World Journal on Educational Technology: Current Issues

Vol 8, Issue 2, (2016) 82-90

http://sproc.org/ojs/index.php/wjet

An e-Learning environment for algorithmic: Toward an active construction of skills

Abdelghani Babori *, Faculty of Science and Technology, Route de Casablanca km 3.5, Settat BP 577, Morocco.

Hicham Fihri Fassi, National School of Applied Science, Route Sidi Bouzid, Safi BP 63, Morocco and Faculty Of Science and Technology, Route De Casablanca Km 3.5, Settat Bp 577, Morocco.

Abdellah Hariri, Faculty of Science and Technology, Route de Casablanca km 3.5, Settat BP 577, Morocco.

Mustapha Bideq, National School of Applied Science, Route Sidi Bouzid, Safi BP 63, Morocco.

Suggested Citation:

Babori, A., Fassi, H., F., Hariri, A. & Bideq, M. (2016). An e-Learning environment for algorithmic: Toward an active construction of skills. *World Journal on Educational Technology: Current Issues. 8*(2), 82-90.

Received May 14, 2016; revised June 20, 2016; accepted July 16, 2016

Selection and peer review under responsibility of Assoc. Prof. Dr. Fezile Ozdamli, Near East University.

Abstract

Assimilating an algorithmic course is a persistent problem for many undergraduate students. The major problem faced by students is the lack of problem solving ability and flexibility. Therefore, students are generally passive, unmotivated and unable to mobilize all the acquired knowledge (loops, test, variables, etc.) to deal with new encountered problems. Our study is structured around building, step by step, problem solving skills among novice learners. Our approach is based on the use of problem based learning in an e-Learning environment. We begin by establishing a cognitive model which represents knowledge elements, grouped into categories of skills, judged necessary to be appropriated. We then propose a problem built on a concrete situation which aims to actively construct a skill category. We conclude by presenting around the proposed problem a pedagogical scenario for the set of learning activities designed to be incorporated in an E-learning platform.

Keywords: Pedagogy, eLearning, Problem Based learning, Algorithmic

*ADDRESS FOR CORRESPONDENCE: **Abdelghani Babori,** Faculty of Science and Technology, Route de Casablanca km 3.5, Settat BP 577, Morocco. *E-mail address*: a.babori@uhp.ac.ma

1. Introduction

Algorithmic is a branch of computer science, basis of all programming languages that provides a foundation for the appropriation of all programming languages and which is defined as the acquisition of abilities related to the description of steps to follow in resolving a problem. Learning this discipline is considered as an immense challenge for the majority of undergraduate students. Kaasbol (2002) stated that their rate of drop out or failure vary from 25% to 80% worldwide. Demotivation, meaningless knowledge and problems in confronting new algorithmic problems mobilizing all the acquired knowledge in order to describe correctly instructions that lead to a solution, are the major challenges that face learners during traditional courses. The student can understand programming concepts (variables, loops, etc.) but still always unable to mobilize them in an appropriate manner.

2. Related work

It has been a common to notice that the majority of first year programming students find the algorithmic very difficult. Feedbacks like “I can’t understand this”, ”this is impossible“, “I will never learn how to program” (Moström, 2011) show that they become jammed and performed poorly in the learning process.

This stuck relies principally on the following elements:

- Miscomprehension of different concepts often seen as fuzzy and difficult to comprehend, many undergraduate students still lack semantic knowledge of what goes on inside the computer for concepts like variable and statements such as declaration and assignment. Bayman and Mayer (Mayer, 2013) stated that 43% of learners say that the assignment instruction “A=B+1” in BASIC means to write the equation A=B+1 into memory
- An intrinsic cognitive overload due to the information flows to be learnt, students are being forced to learn simultaneously arithmetical calculation knowledge (greatest common divisor, factorial, Inverse Matrix, etc.), algorithmic concepts and the syntax
- An extrinsic cognitive overload due to the manner in which information is presented. Courses are based generally on a brief explanation of the structure of algorithmic elements followed by a series of examples leaving little room for analysis and design phases
- Lack of strategies of decomposing problems into sub problems.

Many of learners adopt strategies to get unstuck. Examples include social interaction with peers and others (Moström, 2011). The traditional teaching methods based generally on individual work do not emphasize this interaction. Research studies on teaching computer programming were conducted to promote learning process. In this regard, several systems have been developed to help users to learn the design of algorithms (scratch, AlgoBox, etc,.) that provide programming closest to the natural language as well as projects Codewitz project as an example which aims to develop web-based visualization of programming concepts (Lahtinen, Ala-Mutka, & Järvinen, 2005) and Elearning environments based on the use of example and experience, namely Allogène and Easyalgo (Benabbou & Hanoune, 2006).

Learning with these systems, tools and environments is generally based on problems that have nothing to do with professional world. The traditional courses taught at classes have little relevance for developing critical thinking, problem solving skills and transversal competencies: communication, group work, etc. To address these mentioned difficulties, we are proceeding with the design of an algorithmic eLearning system based on a problem based learning approach (PBL) in which learners confront tangible problems.

Several disciplines have incorporated the use of problem based learning (software engineering, electric, medicines, network, etc.). Richardson and Delaney (2009) discussed how can problem based learning improves students' understanding of concepts in the software Engineering classroom.

3. The proposed approach

Our contribution consists from a cognitive model which represents knowledge elements in terms of concepts, procedures and principles mobilized in a situation and classified into categories of skills to propose a pedagogical scenario based on problem built on a real situation. The sum of the expected results of learning activities will be combined to form the targeted skill. The architecture below described the general idea of our method:

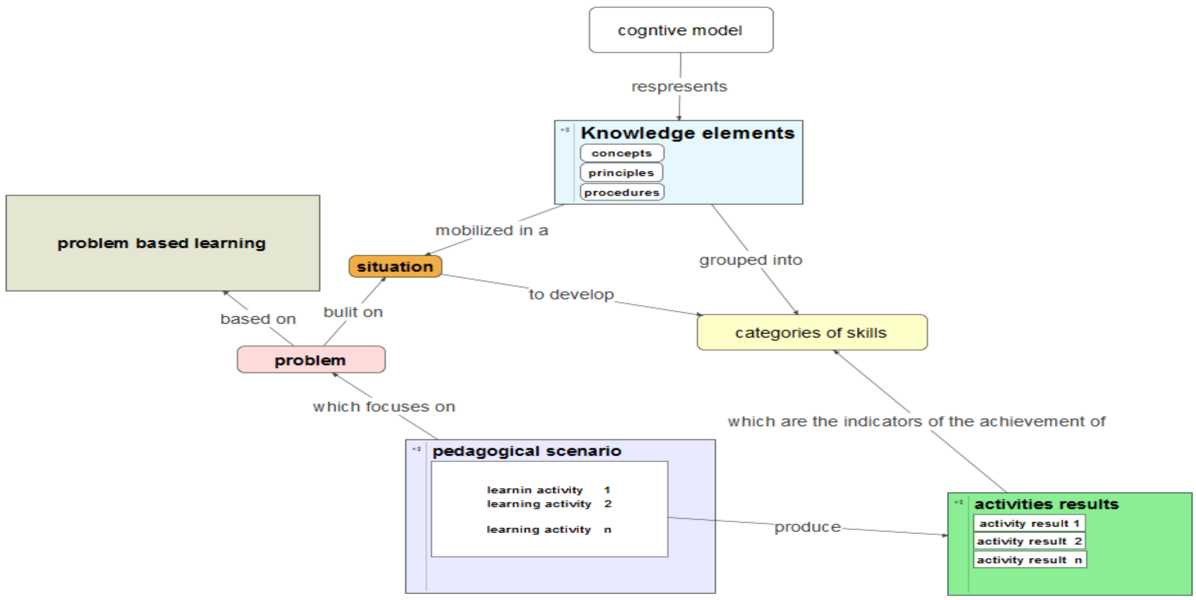

Figure 1. The general idea of the approach

4. Algorithmic eLearning system design

4.1. Cognitive model

To describe the cognitive model of algorithmic teaching, we opted for Object Role Modeling (ORM) meta-model for the following reasons (Jarrar, 2007; Lukichev & Jarrar, 2009):

- ORM allows verbalization of diagrams, this verbalization simplifies communication specifically among non-IT domain experts and modelers
- ORM is a rich meta-model allowing not only to support n-ary relations but also more than fifteen types of constraints graphically
- ORM is the most suitable language allowing modelers to design better systems independently of the logical model unlike UML and entity-relationship modeling model respectively designated for oriented-object programming and databases.

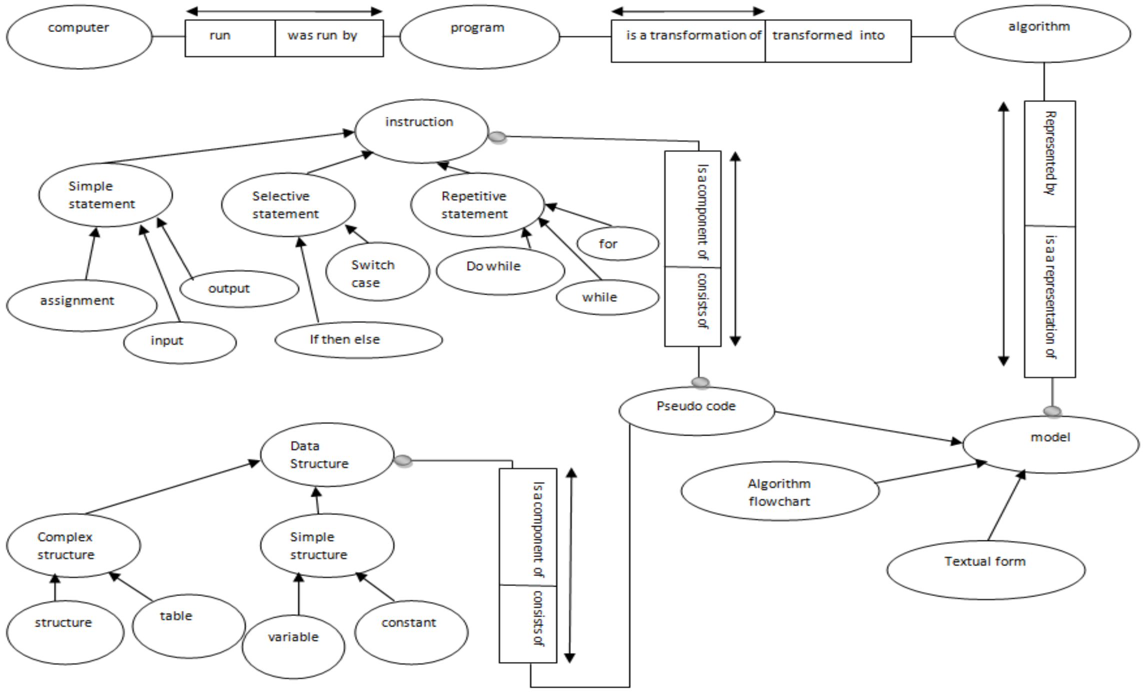

Figure 2. Cognitive model

We proceed by classifying the designated learner's skills into four categories according to the types of instruction and data structure which are part of an algorithm, using the previous cognitive model as shown in the table below:

Treatment / Data structure	Simple treatment	Complex treatment
Simple data structure(const-ant+variable)	**C1**: resolve a problem with a simple treatment and simple data structure	**C3** : resolve a problem with a complex treatment and a simple data structure
Complex data structure (table+structure)	**C2** : resolve a problem with a simple treatment and complex data structure	**C4** : resolve a problem with a complex treatment and a complex data structure

Figure 3. Categories skills

4.2. A proposed problem situation: towards an active construction of competence

In this article we focus on the competence C1: resolve a problem with a simple treatment and simple data structure that we would make novices acquire and which is judged the basis for all the levels. The macro-task presented in this paper is designed in such way to achieve learning objectives identified as being necessary to acquire the competence C1. We propose a concrete problem situation issued from the professional world of engineers concerning the automation of payment system in highways:

The National Company of Moroccan Highways decided to make an automatic payment application. To do this, the company decided to appeal to developers who will have the task of designing this application. After many interviews with the managers of the company, the developers have firstly determined the business processes indicated in the following way:

When we arrive at a toll motorway, we go through a first post entry, where a ticket that you should keep in a safe place is delivered. When we want to leave the motorway, the ticket should be submitted to the exit toll. A Red light will be displayed and the machine shows us the amount to be paid according to the distance covered, Lorries pay more than cars. The rate depends on the type of vehicle. You give your ticket then you put pieces of money into the funnel. A green light will be displayed and the barrier will be raised. It is assumed that the amount to be paid is calculated according to the distance covered and the vehicle type as follows:

- Price= (x+y)*nbkm with x=1DH: fixed cost independent of the vehicle type, y: variable cost dependent of the vehicle type
- nbkm : number of kilometers separating the starting point read on the ticket to be introduced and the supposedly known point of arrival
- $y = \begin{cases} 1\text{DH for class 1} \\ 4\text{DH for class 2} \\ 6\text{DH for class 3} \end{cases}$

It is also important to note that the vehicle class is detected by a sensor (should be linked to the program) that determines the height. The class is determined as follows. The website listing information about that is available here: http://www.adm.co.ma/en/preparez-votre-voyage/pages/tarifs.aspx

- Class 1: Vehicles with 2 axles, whose height (H) is lower than or equal to 1.30 m
- Class 2: Vehicles with 2 axles, whose height is higher than 1 m 30 or vehicles with more than 2 axles whose height is lower than 1 m 30
- Class 3: Vehicles with more than 2 axles, whose height is more than 1 m 30.

Statement of the problem: Design an automatic payment system allowing:

- A display of the message 'introduce the ticket into the reader' as well as the price to be paid
- An entry of the amount to be paid, two cases are presented:

  Case 1: the user introduces an amount greater than or equal to the displayed one

  Case 2: the screen displays that the paid sum is not enough and the user still has to introduce money until the introduced sum is greater than or equal to the amount to be paid
- Print a receipt of payment with a message of goodbye, the paid amount and the refunded amount.

We provide a table including for each phase of the adopted problem based learning tutor interventions in the form of learning activities, expected results formulated based on Bloom's Taxonomy which describe learning objectives and the refereed taxonomy level.

Table 1. Learning activities

Phase	Learning activities in the form of interventions	Expected result	Refereed Bloom's Taxonomy
1. Reading of the problem	✓ Activity 1 : underline key words that you think will solve the problem.	*identify the central elements of the problem.	Knowledge
2. Defining and analyzing the problem	✓ Activity 2 : divide and rank the problem using concept maps.	*divide the problem into sub-problems. *establish the relations between the elements of the presented diagram.	Analysis
	✓ Activity 3 : analyze the input and output elements of the established diagram, deducting elements that characterize an algorithm.	*distinguish the input and output elements.	Analysis
	✓ Activity 4 : *determine what is the computer component where it is possible to store the raised objects? *what do these objects need to be recognized by the program? *explain using a scheme what is a variable declaration from the perspective of machine?	*illustrate using a scheme the concept of variable and its usefulness as well as the semantics of a variable declaration.	Comprehension
	✓ Activity 5 : *what does this expression means : Price= (x+y)*nbkm ? *represent the objects discussing all the possible values of the variable Price. *how to retrieve data that are transferred from the ticket or sensors to the program *what an input instruction brings in more than an assignment instruction. *how to transfer the message as well as the calculated price from the program to the screen?	*illustrate using a scheme the elementary instructions in algorithmic (assignment, input, output). *describe the calculation process of the price(conditional instructions).	Comprehension
	✓ Activity 6 : *how to ensure a dialogue between the machine and the user informing him of the	*discover the usefulness of the repeated treatment to solve this type of problem.	Analysis

	remaining sum to be paid until he introduces the entire amount. *how do we calculate the remaining amount to be paid?		
3&4. Explanation of the problem & assumptions=design of the algorithm	✓ Activity 7: *structure and organize the various explanations issued during the previous phases. *formulate in a textual form an algorithm to solve the problem.	*design an algorithm of problem solving.	Synthesis
5. Self-study	✓ Activity 8 : *determine equivocal elements. *select relevant resources to explain the problem (elements which require a research).	*present a report based on a research on all elements that can explain the problem.	Analysis
6. Applying knowledge	✓ Activity 9: *make a conceptual diagram including the relationships between the newly acquired knowledge.	*create a concept map organizing and synthesizing the acquired knowledge.	Synthesis
7. Reporting phase	✓ Activity 10 : *reject assumptions that are not validated. *assess the achievement of learning objectives as well as the quality of individual or group work.	*judge using criteria (precision, structuration, readability, etc.) the quality of the conceived algorithm.	Evaluation

4.3. Pedagogical scenario

We present a pedagogical scenario, describing the sequentially ordered learning activities. We rely on the model of Pernin and Lejeune (2004) which is simple to be ported to a reusable and interoperable (independent of any delivery platform) scenario in IMS Learning Design. The developed scenario consists of a description of learning situations, specifying the roles, activities and environments required to perform activities.

Table 2. Pedagogical scenario

PBL Phase	Learning activity	role					environment		
		Tutor	Group	Chair person	Learner	Secretary	Tools	Resources	Services
Phase 1	Activity1	✓	✓				------	Web page (for the presentation of the problem)	---------
Phase 2	Activity 2,3,4,5,6	✓	✓	✓		✓	Mind mapping software	----------	---------
Phase 3 & 4	Activity 7	✓	✓	✓		✓	Mind mapping software	---------	A common editing space « **wiki** » + A Survey
Phase 5	Activity 8				✓		----------	Resources proposed by learners	---------
Phase 6	Activity 9	✓	✓	✓		✓	Mind mapping software	---------	Discussion forum
Phase 7	Activity 10	✓	✓	✓		✓	-----------	---------	workshop : self assessment & peer assessment

5. Conclusion

The algorithmic eLearning system designed, based on a problem based learning approach, is intended primarily for the use of undergraduate students. The system highlights a teaching strategy intended to construct step by step algorithmic problem solving skill among novices confronting them with a real world situation. In this article, we limited ourselves to target a problem solving skill, with a simple treatment and simple data structure, which represents a vital basis for all levels. An online Implementation is currently being undertaken planning the pursuit of the research towards a Scenario-based eLearning covering the entire skills categories.

References

Autoroutes du Maroc. (2015). Retrieved from: http://www.adm.co.ma/en/preparez-votre-voyage/pages/tarifs.aspx

Benabbou, F., & Hanoune, M. (2006). EasyAlgo: Un environnement d'apprentissage et d'autoévaluation de l'algorithmique. RIST, 16.

Jarrar, M. (2007). Towards automated reasoning on ORM schemes mapping ORM into the DLR idf description logic. Proceedings of the 26th international conference on Conceptual modeling, 4801, 181-197. Springer-Verlag.

Kaasbol, J. (2002). *Learning Programming*. University of Oslo.

Lahtinen, E., Ala-Mutka, K., & Järvinen, H.-M. (2005). A study of the difficulties of novice programmers. *ACM SIGCSE Bulletin, 37*, 14-18. Finland. doi:10.1145/1067445.1067453

Lukichev, S., & Jarrar, M. (2009). Graphical notations for rule modeling. *Handbook of Research on Emerging Rule-based Languages and Technologies: open solutions and approaches, 1,* 76-98.

Mayer, R. (2013). Teaching for Transfer of Problem-Solving Skills to Computer Programming. *Computer-Based Learning Environments and Problem Solving, 84,* 193.

Moström, J. (2011). A study of student problems in learning to program. Umea, Sweden: University, Department of Computing Science. Retrieved from: http://www.divaportal.org/smash/get/diva2:447104/FULLTEXT02.pdf

Pernin, J.-P., & Lejeune, A. (2004). Dispositifs d'apprentissage instrumentés par les technologies: vers une ingénierie centrée sur les. Technologies de l'Information et de la Connaissance dans l'Enseignement Supérieur et de l'Industrie (pp. 407--414). Université de Technologie de Compiègne.

Richardson, I., & Delaney, Y. (2009). Problem based learning in the software engineering classroom. 22nd Conference on Software Engineering Education and Training (pp. 174-181). Hyderabad, Andhra Pradesh: IEEE Computer Society. Retrieved from: https://ulir.ul.ie/handle/10344/1813