Interactive Educational Multimedia, Number 12 (April 2006), pp. 1-7
http://www.ub.edu/multimedia/iem

Adaptive Tutorial's Constructivist Basis for the Teaching-Learning Process of an OOPL

Gabriela Aguilar-Burguete
Apartado 37 Cintalapa Chiapas Mexico CP 30400
eublan80@yahoo.com

Kenji Kaijiri
Shinshu University
Faculty of Engineering, Information Engineering Wakasato 4-17-1 Nagano-shi, Nagano-ken 380-8553 Japan
kaijiri@cs.shinshu-u.ac.jp

Summary

This work proposes an adaptive tutorial's constructivist basis for the teaching-learning process of an OOPL. The teaching-learning process model is based on the combination of Constructivism, Bloom's Taxonomy and learning strategies in an iterative process. The objective of the incorporation of the Bloom's Taxonomy is to measure how well the learning goals are met. Furthermore, the objective of incorporating the learning strategies is to provide an easy and effective framework for mastery an OOPL. The evaluations given after completing each learning strategy make possible to personalize the teaching-learning process to meet the learner's needs.

Keywords: Constructivism, Bloom's Taxonomy, Learning Strategies.

Introduction

"Is it possible a constructivist environment is more compatible with object-oriented languages? Further research is necessary to explore this dimension of computer science education" (Gibbs, 2000).
Some works about the teaching-learning process of an OOPL have been previously developed, such as those of (Gibbs, 2000; Hadjerrouit, 1998; Hadjerrouit, 1999; Lester, 1997; Turk, 1997). These works propose teaching-learning process of an OOPL based on the Constructivist paradigm. However, they do not give any guidelines about how to implement those approaches in Adaptive Web-based Tutorials Systems. In this work the teaching-learning process of an OOPL based on the Constructivist paradigm is designed for use in an Adaptive Web-based Tutorial System.

The proposed approach is based on the combination of the Constructivist paradigms (Cognitive and Social Constructivism), Bloom's Taxonomy and learning strategies in an iterative process. The objective of incorporating the Bloom's Taxonomy is to measure how well the learning goals are met. Furthermore, the objective of incorporating the learning strategies: modeling, problem solving and project development is to provide an easy and effective framework for mastery an OOPL. At the same time, the evaluations given after completing each learning strategy will make possible the adaptation of the teaching-learning process to meet the learner's needs.

This paper is organized as follows: The Constructivist basis for the teaching-learning process is analyzed first then; the Bloom's Taxonomy is introduced. Then, the learning strategies are given. The teaching-learning process of an OOPL is posit next. After that, is given the analysis of the effectiveness of the proposed teaching-learning process. Conclusions are given in the last Section of the present work.

The Teaching-learning process based on the Constructivist Paradigm

In this Section a teaching-learning process model based on the Constructivist paradigm is proposed. This novel teaching-learning process consists of the combination of Cognitive and Social Constructivist Paradigms.

Social Constructivism proposes that the learner acquires from the environment (social context) the knowledge that he/she will learn. But the environment also teaches how to learn by giving the learner pre-established mental schemas that he/she simply adopts and uses, so the learner can adopt and learn beliefs, attitudes, and knowledge from the environment. The environment or social context consists of teachers, parents, friends, books, etcetera. The learners are social individuals, thus, the learning process occurs in a social context; but Cognitive Constructivism proposes that it is up to the learner to experience the learning process through assimilation, accommodation and adaptation, because each learner has his/her own learning styles and constructs his/her own realities while experiencing life in his/her own.

We posit that with the learners' own understanding about a topic they can apply the new knowledge to the generation of creative ideas. After the learners have adopted their own understanding as their own knowledge, then they can apply the adopted knowledge interacting with the environment. And from here, they can generate new knowledge from the application of the adopted knowledge. After this process, they can verify whether or not their new knowledge is correct with the environment.

A learner acquires from the environment (social context) the knowledge that he/she will learn; in this case, the OOPL is the knowledge to be learnt. The proposed tutorial system will help the learner to establish a relation between background knowledge and new knowledge through the use of background knowledge evaluations, making the assimilation process easier. The development of a program implies a direct experience through the solution of problems. This process helps the learner to create his/her own understanding, that is, helps the learner to undertake the accommodation process. Mental schemas are constructed through assimilation and accommodation. The learner adapts his/her schemas to the environment through problem solving, modeling, and project development learning strategies. These learning strategies will allow the learning process to be experiential in a meaningful context.

As explained above, this Constructivist paradigm provides a method to teach and learn OOPL concepts easily.

Bloom's Taxonomy

Bloom's Taxonomy provides a framework within which programmers' knowledge of some domain can be assessed (Zachary, 2003). The proposed tutorial system will be focused on an introductory course of C#. For this reason the Courseware was designed considering only the first three levels of Bloom's Taxonomy. The first three levels of Bloom's Taxonomy hierarchy are given below:
In general, learners with superior Intelligent Quotients (I.Q.) reach superior levels of the Bloom's Taxonomy hierarchy more easily than learners with a lower I.Q. For example, learners with high I.Q.s remember the Object Oriented Programming (OOP) Concepts, comprehend and apply them without additional instruction. Furthermore, when a learner reaches superior levels of the Bloom's Taxonomy hierarchy, he/she uses more types of intelligences. The same thing happens in relation to higher levels of background knowledge and prior abilities. If the learners have more background knowledge and abilities it will be easier for them to reach superior levels in the Bloom's Taxonomy hierarchy.

Level	Definition
Knowledge	Student recalls or recognizes information, ideas, and principles in the approximate form in which they were learned. The verbs that can be used to assess are: define, describe, identify, list, match, name, and etcetera.
Comprehension	Student translates, comprehends, or interprets information based on prior learning. The verbs that can be used to assess are: convert, defend, distinguish, estimate, explain, generalize, rewrite, etcetera.
Application	Student selects, transfers, and uses data and principles to complete a problem or a task with a minimum of direction. The verbs that can be used to assess are: change, compute, demonstrate, operate, show, use, solve, etcetera.

Table 1: Bloom's Taxonomy classification (Cooper, 2003)

Conversely, learners with lower I.Qs will need some guidance in order to reach all three levels of the Bloom's Taxonomy. For this reason the tutorial must consider not only intelligence types but also background knowledge and abilities, because the tutorial system must be able to keep the attention of the learners through interesting activities appropriate to the personal needs of each learner.

In order to measure how well the learning goals are being met, there will be evaluations after each theme is taught. These evaluations were designed to help each learner reach the knowledge, comprehension and application levels. Each learner must master the knowledge evaluation before studying OOPL in comprehension level. He/she must likewise master the comprehension evaluation before studying OOPL in application level.

Learning Strategies

The learning strategies to be incorporated in the proposed tutorial are: modeling, problem solving and project development (University of Kansas, 2002). Modeling consists of teaching concepts and well-formed programs to learners with the expectation that the learners will learn from the models. Problem Solving consists of solving problems related to the target OOPL. Problems are introduced to the learners with the expectation that the learners will solve the given problems by applying the concepts and syntax learnt through modeling. In project development, learners will implement real and complex problems, using the newly learnt OOPL. The tutorial system will foster the acquisition of knowledge and the development of programming abilities through an iterative process. This iterative process consists of a loop, which runs through the learning strategies: modeling, problem solving and project development. It is expected that, as a result of the modeling process, the learner can write well-formed programs by learning from the models. Each problem to be solved by the learner has related the background knowledge, OOPL exercises, programs as models, etcetera which will help the learner in the implementation of a given problem. Also it is expected that, as a result of the problem solving process, the learners can implement solutions to simple problems. It is expected that the learner can, as result of the project development process, implement complex problems.

Teaching-Learning Process of an OOPL

In this section a teaching-learning process of an OOPL which can be implemented for Adaptive Tutorials is introduced. The proposed teaching-learning process is given in order to illustrate the adaptive process and how this process is related with Bloom's Taxonomy and learning strategies. In order to incorporate the learning strategies, modeling, problem solving and project development in the teaching-learning process of an OOPL, the following learning activities are proposed:

- *A) Modeling: The learners will learn using experts' implemented programs as models. The tutorial will introduce to the learners well-formed programs for reading and completing practice exercises.*
- *B) Problem Solving: The learners will use the problem solving learning strategy by implementing programs from scratch.*
- *C) Project development: The learners will get practice in the project development learning strategy by reading at first some modules of a whole project, then correcting some other modules of the same project and at the end, implementing from scratch the final modules of the project.*

The teaching-learning process was designed using an approach, which allows the evaluation of the acquisition of knowledge and the evaluation of the development of programming abilities. A process represents a stage in the teaching-learning process as outlined below:

1) Knowledge Process: The learner will acquire the knowledge about OOP Concepts.
2) Abilities Process: The learner will apply these concepts in order to develop the ability required to implement programs.

The Knowledge Process uses the Web pages which introduce the OOP Concepts. The knowledge is introduced to the learner based on the first three levels of the Bloom's

Taxonomy and the Learning Strategies:

1) Knowledge level: Concepts are introduced to the learners.
2) Comprehension level: Explanations are introduced to the learners with the objective that the learners comprehend the concepts introduced in the knowledge level.
3) Application Level: The learners will be able to apply the concepts and skills they have mastered.

The modeling learning strategy includes the knowledge, comprehension and application levels. This is because, using the modeling learning strategy, the tutorial will introduce programs as models which the learners can use and adopt in order to understand and apply the learnt concepts. Problem solving and project development include the application level. This is so that the learner will be able to apply the learnt concepts through problem solving.

The Abilities process consists of the following sub-process: Syntax understanding, Program implementation and Project development:

1) Syntax understanding sub-process: The learner will learn the syntax required to write programs.
2) Program implementation sub-process: The learner will practice the implementation of programs in order to acquire the necessary programming ability.
3) Project development sub-process: The learner will apply the knowledge and abilities learnt in order to acquire experience in implementing real problems.

The objective of the teaching-learning process is that the learners acquire knowledge about OOPL concepts and develop programming abilities using an OOPL. The evaluations between learning strategies modeling, problem solving and project development were designed following the Bloom's Taxonomy hierarchy. In order to develop the learners' programming abilities some other learning activities were also designed. These learning activities are syntax understanding, program implementation and project development. Each learning activity matches the modeling, problem solving and project development learning strategies.

If the learner has not acquired the determined mastery level of the knowledge or abilities of the previous process, then the tutorial system will re-introduce those themes which the learner failed to master, and this process will continue until the learner acquires the requisite level of mastery. Only then, the learner will be able to start studying the next process. The results of the assessments will be used to determine the weak points of the learners and determine if those learners can proceed to the next module or session.

There is an adaptation process after each evaluation. Modeling, problem solving and project development will be adapted to each learner's intelligence types, learning styles, and background knowledge. The tutorial will adapt the courseware, the list of problems, exams, tests or quick quizzes to the learner's changing needs.

Effectiveness of the Teaching-learning Process

(Holmboe, 1999) provides the results of a study, which its goal was to outline a cognitive framework for describing development of different types of knowledge in informatics.
The results of the study provided by (Holmboe, 1999) give a basis for the effectiveness of the proposed Constructivist based teaching-learning process model. (Holmboe, 1999) wrote, "A person with theoretical knowledge needs rehearsal of skills and hands on experience to be able to construct his or her own holistic knowledge". The proposed model provides "rehearsal of skills and hands on experience" through an iterative process from modeling to project development. Furthermore, each learning activity was developed in order to reach all of the first three levels of the Bloom's Taxonomy. And the Constructivist basis of the proposed teaching-learning model goes a step further and foster in the learners the generation of creative ideas. In this way, the proposed teaching-learning process for an OOPL will help the learner to "experience the connection between reality, model and implemented program, and thus reach holistic knowledge of object-orientation sooner in their learning process".

Conclusions

An approach has been proposed for the teaching-learning process of an OOPL based on the Constructivist paradigm, which can be implemented in an Adaptive Web-based Tutorial System. The incorporation of the Bloom's Taxonomy allowed for the more effective design of the courseware and the classification of the questions. The questions contained in the evaluations between each learning strategy were divided in three levels knowledge, comprehension and application. The incorporation of the learning strategies, modeling, problem solving and project development, allowed the effective adaptation of the Web-based Tutorial. This adaptation process is an iterative process, whose cycle starts when the learner does not master the required skills for a given session or course. In each iteration, there is an adaptation of the courseware presentation based on the learner's learning styles, also there is an adaptation of the questions and the difficulty of the problems based on the learner's I.Q. The adaptation of the courseware, with regard to the presentation and difficulty of the questions and problems, are based also in the learner's background knowledge and prior programming abilities. A future problem of the present research is a standardization of the I.Q. The objective of this standardization is to determine normal and superior I.Q. in each intelligence type. With the I.Q.s range, the system will be able to personalize the kind of problems required to best meet the needs of each learner.

References

Gibbs, D. (2000) The effect of a constructivist-learning environment for field-dependent/independent students on achievement in introductory computer programming. *ACM SIGSE Bulletin, Proceedings of the thirty-first SIGSE technical symposium on Computer Science Education*, 32(1), 207-211.

Cooper , S., & Dann W., & Pausch R. (2003). Teaching Object-first In Introductory Computer Science. *SIGCSE'03*, 35(1), 350-354.

Gray, J., etl: (1998). A Constructivist Learning Environment Implemented in C#. *Proc. ITiCSE*, Conf. Integrating Technology into Computer Science Education, 30(3), 94-97.

Hadjerrouit S. (1998). A Constructivist Framework for Integrating the Java Paradigm into the Undergraduate Curriculum. *ITiCSE 98*, 30(3), 105-107.

Hadjerrouit S. (1999). A Constructivist Approach to Object-Oriented Design and Programming. *ITiCSE 99*, 31(3), 171-174.

Holmboe, C. (1999). A cognitive framework for knowledge in informatics: the case of object-orientation. *ACM SIGCSE Bulletin, Proceedings of 4th annual SIGCSE/SIGCUE ITiCSE Conference on Innovation and technology in computer science education*, 31(3), 17-20.

Zachary J. & Jensen P. (2003). Exploiting Value-Added Content in an Online Course: Introducing Programming Concepts via HTML and JavaScript. *SIGCSE'03*, 35(1), 396-400.

Lester J. & FitzGerald P. & Stone B. (1997). The Pedagogical Studio: Exploiting Artifact-based Task Models for Constructivist Learning. *International Conference on Intelligent User Interfaces, Proceedings of the 2nd international conference on Intelligent user interfaces.* 155-162

Turk M. (1997). Introducing Object Orientation to Experienced Procedural Programmers. *ACM International Conference Proceeding Series, Proceedings of the second Australasian conference on Computer science education*, 135-140.

University of Kansas (2002), Multiple Intelligence in Education [On-line], Available: http://wizard.hprtec.org