An Innovative Approach to Teaching Computer Programming to Undergraduate Computing Student Based on Lonergan's Theory

Cornelia Connolly[1], Eamonn Murphy[2] and Sarah Moore [3]

Abstract **- There has been much research into best practices in teaching computer programming and into new and innovative module design and curricula. However students' results and interests do not reflect this effort and retention rates in computer engineering class are still low in comparison to other subjects. This paper presents a framework for problem solving amongst undergraduate computing students. It evaluates the contributions that Lonergan's (1957) 'Insight: A study of Human Understanding' to develop a program to assist computing students in the problem-solving process. By creating the environment and utilizing resources and skills, we envisage enabling students befriend the technology and visualize the abstract program, in order to develop the solution. In this method, a theoretical approach is firstly made of each topic to be learned and certain mental structures are specified. It is proposed that if a student constructs them, then he/she will be able to learn the concept in question. A mix of Action Learning and Lonergan's 'program for life' led to the development of this strategy in the context of computing education.**

Index Terms – Computer Programming, Insight, Understanding.

INTRODUCTION

It is believed that optimal learning occurs when students are given the opportunity to explore, in a guided environment. A major objective of this study is to find a solution to help those students who have difficultly in problem solving aspects of computing courses. Some students lack a strong abstract reasoning ability and are unable to fully comprehend the concepts involved in programming. However, when they actively construct things in the physical world, it helps build knowledge in their minds. This knowledge will then enable them to create even more sophisticated solutions, yielding more skills, more knowledge and solve more challenges in a self-reinforcing cycle.

This is a longitudinal study, undertaken in Dundalk Institute of Technology, in conjunction with the University of Limerick. The theoretical framework behind the Program is based on Bernard Lonergan's theory of knowing [1]. Bernard Lonergan, born in 1904, is recognized as one of the most significant philosophic thinkers of this century. Lonergan clearly demonstrates in his book '*Insight: A Study of Human Understanding*' [2] that the same process of understanding that applies to insights in the sciences also applies to metaphysical and epistemological questions.

Insight is that mental (or inward) activity by which the mind grasps the intelligible connections between things that previously had appeared merely disparate. This act of understanding 'sees' a pattern in data where 'seeing' is only a metaphor from physical sight. Sometimes referred to as the 'Ahh!' experience, it is often experienced as a sudden breakthrough. In insight the mind is obviously at a high pitch of creative activity; yet there is another way in which, it was noted, insight is often experienced with certain passivity: 'it struck me' 'it dawned on me' are common phrases. That is perhaps because insight is once deliberately intended, and yet beyond specific control. Memory is seen to play an important role in many insights: sometimes the insight grasps the relation between present circumstances and previously garnered information; sometimes the present insight is sparked by the memory of a past insight. Trial and error are also seen to figure in many insights.

As a Jesuit and teacher, he spent most of his adult life working on a theory of knowledge that covers every area of human understanding and to discover the nature and quality of the process of insight. What sets Father Lonergan's work apart from most theology-oriented philosophers is that he shares modern philosophy's concern about the uniqueness of each man, however Lonergan believed that the process by which adults come to know and design is the same for all normal adults. Lonergan's approach rested on the fact that all human beings have minds and our minds function according to the readily recognizable pattern, not only is the process the same, the individual learners activates and employs it without direction. Lonergan felt that this pattern could become a basis

[1] Cornelia Connolly, Dept. of Mathematics and Statistics, University of Limerick, Limerick, Ireland. cornelia.connolly@ul.ie
[2] Eamonn Murphy, Dept. of Mathematics and Statistics, University of Limerick, Limerick, Ireland. eamonn.murphy@ul.ie
[3] Sarah Moore, Centre for Teaching and Learning, University of Limerick, Limerick, Ireland. sarah.moore@ul.ie

for some unity in our approach to the diversity of fields and systems with which we are confronted. Consequently, the cognitional structure is invariant, in that is remains the same for each student and it is naturally innate because it happens without direction or effort on the part of the knower. This aspect of his teaching has been found to be very helpful for students, in particular, in developing an understanding to learning and knowing. Lonergan's theory forms the theoretical foundation of the Action Program presented.

LONERGAN'S THEORY OF KNOWING

The preliminary stage of Lonergan's program is the uncovering of the cognitional structure of the person that is the innate, invariant thinking process. The process Lonergan defines leads to an improvement in problem solving and understanding of students [2]. In *Insight* Lonergan's primary focus is not the known, but the knowing, and he intends for the reader to begin to pay attention to his or her own knowing. Knowing, according to Lonergan, has a compound structure. First, insight is always insight into sensible or imaginable presentations. Thus there is a distinction between experience and insight. Experience occurs on one level, understanding on another. Second, insights occur spontaneously, but it remains to determine whether they are correct of not. Thus there is a distinction between insight and judgment or reflective understanding. Judgment takes place on a third level, separated from but related to both experience and understanding. This three-fold structure of experience, understanding and judging constitutes knowing, and in *Insight* Lonergan invites the reader to concretely identify the three-fold structure in his/her own mind. The three-fold structure can be represented and explained by the following three levels of knowing:

1. Experience	➡	1. Common Sense Knowing
2. Understanding		2. Scientific Knowing
3. Judging		3. Critical Knowing

Common Sense Knowing, because in the concrete world this occurs spontaneously and does not require the engagement of the problem-solving process, i.e. it is the experience one encounters. Insight is very much part of common sense knowing, and common sense is to be aware of insight. It does not, of course, attempt to define it, and its theoretical interests in the subject are strictly limited; yet it possesses a shrewd grasp of what it is to understand or to fail to understand readily and to be slow to get the point. There is even a limited awareness that insight functions in different patterns, and that common sense has no monopoly on modes of understanding.

Scientific Knowing is employed when an individual engages in a novel situation and the mental processes outlined in the cognitional structure move from concrete to abstract, and thus generates an understanding. Science moves, often with bewildering rapidity, from the familiar world of sights, sounds, and tactile impressions into a highly complex world of theoretical constructs and mathematical formulas, however scientific knowing does not exclusively inhabit the abstract world.

Lonergan made the argument that the same intellectual capacities that operate in science also operate in the life of the subject and world of common sense. But there is a difference, in that in science-world, the human has to recede, however in common sense it can be a dominant factor. Common sense knowing accelerates from insight to decision, while scientific knowing spends much more time questioning in the reflective phase of the cognitional structure.

The third level of knowing, Critical Knowing enables the student not only to solve problems during the course of an educational program, but it also provides the means by which they can engage with confidence in new situations, which are recurrently presented in their ever-evolving life. Critical knowing or metacognitive knowing, is concerned with knowing what we are doing without cognitional structure when we are generating understanding with common sense or scientific knowing. Insights are preceded by questions, and questions are therefore crucial to having insights. This type of knowing is regarded as problem solving in the indirect mode. When learners come to understand and know their cognitional structure and have the ability to manage their critical/metacognitive knowing, they will have a fixed base from which they can learn all there is to be known.

The Action Program described follows these three types of knowing in the development of a strategy to assist students in the improvement of learning computer programming.

THE ACTION PROGRAM

All DkIT first year computing students were asked to complete an initial questionnaire on computer anxiety - control, self-concept and goal orientation, and a random sample were then contacted to participate in the 10-week Action Program. In the first week, students completed an abstract reasoning and numerical ability assessment. Throughout the program one envisaged making 'the problem' more concrete and creating a conceptual framework for students to adopt in their programming. By incorporating a sequence of structured exercises using physical objects and virtual icons, one helped overcome the mental 'block' that was deterring them from understanding and learning. The strategy essentially consists of three stages, as depicted in Figure 1, based on Lonergan's three levels of knowing.

In the First Stage of the course, the focus was on teaching the students the most basic elements of programming, by using LEGO MINDSTORMS©. Students learn to construct program fragments, and are taught how to connect a set of instructions to create a program. Students see at first hand the physical, or practical implementation of their code. In the Second Stage the students are introduced to LOGO©, and again students are introduced to the virtual / abstract thinking. However in this stage they no longer see their code resulting in the actions of a physical object, but as an icon on the screen.

Students observe the behavior of the turtle to the set of instructions they execute. In Stage Three, one addresses the issue of how the sequence of instructions and the correct syntax and engagement operate together, as before. These characteristics, combined with logical, abstract thinking and sequencing is the next phase of the program. Students have progressed from programming a physical robot, to a virtual icon to finally tackling their 'classroom' code. Throughout the three stages we are focusing on the students abstract thinking and evolving it through the physical, virtual icon and finally the classroom code.

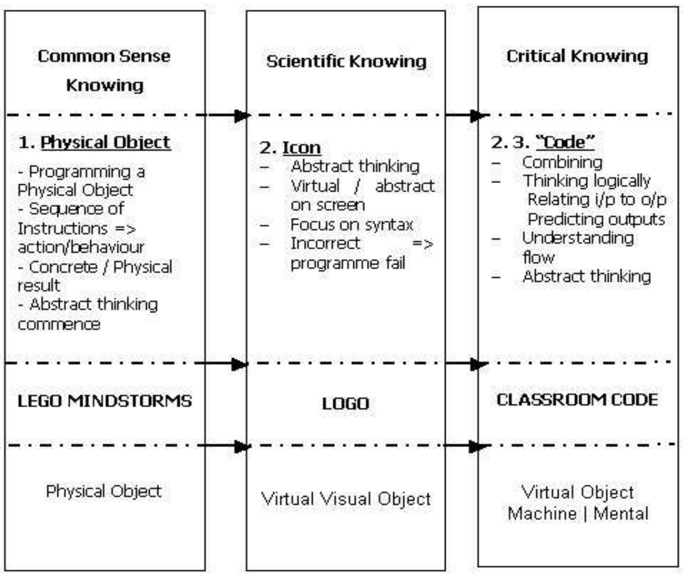

FIGURE 1
OVERVIEW OF ACTION PROGRAM STRATEGY

Stage 1 – Physical Object and Commonsense Knowing

In the first weeks of the course, the focus is in teaching the students the most basic elements of programming, by using LEGO MINDSTORMS. During this stage students learn how to connect a set of instructions to create a program. Students observe variation in overall behavior as they modify their code and as their code interacts with various other entities. At the same time, students observe the relationships between the code they write and the behavior of the system as a whole.

In using LEGO MINDSTORMS students start by building machines out of LEGO pieces, using not only the traditional LEGO building bricks but also newer pieces such as gears, monitors and sensors. In the Action Program the basic structure/body of the robot is pre-built for students, as a timesaving initiative, but students build the wheels and complete the robot. LEGO MINDSTORMS enables students to build behaviors into their robots; therefore they see at first hand the physical/practical implementation of their code. They observe variation in overall behavior of the robot when they execute their code.

Educational technologies, when applied effectively, allow students to partake in hands on learning and comprehensive experimentation. In order to inspire students to learn with technology, teachers must apply the theory of constructivism to their learning [3][4]. According to the basic constructivist theory, students learn the most when they are given the opportunity to explore and create knowledge that is of personal interest to them [5]. Students should be given the chance to work with hands-on projects that they are interested in, and to explore and test their ideas. This style of learning encourages students to create tools and environments that sustain projects that are meaningful to them on a personal level. In building and programming the robots, students develop new ways of thinking about computation, programming and control. No longer do they see computers as boxes that sit on the desks, controlling images and applications without their knowledge as to how or why. No longer do they see programming as intimidating or as something for experts only. Students play the role of computer scientists and electrical engineers and by constructing machines with behaviors, and students develop a new perception of programming.

At this stage the insight, knowing and understanding is very much part of common-sense knowing, and common-sense is unaware of the insight. Through the use of the LEGO MINDSTORMS the theoretical aspects in the programming are limited; yet the student possesses a perceptive notion of understanding the application or failure to.

Stage 2 – Virtual Object and Scientific Knowing

In the second stage students are introduced to LOGO. In LOGO students observe the behavior of the turtle to the set of instructions they execute. Students encounter the rules of syntax and engagement, which they must adhere to, in order for their program to be successful. Research has shown that students who work at a syntactical level are not necessarily provoked to think about the processes involved in their work. However the immediate feedback in LOGO from the screen can sometimes encourage students to work in this way – particularly when the computer laboratory environment is competitive and product oriented. This stage also introduces peer programming.

Excellent insights into what cybernetic thinking is, comes from contemplating various flaws in the logic. The student will learn that a program will work if, and only if, everything goes exactly according to plan. It has no margin for error. It will fail if the turtle turns a tad too soon or a tad too much. Granada LOGO© is accommodating of pupils of all abilities in programming. Reference [6] describes an application as "highly interactive" if the student and the computer control the flow of it together. With LOGO, the user's commands are instant, appreciative of the fact that students tend to want a high level of interaction [7]. The software can be described as proactive and induces cognitive interplay. Reference [6] contends that good computer-aided instruction considers both the subject area and the human learning processes involved.

LOGO is undoubtedly constructivist in its design. In addition, research claims the user interface should be carefully designed to be effective, a feature LOGO also delivers with its efficient layout [6].

In this stage of the Action Program, students are encouraged to work together in teams and to some extent monitor their own learning and progress. Allowing students to explore and interact with projects in pairs, is giving them the opportunity for discovery. In testing out their thoughts and designs, students will develop notions that they have never thought about before. Created by the students, and relating to a meaningful subject, these notions, or powerful ideas, allow the student to see *how* and *why* something works [8], thus they will not only have a better understanding of the information, but he/she will also have the skills to apply the concept elsewhere. In this context, because the student develops the idea/program through his/her own experimentation, he/she experiences a connection to the idea, and a positive outlook on learning [9].

In relating this stage to Lonergan's theoretical approach, the move from common sense to science is a move from the subjective to the objective, a move from the physical to the virtual. Scientific knowing seeks to replace various subjective meanings of words/actions with one commonly agreed-upon definition. It is a move from the personal to the impersonal. Ultimately, the knower tends to become the instrument of a scientific project leader larger than him/herself, rather than intellect being a tool of his more egotistical needs. Definition seeks an expression of the universal case, and scientific laws aim always toward larger and larger generality. The qualitative different between commonsense and scientific knowing is illustrated in the spontaneous, personal descriptions of commonsense and the rigorous, systematic explanations of science. Commonsense knowing accelerates from insight to decision, while scientific knowing spends much more time questioning in the reflective phase of the cognitional structure.

Stage 3 – Classroom Code and Critical Knowing

The task of a specializing programming environment for novices begins with the recognition that programming is a hard skill to learn. The lack of student programming even after a year of undergraduate studies in computer science was noted and measured in 1982 [10] and again in this decade [11]. By introducing the students to their 'classroom' coding environment, and actively working with the students on the topics and components learnt to date, students understanding and train of thought improves.

When pupils work on well defined abstract goals, they are more likely to take on a global perspective and plan their work in a way that more naturally suggests the idea of breaking a problem into parts and defining each part as a separate sub-procedure. This is important and is addressed in the Action Program; the tasks are designed whereby students will perceive the functionality and power of the idea of sub-procedures.

This stage combines the skills and 'best practices' addressed in the previous stages - how the sequence of instructions and the correct syntax and engagement operate together, combined with logical and abstract thinking. Students have progressed from programming a physical robot, to a virtual icon to finally tackling their 'classroom code'. It is imperative at this point that students have learnt how to and fully deconstruct the initial problem, that they view the problem as a series of inputs and outputs, and step through the problem logically. Managing one's cognitional structure and critical knowing at this stage will enable learners to learn and solve problems more effectively because they now have affirmed the method by which they come to know: they have learned how they learn and proactively participated in the process.

DISCUSSIONS AND CONCLUSIONS

It is generally believed that when students are placed in a supportive learning environment they will enter the learning cycle, confident that they will be able to solve any challenge. A supportive learning environment is one that allows the student to construct his or her own knowledge with a certain framework. This means that the optimal learning environment should allow for choice and diversity in a congenial atmosphere. It is the emotionally rewarding state of being optimally challenged, or being in a state of flow, that motivates us all to learn and this is where learning is richest. If the student is insufficiently challenged, they will learn nothing. By sending a signal of boredom, the body will tell itself that they are wasting their time and thus not learning effectively enough. The student will instinctively try to do something about it and move on. One will seek a greater challenge to match the skills in order to get into a state of flow. On the other hand, if the student is challenged way beyond their abilities, they may give up before they begin. For the student who finds him or herself in a state of anxiety, they may not even try to solve the problem. Again, they learn next to nothing, as most of their mental energy is spent on feeling defensive. In this situation, the best thing that can happen is that they either quickly adopt new skills or lower the challenge so that they do no become in a state of feeling inferior.

The type of knowing engaged by an individual, whether commonsense, scientific or critical, is determined by the situation and the individual's need. There may be an imperceptible shift from commonsense to science as one engages a problem. However, when the individual engages in effortful, problem-solving thinking commonsense knowing is of little value – scientific knowing is required in these situations. Students who create their own solutions to problems experience the thrill of genuine achievement – an experience altogether different from simply memorizing the achievements of others. And it is important to stress that learners need to feel that they are in control [12]. Extrinsic motivation is less likely to lead to effective learning than intrinsic motivation (reward from the activity itself).

Throughout the three stages of the Action Program we are empowering students to feel in control of both of themselves, their programming and the technology they are using, in order to enjoy their classes and begin to take charge of their own learning process. It is hoped that through the appropriate use of educational techniques and technologies we can help make our students aware of, and reflect upon, their own thought and learning processes.

REFERENCES

[1] Colleran, C. N., "Improving Adults' Quantitative Problem Solving Skills: Theory, Context and Program Design." Dept. of Mathematics and Statistics, University of Limerick, Doctor of Philosophy, 2002

[2] Lonergan, B., *Insight: A Study of Human Understanding*. New York, Philosophical Library, 1957.

[3] Piaget, J., *Origins Of Intelligence in Children*, New York, International Universities Press, 1952.

[4] Piaget, J., *Language And Thought Of The Child*, New York, Humanities Press, 1959.

[5] Papert, S., *The Children's Machine*, New York: Basic Books, 1993.

[6] Kuittinen, M., Criteria for Evaluating CAI Applications, *Computers and Education*, Vol. 31, pp.1-16, 1998.

[7] Morgan, M., "Selection, Implementation and Development of Hypermedia Resources for the Classroom," University of Limerick, 2002.

[8] Papert, S., *Mindstorms: Children, Computers, and Powerful Ideas*, New York: Basic Books, 1980.

[9] Papert, S., What's The Big Idea? Towards A Pedagogy Of Idea Power. *IBM Systems Journal*, Vol 39 (Nos3-4), 2000.

[10] Soloway, E., Ehrlich, K., Bonar, J., Greenspan, J., "What do Novices know about Programming?" In Andre Badre and Ben Schneiderman, (eds) *Directions in Human –Computer Interaction*, pp 87-122. Ablex Publishing, Norwood, NJ, 1982.

[11] McCracken, M., Almstrum, V., Diaz, D., Guzdial, M., Hagan, D., Kilokant, Y., Laxer, C., Thomas, L., Utting, I. and Wilusz, T., A Multi-National, Multi-Institutional Study Of Assessment Of Programming Skills For First Year CS Students. *ACM SIGCSE Bulletin*, Vol.33 No.4, pp125-140, 2001.

[12] Donaldson, M., *Children's Minds*, Fontana, London, 1978.