IDEA GROUP PUBLISHING
701 E. Chocolate Avenue, Hershey PA 17033, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com
ITP4456

Constructionist Principles as an Aid for Teaching: Notes from Experience

Carlo Gabriel Porto Bellini
Postgraduate Program on Administration – PPGA, School of Administration – EA,
Federal University of Rio Grande do Sul – UFRGS
cgpbellini@adm.ufrgs.br, Av. Ipiranga, 3377 / 901, 90610-001, Porto Alegre, RS, Brazil

ABSTRACT

This paper presents an application of constructionist principles and the programming language LOGO (Papert, 1994) as an aid for teaching Logic for Computer Science. The work reflects observations made by the author during a two-year experience as an educator in the Data Processing curriculum at the Young Men's Christian Association (YMCA) in Porto Alegre, South Brazil. During that period, it was possible to cross data from "traditional" class meetings and from meetings supported by LOGO and a set of constructionist assumptions. From the experience, perceived benefits and difficulties faced by both tutor and pupils are presented, as well as comments on applying constructionist principles and the programming language LOGO to class meetings.

The general conclusion is that Constructionism is actually very appropriate to foster the individual commitment to building the individual and the collective knowledge in a group of students. Some important benefits observed in the study were (1) the students' increased motivation to participate in the class meetings, (2) their improved depth of analysis of problems, and (3) the unique hands-on experience provided by this approach to teaching and learning. Benefits are supposed to be related to the approximation of examples and exercises to the reality of each individual student. On the other hand, it is worth of note that applying constructionist ideas to class meetings requires teachers to be well-prepared in fundamentals of this approach, as well as students to be open to a new class format (not passive, but proactive). The programming language LOGO is another factor of influence in meetings, as it requires instructors and students to have at least a minimum knowledge about Math and computer operation.

INTRODUCTION

Papert (1994) suggests the use of the programming language LOGO in learning activities. The underlying idea is that people learn more effectively when they get involved with the topics under study, for instance by preparing their own personally-meaningful products for constructing knowledge (Bruckman and Resnick, 1995). An additional hypothesis is that this would hold true for all areas of study and for all educational levels – although LOGO was originally designed for children. This relates to computing increasingly being taught across the curriculum instead of as a subject in its own right (Latham, 1998, p. 268), and to the technology of education being transdisciplinary (Bertrand, 1991, p. 82) – what is also accordant to a supposition of Gaines (apud Bertrand, 1991, p. 95) stating that the tools for acquiring knowledge should be independent of a particular dominion. For Papert, LOGO is a very powerful tool to address these ideas.

Papert names his research line "Constructionism", which is derived from Piaget's "Constructivism". Constructionism is part of a broader area of studies on Education known as the "Technological Theories" (Bertrand, 1991), whose argument is that logically organizing the concrete means to support teaching would help improve the pedagogical communication and facilitate the apprenticeship (Bertrand, 1991, p. 81). The information technology (IT) – the hardware and the software supporting the information systems (Alter, 1996, p. 2) – may be valuable in this process (Tickton apud Bertrand, 1991, p. 80), when appropriately designed for educational purposes (Leidner and Jarvenpaa, 1995), since it requires a fundamental change in cognitive arrangements (Levacov, 2000, p. 282). Therefore, schools need to update staff skills, and address IT use and teaching (Latham, 1998, p. 268).

The aim of this article is, from the author's experience with adopting a set of constructionist assumptions and the programming language LOGO for introducing Logic for Computer Science to students, to present perceived results from the application and to suggest topics for future researches. In the first section, Logic for Computer Science is presented, along with difficulties identified in its study; the second section proposes applying the programming language LOGO to cope with these questions in a prepared setting; subsequently, it is shown how the idea was put into practice, and important insights from the experience are provided; finally, comments are made on exploring new forms of instruction through the use of the programming language LOGO as inserted in the constructionist approach to Education.

LOGIC FOR COMPUTER SCIENCE

According to Salmon (1987, p. 23), Logic relates to the study of arguments, entities of Linguistics (p. 22) representing statements related to one another (p. 15). Since computer systems are based on algorithms to solve problems, and as long as algorithms are built over sentences that follow grammar rules (for appreciating how grammar rules give sentences a meaning, refer to Dauer, 1996, p. 346), it is possible to say that there is a strong relation between arguments and algorithms. So, approaching Logic is fundamental to understand the computer systems.

After two years (from 1994 to 1996) teaching Logic for Computer Science for the Data Processing curriculum at the Young Men's Christian Association (YMCA) in Porto Alegre, South Brazil, this paper's author was given the challenge of helping students improve their learning processes. Although they had a good general performance in the studies, it was evident that hardly the ideas dealt with in class were applied to their daily activities, sometimes even not to the problems involved in programming computers (the utmost objective of the course). The

This paper appears in *Managing Information Technology in a Global Economy*, the proceedings of the Information Resources Management Association International Conference.

most critical aspects to be addressed seemed to be the following:

1. How to link class themes to the students' reality, in order for them to remember the theory when in touch with a corresponding situation? After all, we learn better when practicing (Carroll apud Bertrand, 1991, p. 97).
2. How to make the students experience a problem's solution without compelling them to simply accept the tutor's explanation? This was a critical problem in some particular occasions when students were uneasy to forgo their intuition in favor of a valid logical solution presented to them.
3. How to make use of computers during the class meetings, allowing the students to since early associate Logic to practical issues of the Computer Science? And how to succeed using computers when their experience with IT was in its first steps?
4. How to deepen the study of Logic in a friendlier fashion?

OUTLINING A SOLUTION

It was supposed that the aforementioned questions could be handled if the programming language LOGO – and associate constructionist principles – was used in class meetings. The reason was that LOGO was developed to serve as a tool for improving the knowledge-building process, whatever the area of studies was and wherever people were in their personal development continuum. Moreover, LOGO is relatively easy to operate by people with very different backgrounds (what is in line with the motivation people find in *using* the software, not in *learning* it – Bertrand, 1991, p. 97), and that was the case of the group under analysis: 18- to 40-year-old people from diverse professional areas. Additionally, LOGO interpreters usually have a friendly visual interface – characterized by the image of a turtle representing the cursor's position –, what makes its use more appealing.

The four questions raised would be approached in the following manner:

1. To approximate Logic to each student's reality, exercises would be taken as close as possible to daily life, as well as each student would be given the opportunity to suggest themes for study. This way, knowledge would be applied to a variety of well-known contexts, and consequences could be freely approached (Gaines apud Bertrand, 1991, p. 96).
2. To set the students free from the tutor's particular vision of reality, computers would be used to make them test their own reasoning, as well as the tutor could use the computer to "prove" a solution given to a particular problem.
3. To make use of computers since early in the course, the programming language LOGO would be the way to do it. To introduce the students to the activity of programming, LOGO promised to be an easy-to-use language.
4. To deepen the study of Logic in a smooth way, it would be relied on the visual components of the LOGO interpreter and on the fragmentation of problems into pieces of reasoning. In this sense, students would be asked to decompose problems into smaller ones, and each minor problem would be then codified into a LOGO source-code; after all the problems were codified and tested, the source-codes would be integrated in an algorithm representing the original complex problem. So, students would learn to solve problems of Logic in parts and would do it in a pleasant way, as provided by the graphical interface.

THE TURTLE'S WALK

The proposed activities were applied to 20 students during two months, and results were compared to a two-year experience in traditional (*objectivist* – Leidner and Jarvenpaa, 1995) classes. Unfortunately not all the objectives could be completely attained by the end of the experiment, because it was hardly possible to operate computers during the class meetings, since the school had trouble allocating IT resources to the several parallel disciplines requesting them (schools are businesses where IT is probably in its infancy – Latham, 1998, p. 267). The implication was that students were asked to apply independently the ideas dealt with at class to computers they had access to outside school.

Nevertheless, the experience with LOGO (although mostly conceptual) made clear some points not fully realized prior to this experiment. One of them was that students had basic deficiencies regarding Math, mainly Geometry. Due to LOGO relying heavily on Geometry (to build meaningful objects on the computer screen), exercises were not easily carried out with that group of students. This fact was not supposed to occur, since all students had the necessary background. Therefore, although LOGO is capable of being adapted to any field of studies, some fundamental assumptions should be worked prior to its introduction to people (an IT strategy is necessary for supporting school objectives – Latham, 1998, p. 268).

Another crucial aspect was the difficulty implied in applying LOGO to some specific problems. As some ideas require algorithms far from trivial to be implemented, tutors must carefully observe the computer programs generated by students, not letting them get discouraged with LOGO if trying to build complex computer programs. Programming must adhere to the subject being taught and be relatively free from external influences (free use of technology, where applicable, is necessary for developing cognition – Griffith and Northcraft, 1996, p. 101), but also the exercises must fit the programming activity.

Some questions raised by the literature – such as the teacher loosing control over the meetings (Bertrand, 1991, p. 104) – could not be verified, since they require extensive use of computers *during* the class meetings (what was not completely practicable, as mentioned). Such questions are planned to be addressed by a study to be carried out in 2001 with undergraduate students. On the other hand, some issues treated by the literature could be fully appreciated, such as the increased satisfaction of the students with the learning process (Papert, 1994). Also, it was clear that the students attended the meetings more proactively, they deepened the analysis of problems, and the hands-on experience provided by this approach to teaching and learning (when made real – due to the problems for allocating computers) was felt by both tutor and pupils as more agreeable than the traditional class format.

CONCLUSIONS

Although it was not possible to test all the hypotheses, mainly due to problems for using computers during the class meetings, some important questions were raised regarding the introduction of IT (as represented by LOGO) to a group of students. Basically, it was observed that both tutor and pupils must be prepared in fundamentals of this technology, for them to feel the benefits it offers for teaching and learning.

After this first experience with reshaping class meetings through the use of some constructionist principles and the programming language LOGO, the Data Processing curriculum at the YMCA in Porto Alegre was reformulated to face market demands, so it was not possible to keep going with the experiments. But another research similar to this is to be carried out in the next months with undergraduate students, and it will embody the suggestions for research made by the present study.

REFERENCES

ALTER, Steven. **Information Systems: A Management Perspective**. Menlo Park, USA: Benjamin & Cummings, 1996.

BERTRAND, Yves. **Teorias Contemporâneas da Educação**. Lisbon, Portugal: Instituto Jean Piaget, 1991.

BRUCKMAN, Amy, RESNICK, Mitchel. The MediaMOO Project:Constructionism and Professional Community. **Convergence**, v. 1, n. 1. Luton, UK: University of Luton Press, Spring 1995. Available at: http://www.cc.gatech.edu/fac/Amy.Bruckman/papers/convergence.html.

DAUER, Francis Watanabe. **Critical Thinking – An Introduction to Reasoning**. New York, USA: Barnes & Noble, 1996, 542 p.

GRIFFITH, Terri L., NORTHCRAFT, Gregory B. Cognitive Elements in the Implementation of New Technology: Can Less Information Provide More Benefits? **MIS Quarterly**. Minneapolis, USA: MIS Research Center, University of Minnesota, March 1996, pp. 99-110.

LATHAM, Ann. Strategic Information Systems Planning: A Necessary Evil for Schools? **Journal of Applied Management Studies**, v. 7, n. 2. Telford, UK: Carfax Publishing Ltd. & Wolverhampton Business School, 1998, pp. 267-274.

LEIDNER, Dorothy E., JARVENPAA, Sirkka L. The Use of Information Technology to Enhance Management School Education: A Theoretical View. **MIS Quarterly**, v. 19, n. 3. Minneapolis, USA: MIS Research Center, University of Minnesota, September 1995, pp. 265-291.

LEVACOV, Marília. Bibliotecas Virtuais. In.: MARTINS, Francisco Menezes, DA SILVA, Juremir Machado. **Para Navegar no Século XXI – Tecnologias do Imaginário e Cibercultura**. Porto Alegre, Brazil: Editora Sulina/Edipucrs, 2000, pp. 261-286.

PAPERT, Seymour. **The Children's Machine: Rethinking School in the Age of the Computer**. Basic Books, May 1994, 256 p.

SALMON, Wesley C. **Lógica**. Rio de Janeiro, Brazil: Editora Guanabara S. A., 1987, 143 p.

Related Content