form of toys, children can learn the concepts of electrical and systems engineering or computer science. More importantly, Resnick suggests that with such behavioural constructs, "children develop new images not only of machines and computers, but of themselves", referencing the emotional syntonicity Papert introduced in *Mindstorms*.

As with previous ventures, LEGO/Logo is based on the work in the LOGO language performed by Papert and extended by Resnick. What differs is that this system is wired into mechanical and electrical parts, extending the virtual microworld Papert designed into the real world. The extension of LOGO into Lego bricks is natural: The abstract nature of LOGO, where students are encouraged to build "blocks" of code forms a neat parallel with the Lego constructions, which are built of intricate pieces.

A number of educational uses are offered, including the use of LEGO/Logo for data acquisition on a hamster and the creation of a Murphy bed triggered by an alarm clock. Eventually Resnick finds that the tethering of this system to a personal computer is a major drawback and offers the development of a more freeform system: The "Programmable Brick" in which microprocessors and electronic logic are integrated with plastic Lego bricks.

Apart from the direct educational benefits of learning concepts like systems feedback, Resnick offers that students exposed to these building kits view constructions as living creatures. They are examined on mechanistic, information and psychological levels, describing their systems both on procedure as well as behaviour.

As with other tools like LOGO, the Programmable Brick “enable(s) young children to explore certain domains of knowledge that were previously inaccessible” (Resnick 1993). Like the gears of Papert’s childhood, these special bricks allow for the exploration of a new microworld and new experiences to be gained from it. The author expands this idea to cover the emerging computerized world, suggesting that if young children can be comfortable with advanced technology if they have a hand in participating in its construction, modification and extension.

New Paradigms for Computing, New Paradigms for Thinking (Resnick, 1995)

Inspired by the increasing use of parallelism in computer science, Mitchell Resnick proposed a number of extensions to the LOGO environment that encouraged the study of decentralized systems. One of the goals of this venture was to “recast areas of knowledge, suggesting fundamentally new ways of thinking about the concepts in that domain, allowing learners to explore concepts that were previously inaccessible” (Resnick 1995).

Most computer languages are relatively procedural, including the early incarnations of LOGO. Given that Resnick and collaborators at MIT were now examining the control of multiple procedures with Lego robots, an adaptation of LOGO to support parallelism was needed. The author set upon “providing new ways for programmers to model, control and think about actions that actually happen in parallel” (Resnick 1995), on the assumption that in many cases, it was the most natural way to express the problem anyway.

The decentralized mindset, observes Resnick, is present in a number of natural and artificial situations, including the movement of a flock of birds, ants foraging for food or the behaviours of managers and staff in an organization. LOGO was expanded to create a new “microworld”, an environment for modelling in the words of Papert, which could be used simulate these behaviours. The models would bring a more friendly face to these novel computer science ideas.

Not only did the upgraded software now support multiple procedures running concurrently, the “microworld” or artificial environment of LOGO turtles was now increased to support thousands of turtle objects. These new turtles could lay down and detect properties in the environment in meta data called “patches”, and detect and avoid each other.

Such changes allow for the turtle “microworld” to change from the arena for basic geometry and calculus applications Papert suggested to an environment for students to study statistics and probability. One such experiment suggests the creation of thousands of virtual turtles and asking each to pick a random direction and drive forward 50 pixels.

While this is impossible in the physical world, the parallel LOGO microworld allows for this quite easily, benefiting students with a practical visualization of statistical distributions.

Resnick offers the upgraded LOGO as an alternative tool to other methods of solving problems: Another experiment suggested is that of optimal angles to kick rugby balls into a "conversion". While other educators may consider using algebraic methods, the parallel nature of this new LOGO variant allows for the use of massive trial and error.

Similarly, the analysis of complex systems such as traffic flow is made trivial by an easy to program for language and the computational power of a computer. Normally the domain of experts, such problems are modelled by Resnick and students easily.

The ability to generate thousands of events is not foreign to scientists or engineers- simulations of atomic explosions are created using a similar method, the Monte Carlo approach. Resnick's expansion of LOGO into this new microworld of massive parallelism allows for new ways of solving problems for students, encouraging the idea that if new "material" is given in an environment for young people to use, they will learn from the experiences gained in experimenting with it.

Programmable Bricks: Toys to Think With (Resnick *et al*. 1996)

As an obvious evolution to LEGO/Logo, the Programmable Brick would eventually develop into the Lego Mindstorms commercial product and was used as a vehicle for a number of MIT Media Lab studies during the mid-90's. Resnick considered the intelligent brick to be one of Papert's microworlds, just like LOGO but expanded to the physical world of construction toys.

One interesting difference in this paper is that Resnick introduces the topic of ubiquitous computing and proposes that the Brick could be considered within that area of research. However, unlike the ubicomp initiatives of intelligent desks and active badges, the Brick would "make ubiquitous-computing activities meaningful to the lives of children" (Resnick 1996a: 444). The paper describes activities performed by local schoolchildren, such as building active environments such as intelligent house monitoring (door alarms, temperature controls) and personal science experiments (a bicycle speedometer). Remarkably, many of these projects are very similar to the types of experiments performed at Xerox PARC by Marc Weiser and team.

From a curriculum design and administration standpoint, Resnick makes some important points about picking student activities. Some of the Programmable Brick activities are not related to ubiquitous computing, but are aimed at the worldview of children. One activity Resnick attempted was a robotic zoo, where students observed real animals and built simulations of them. This activity was chosen "in an effort to make the project appealing to a broad range of students" (Resnick 1996a: 448). It appears personal interest in a project is a significant component of success. Self-relation was also a significant reason for choosing projects, such as the aforementioned personal science experiments. Children were encouraged to design experiments around their everyday lives (riding a bicycle to school, for instance) with the belief that "students are much more likely to make deep connections to scientific thinking" (Resnick 1996a: 448) when the projects are related directly to personal experiences, touching on the syntonicity Papert proposed.

Interestingly, the technical specifications of the Programmable Brick are, by Papert's guidelines for a microworld, designed to present the user with rich facilities to explore and construct with. For example, the Brick and the commercial Mindstorms RCX device both feature inter-Brick communication via IR. This was deliberately added to facilitate the creation of interacting creatures and "explore the scientific ideas of emergence and self-organization" (Resnick 1996a: 444). Similarly, the Brick operating system and language support multiple processes, no doubt stemming from the experiences found earlier in the LEGO/Logo and parallel LOGO implementations Resnick developed before. The study participants tended to want to control multiple processes simultaneously, encouraging this design requirement. The Brick's capabilities act as enablers for children "to perform new types of explorations and to engage in new types of thinking" (Resnick 1996a: 450).

Extremely telling in this paper is the admission that using these tools in a school setting is difficult, which may be one of the reasons why the Mindstorms product has been a critical success but a marketing failure. Unfortunately, Resnick leaves the integration of the Programmable Brick with the existing educational system as an open subject.

Pianos Not Stereos: Creating Computational Construction Kits (Resnick *et al.* 1996)

A market of "edutainment" titles emerged in the 1990's with the introduction of low cost multimedia hardware for personal computers. In this article, Mitchel Resnick and co-authors rail against this passive form of information delivery, termed "stereos", and instead offer more active constructionist projects instead, termed "pianos". Of the three detailed, two have already been covered by previous papers, the Programmable Brick and the parallelized LOGO implementation, StarLogo.

The third, a multiplayer environment called MOOSE Crossing, touches on a previously minimized area of constructionism, that of social relation. Ego syntonicity has already been covered as a required characteristic, however, the social interaction and structure for these activities has not.

Papert described the Latin samba school as a constructionist activity with the added benefit that a social environment was also present. MOOSE Crossing, a multiuser game similar to a Multi User Dungeon (MUD) is both programming exercise, creative writing outlet and social scene. Students are invited as an after school project to play and build in this virtual space, programming automated interactions and crafting textual descriptions of people and objects in this microworld. "Knowledge is not passed from teachers to students but is developed by everyone through their activities and interactions with one another" (Resnick *et al*. 1996b: 48).

The environment is a social place, and there is a desire to share creations in it. Many aspects of real-life relation are present in this community context, and manifest themselves as part of creating projects or experiences in the virtual world. In one case, the authors noted the students built social capital by creating objects others liked.

Also unique to MOOSE Crossing is its verbal aspect: While other projects highlighted so far have leaned towards mathematics and science, this environment encourages writing. Objects in the virtual world must be described, appealing to students who have strong verbal skills. The authors also suggest this as a way to link verbal skills to developing analytical abilities

One can draw parallels with other constructionist activities: The students in MOOSE Crossing work on projects that are very personal, often driven by personal interests and meanings. At the same time, this is expanded into a social context that goes beyond the individual.

The Construction of Meanings for Geometry Construction (Pratt and Ainley, 1997)

Dave Pratt and Janet Ainley explored the use of a dynamic geometry software package called Cabri Geometry as part of a larger technology education project called the Primary Laptop Project where British schoolchildren were given notebook computers. The students, over successive terms at school had already become familiar with using these computers. When the laptops were distributed, each contained a copy of the Cabri software as well as other programs like LOGO and ClarisWorks. (Pratt and Ainley 1997 : 7)

Two groups of students were compared: One batch of students were allowed to use the software free form without constraint or guidance, while a second class were instructed to use the software in a specific guided task. Since the students had garnered significant experience with the laptops over a year, some of them had already "played" with the Cabri software. The package has two sections, a free form drawing mode as well as an analytical transformation mode which allowed objects to be related geometrically together.

Pratt and Ainley found through interviews with the first group that their creations always only used the free form drawing mode. This spontaneous use never extended to the analytical transformation screen. Even when prompted to experiment with the second screen, the students quickly returned to using the first mode, drawing. One student, when given a demonstration, quickly responded with the insight that the analytical mode was like "mechanical glue". Yet even with this conclusion, he went back to drawing free form. "It was clear that they saw it as a drawing package, such as they had used in ClarisWorks" (Pratt and Ainley 1997 : 11). The reason why was that the task suggested to the children during interviews was to draw. For the children involved was more important that they finish their picture than to investigate the properties of using the analytic mode which had “no direct pay-off” (Pratt and Ainley 1997: 13).

In contrast, a second experiment was set up with another group of student. The premise of the task was to build a “construction kit” of geometric parts for younger children to use in drawings. The analytic mode was introduced and its powers of scalability and geometric relationship demonstrated. This class of students intuitively picked up on the benefit of this mode, understanding that the macros this mode offered and the power of being able to rescale objects cleanly would be useful in making the construction kit. Being exposed to LOGO before meant they automatically drew parallels of Cabri macros with the LOGO microworld concepts of procedure.

Pratt and Ainley link this activity with a concept called “webbing”, which defines the environmental materials in the microworld presented. The conclude that a “well designed activity…will optimize the chances of a child exploring and recognizing the value of those structures within the web” (Pratt and Ainley, 1997: 21) and a sense of utility for introducing new tools such as macros or procedures is highly important.

To Mindstorms and Beyond (Martin *et al.* 2000)

Fred Martin and colleagues give a technical overview of the evolution of the various constructivist learning kits MIT has produced over the years, from the floor driven cybernetic turtle robots of the 1970’s, to LEGO/Logo, to the Programmable Brick and finally, the commercialized Mindstorms product.

The article gives some insight into how the various design features of each system were driven by observations made during each successive stage of experimentation with student learners: For instance, the Programmable Brick and Mindstorms RCX brick both feature LCD displays, an obvious high cost item. Classroom use showed that students used it to monitor the status of their code. Regardless of price, the display was then considered critical to aid children to debug their programs, a key characteristic of Papert’s methods (Martin *et al.* 2000: 5). Other elements, such as the iconic programming of the later MIT Logo environments, would be adapted by the Lego Company in their own commercial products.

The latest project is the MIT Cricket system, which features a smaller programmable device that integrates a generalized bus for expansion to input and output devices such as buzzers and sensors (Martin *et al.* 2000: 10). Its purpose is to support children in building instruments for their own personal investigations into science and art. In one application, the Cricket device was used to as a temperature logger to encourage students to measure and understand what a temperature scale meant. In another, the Cricket powered a “nail salon” which polished and dried fingernails.

Building Rules (Goldstein *et al.* 2001)

Similar to Harel's *Instructional Software Design Project*, Goldstein and co-authors set an experiment where British schoolchildren aged six to eight developed videogames using a graphical design environment called Pathways. Pathways was intended to be a system the learner to create objects in an environment and reflect on what they are building, drawing benefit from the entire process.

Much like Harel's IDSP experiment, the students were set loose with little direction other than to create a game. The subject, rules and content of the game were to be decided by the children and the subsequent programming to be performed by the students themselves.

The main thesis of this investigation is that these systems are not easy, especially since many of them work in a mindset that cognitively differs from the perspective of the child creating. As programming tools built to work on a personal computer with stringent requirements, "constructing and manipulating executable representations of objects and relationships has not always been straightforward" (Goldstein *et al.* 2001: 268).

The authors describe three perspectives: Game Inside, the mental model in the head of the child; Game Formal, the logical representation in Pathways software code; and Game Outside, the actual product delivered. One of the major findings from these experiments was that the Game Formal structures did not always align with the participants expectations of how they should work. Most of the children could describe the actions they wanted in the game literally, but the leap to coding concepts such as object collision detection was not obvious, causing the students grief in trying to overcome the gap. Additionally, the division of programming conditions amongst the game objects confused the students. Unaccustomed to object oriented development, the children often expressed global conditions in Game Internal that were the inverse of what needed to be programmed in Game Formal. Which object had to have characteristics modified confused the participants.

The experimental outcome of this set of activities is related to the epistemological pluralism Turkle and Papert write about in Chapter 9 of *Constructionism*. The programming orientation of the Pathways language does not suit everyone and makes it difficult for young children to make the leap between their own mental models and programming.

Constructivist Learning Design (Gagnon and Collay, n.d.)

Conventional lesson planning, writes Gagnon and Collay, is about planning what the teacher will do. These traditional planning models are "based on verbal explanations or visual demonstrations" by the teacher who then combines it with "practice of this method or skill by the student" (Gagnon and Collay, n.d.) An alternative, termed *Constructivist Learning Design*, is to decide what the students will do instead.

Unlike the other articles referenced in this collection, this paper focuses on classroom activities instead of computer oriented projects. The authors present a six element approach to designing constructivist learning activities for school teachers: Situation, Grouping, Bridge, Questions, Exhibit and Reflections:

- In *Situation*, the teachers are asked to propose a situation for students to investigate. The leader must offer the students a process of exploring the problem space through posing questions, making decisions and formulating conclusions.
- The focus then turns to formation of *Groupings*, both of students (social aspect) and materials (microworld orientation). Unlike the MIT activities, it appears the authors are suggesting that a classroom microworld could very well include traditional paper materials.
- Once the stage is set, Gagnon and Collay turn to *Bridging*, a form of determining the extent of and building upon situated knowledge based on the prior experiences of the students.
- Like Harel's IDSP programme, the authors suggest process documentation for each learner. Termed an *Exhibit*, this includes the presentation of ideas and the recording of the thought process that led them there. This is confirmed in the eventual *Reflection* of the entire process, where students are asked to think about how they learned and what they will take with them from the process.
- *Questions* are asked throughout the process to help frame the investigation. These are used to encourage the process, from *Situation* creation to eventual *Reflection*.

Constructivist Learning Design differs in that it encourages teachers to perform assessment in this process, both initially during the stage of *Bridging* and afterwards, creating questions to assess what the students have learned, and for self-evaluation during the *Exhibit* phase. It also directly asserts that such an approach can be used in existing frameworks for classes. Unfortunately, it does not offer example case studies where such an approach has been applied.

4.0 Conclusions

The cynic would note that one of the unanswered questions in all of these readings is practical measurable benefit. Papert might argue this is a remnant of the traditional educational system and not important, but it outside of ideological circles, one still has to consider we cannot ask for wholesale change. It forces a thorny question: How would you sell this to an instutition, be it a public school or even a university offering high school students enriched courses?

Resnick recommends "(d)evelopers of design-oriented learning environments need to adopt a relaxed sense of 'control' "and states developers "cannot 'program' learning experiments directly" (Resnick 1996b: 49). Unfortunately, that poses a problem for those educators working within limited timeframes. Short of unrealistic wholesale educational reform, some sort of incremental approach is probably required.

Most of the projects researched in this investigation are oriented towards young children. Much of Papert and colleagues' work was based on the assumption that by the time these children grew up, the world would be would be full of ubiquitous technologies, a prescient belief that largely became true. For example, most business people have at their disposal a wide variety of experimental analysis tools in the form of spreadsheets and database query packages, which allow them to work on and visualize a problem the way students using LOGO did.

The problem of using "fun" design projects such as Lego and Lego Mindstorms as a method of enticing students to consider a career in engineering is that engineering school is notoriously traditional. While design, communication and synthesis projects are present in many modern engineering curricula, a solid majority of "piped knowledge" courses, involving rote learning and traditional problem solving. To lure students into an engineering faculty with a Papert-esque summer course, and then swap back to traditional methods may appear to be a moral bait and switch.

With that concern stated, some recommended techniques and approaches appear evident for the development of the proposed course:

- Allow each student to choose/discover a problem to work on that is meaningful to them, to achieve the emotional syntonicity Papert describes.
- Offer a multitude of facilities/materials/tools for solving problems, but minimize the learning curve and switching between them for practical reasons.
- Acknowledge that different styles and levels of understanding exist for each of the facilities/materials/tools.
- Encourage the sharing of techniques and code and the cooperating and help of students together. Develop a social but not rowdy atmosphere that encourages collaboration and cooperation.
- Present the real world problems and solutions of practising engineers, but do not take their problems as ones to solve by students. The students likely will neither understand the full context of the problem, nor accept the problems as their own.
- Define both purpose for an activity and utility for the tools being presented.

The projects highlighted in the latter half of this project are tremendously heart-warming and inspiring. Regardless of the practical concerns noted above, it is reassuring that technology has been shown to make a difference in education of young people, even in these isolated cases. Their example can only be inspiration for continued research.

Bibliography

Brooks, J. and Brooks M. 1993. *The case for constructivist classrooms.* Alexandria, Virginia: Association for Supervision and Curriculum Development.

Fosnot, C. 1997. *Constructivism: Theory, perspectives and practice.* New York, New York: Teachers College Press.

Gagnon, G. and Collay, M. n.d. "Constructivist Learning Design." <http://www.prainbow.com/cld/cldp.html>. Accessed April 29th, 2004.

Goldstein, R., Kalas, I., Noss, R., and Pratt, D. 2001. Building rules. *Cognitive Technology 2001 Proceedings.* 267-281.

Harel, I. and Papert S. 1991. *Constructionism.* Norwood, New Jersey: Ablex Publishing Corporation.

Martin, F., Mikhak, B., Resnick, M., Silverman, B., and Berg, R. 2000. To Mindstorms and beyond: Evolution of a construction kit for magical machines. *Robots for kids: Exploring new technologies for learning.* Morgan Kaufmann.

Papert, S. 1971. Teaching children thinking. *MIT AI Lab Memos.* #247.

Papert, S. 1980. *Mindstorms: Children, computers and powerful ideas.* New York, New York: Basic Books.

Perkins, D. 1986. *Knowledge as design.* Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Pratt, D. and Ainley, J. 1997. The construction of meanings for geometric construction: Two contrasting cases. *International Journal of Computers for Mathematical Learning.* 1.3, 293-322.

Resnick, M. 1993. Behaviour construction kits. *Communications of the ACM.* Volume 36, No. 7.

Resnick, M. 1995. New paradigms for computing, new paradigms for thinking. *Computers and Exploratory Learning.* New York, New York: Springer-Verlag.

Resnick, M., Martin, F., Sargent, R., and Silverman, B. 1996. Programmable bricks: Toys to think with. *IBM Systems Journal.* Volume 35, No. 3&4.

Resnick, M., Bruckman, A., and Martin, F. 1996. Pianos not stereos: Creating computational construction kits. *ACM Interactions.* September/October 1996.

Creating next generation blended learning environments using mixed reality, Video Games and Simulations. Sonny E. Kirkley Ph.D. & Jamie R. Kirkley.Â Games and simulations and their relationships to learning. In D. Jonassen (Ed.),Handbook of research on educational communications and technology (pp. 813â€“828). Mahwah, NJ: Erlbaum. Google Scholar. Gunawardena, L. (2003). Theory into practice: The challenge in designing inquiry-based online learning environments.Â Problem Based learning: An instructional model and its constructivist framework. In B. Wilson (Ed.),Constructivist learning environments: Case studies in instructional design (pp. 135â€“148). Englewood Cliffs, NJ: Educational Technology Publications. Google Scholar. Constructivist Learning Theory. The Museum and the Needs of People CECA (International Committee of Museum Educators) Conference Jerusalem Israel, 15-22 October 1991 Prof. George E. Hein Lesley College.Â Principles of learning What are some guiding principles of constructivist thinking that we must keep in mind when we consider our role as educators? I will outline a few ideas, all predicated on the belief that learning consists of individuals' constructed meanings and then indicate how they influence museum education.Â In teaching people to read, the use of different words which have powerful connections for individuals was dramatically described years ago by Sylvia Ashton-Warner18 and widely emulated since.