World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:4, No:6, 2010

EASEL: Evaluation of Algorithmic Skills in an Environment Learning

A. Bey, T. Bensebaa, H. Benselem

***Abstract*—**This paper attempts to explore a new method to improve the teaching of algorithmic for beginners. It is well known that algorithmic is a difficult field to teach for teacher and complex to assimilate for learner. These difficulties are due to intrinsic characteristics of this field and to the manner that teachers (the majority) apprehend its bases. However, in a Technology Enhanced Learning environment (TEL), assessment, which is important and indispensable, is the most delicate phase to implement, for all problems that generate (noise…). Our objective registers in the confluence of these two axes. For this purpose, EASEL focused essentially to elaborate an assessment approach of algorithmic competences in a TEL environment. This approach consists in modeling an algorithmic solution according to basic and elementary operations which let learner draw his/her own step with all autonomy and independently to any programming language. This approach assures a trilateral assessment: summative, formative and diagnostic assessment.

***Keywords*—**Algorithmic, assessment of competences, Technology Enhanced Learning (TEL).

I. INTRODUCTION

THE revolution of technology has allowed all educational fields to join this technological framework in order to improve the education quality and to liberate from constraints of learning in a classroom. Fig. 1 illustrates the problematic of our work.

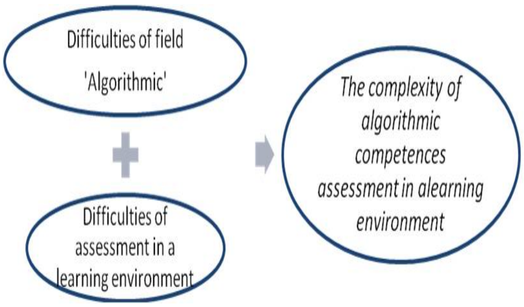

Fig. 1 Problem domain

Anis Bey, Department of Computer Science, Badji Mokhtar University, BP12, 23200, Annaba, Algeria (Fax: +213 38 87 27 56; e-mail: a-bey@hotmail.fr).

Tahar Bensebaa, Department of Computer Science, Badji Mokhtar University, BP12, 23200, Annaba, Algeria (e-mail: t_bensebaa@yahoo.com).

Hana Benelem, Department of Computer Science, Badji Mokhtar University, BP12, 23200, Annaba, Algeria (Fax: +213 38 87 27 56; e-mail: benhandz @yahoo.fr).

The first difficulty is original; it comes from the field itself. Insufficiently perceived, the intern complexity of the field or how didacticians call *the didactic blur of field* may be a source of a systematic difficulty.

Algorithms play an important role in Computer Science. Because of their importance, many researchers have been trying to find the best way to learn and teach algorithms [1]. Nevertheless, many experienced teachers in many universities in spite of their experience were since a long time confronted in difficulties of their students facing this field. [2] confirms that a high dropout rate in initiation of programming courses, during the first study years, varies from 25 to 80 % around the world.

It must admit that this subject have been used since a long time with an artless manner [3], without any particular formalism. Usually, a teacher introduces a new control structure or data structure, shows some examples to students, and then expects students to be able to solve problems that are either novel or possibly similar to those that he or she has demonstrated [4].

This example, among others shows problems acuteness which comes down for both teacher and learner. For teachers, because they must find adequate methods to assimilate abstract concepts to students, which are only in their initiation phase. For students, algorithmic oppositely to others sciences like physic, it doesn't give to the beginner an artless model viable of computer, which they can use it as a base to construct models more sophisticated. Frequently, the scientific models describe phenomena for which students have no real-life references and incorporate invisible factors and abstractions [6]. This is particularly true in the case of learning programming which is a multiple abstract process [7]. On the contrary, the experience student with algorithmic seems favoring an anthropomorphic modeling, which doesn't allow him/her to understand the brute error return which it confronted in the beginning of her/his practice [8].

Algorithm is, in addition, characterized by the multitude of solution for one problem. This characteristic increases difficulties of assessment in learning systems; it's a difficult task to expert of field to foresee all possible solution of a given problem in order to integrate them in the solution base (*indeed, he/she always forgets them !*).

Second difficulty is intrinsically connected to assessment which holds a preponderant place in large number of pedagogical activities. Researchers conceptualizing effective teaching did not assign a meaningful role to assessment as part of the learning process [9]. Assessment at classroom or in a

Technology Enhanced Learning (TEL) environment was always source of ambiguity among evaluator and learner, and sometimes among evaluators themselves. Thus, assessment is rarely considered, because it is often absent and obsolete [10]. Nevertheless, it is an integrate process in the pedagogy.

More, assessment is not restricted to attribute a mark, although it is important as far as we want to quantify skills. Evaluation constitutes a guide hall for learner's progression and intervenes in the interaction level between teacher and learner to optimize the transfer and the purchase of knowledge, skills and practices. It overtakes so theoretical framework. So, its importance is capital.

In this sense, many methods and tools have been expended to assessment in a TEL. But, they always suffer inadequacy. This inadequacy is characterized either by inefficacity, doubtful result, or by uniqueness, i.e. they can't be applied into all fields (*e.g. we can't assess algorithmic skills with QCM*).

Through this ensemble of obstacles that we can measure the size of difficulties toward pass round to strive for automatic assessment of algorithmic competences.

II. Proposal

It is known in algorithmic that to execute complex tasks, every task must be decomposed on succession of simplest tasks. This decomposition is repeated until to have elementary tasks. The number of step of decomposition depends on the complexity of problem to resolve.

This descendant approach (also named *divide-and-conquer*), source of our inspiration, allows to pass gradually and with a maximum chance to success, from the abstract description of the problem solution (*with a complex process*) to an algorithm resolving the problem. We can say that an algorithm is on the last level of decomposition when it contains only elementary operations, basic operations (*known operations*) and control structures.

We define a basic operation (BO) such as an operation well-known in algorithmic like *Sorting, etc*. Whereas elementary operation (EO) is simple instructions such as *assignment, etc*.

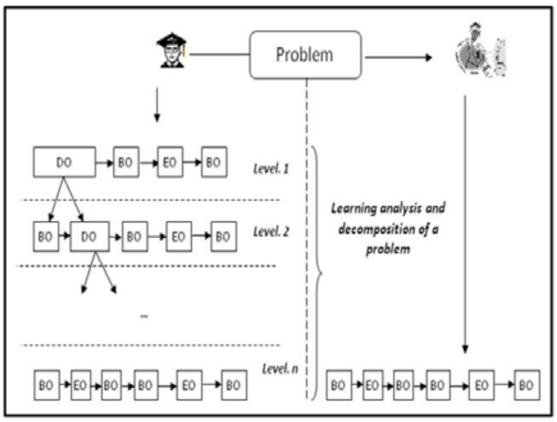

Fig. 2 Decomposition of a problem

At level one, the problem is divided into an ensemble of decomposable operation (DO), basic operations and elementary operations connected by control structures if necessary (*condition, loop*).

At level two, every decomposable operation of level one is decomposed, either at decomposable operation again, or at basic operations or elementary operations. This depends on the complexity of problem.

Thus, in an assessment context, learner may present her/his solution, to a given problem, in terms of decomposable, basic and elementary operations. These operations may be connected by control structures. This gradual draft of solution and the use of a high-level (*basic operations*) allow an efficient assessment and totally adapted to the field.

This manner to do aims to bring learner to express his/her solution (kind of action plan) in a form automatically assessed. It arises from it two consequences extremely interesting for learner:

- He/she learns to decompose a problem. This teaches him/her to decrease gradually the complexity of problem and evades him/her drowning in details at first.
- He/she focuses its efforts on the problem and not on secondary questions. Example, in the sorting problem, what interest us is if learner opts for a sorting and not how he/she makes sorting. However, basic operations may do themselves a learning object.

III. Modeling And Evaluation

The variety of solutions for an algorithmic problem allows learner, during solving this problem, take a path among several which leads to solution. See Fig. 3.

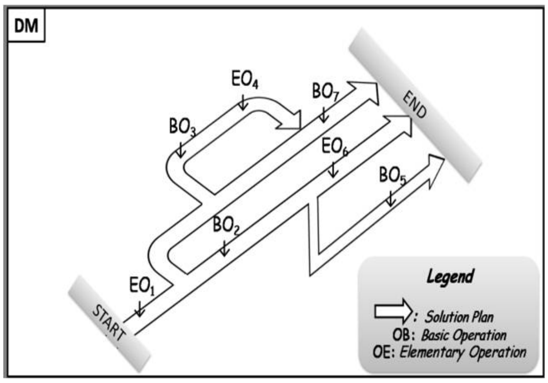

Fig. 3 Descriptive Map

We call plan, a path constituted by basic and elementary operations. This plan describes a solution pedagogically interesting. The whole of plans (SP) constitute a Descriptive Map (DM) of the problem.

This descriptive map allows to locate learner and to recognize its step (right or erroneous).

Holding BO (in the form of libraries) and EO, learner can express its solution freely without any restriction neither influences.

Teacher, for its part, should define beforehand for every problem a DM. we know in advance that it will not be complete (describes all solutions/errors of problems), but he/she must foresee the most plausible SP, leaning on his/her experience and given for each SP an interpretation and an appreciation. In addition, during the definition of a SP, teacher attributes for each BO/EO a weight expressing its importance in the solution. The assignment of weight depends on the pedagogical objectives aimed.

DM is advancing. They can be enriching by other SP, integrating other solutions proposed by learner but not foresee by teacher. Indeed, every solution not recognized, its evaluation should be suspended until the intervention of a human teacher which if he/she judges it interesting (should be correct or not), will add it into the DM.

IV. Assessment Script

The assessment produced is an individual, summative, diagnostic and formative assessment. To evaluate a plan proposed by learner, the last one must be expressed only with BO and EO; next, it should recognize in the DM the nearest plan (called the reference plan). The recognition of the reference plan is done by (1) which calculates the similarity degree SD of the plan proposed by learner with all SP of the problem map. This formula is essentially based on two parameters, succession of BO/EO (the order) and use of an inappropriate BO/EO.

$$SD = \frac{Nb'S}{Nb} - \left(P_1.10^{-1}\frac{Nb'SD}{Nb} + P_2.10^{-1}\frac{Nb'I}{Nb} \right). \quad (1)$$

(Nb'S/Nb): Represents the similarity operation by operation,
(Nb'SD/Nb): Represents the penalty for the disordered BO/EO,
(Nb'I/Nb): Represents the penalty for the useless BO/EO,

After simplification, (1) becomes (2).

$$SD = \frac{1}{Nb}\left[Nb'S - \left(P_1.10^{-1} Nb'SD + P_2 10^{-1} Nb'I \right) \right]. \quad (2)$$

Where:

Nb: number of BO/EO of the DMs' plan,

Nb'S: number of BO/EO of the proposed plan, which are similar to BO/EO of DMs' SP,

Nb'SD: number of the disordered BO/EO of the proposed plan,

Nb'I: number of the useless BO/EO of the proposed plan.

When the max SD is computed, it will be interpreted as following:

If SD=1, the mark attributed is the reference plan mark (the proposed plan is identical to the reference plan) and comments are presented to learner in order to reinforce its competences.

If Threshold ≤SD<1, the mark attributed is calculated by (3), which brings weight of every BO/EO, even so comments and explanations are presented to learner alike.

$$eval = \begin{bmatrix} \left(\sum p(OS)\right) - \\ \begin{pmatrix} P_1.10^{-1}\left(\sum p(OSD)\right) + P_2.10^{-1} \\ \left(1 - \left(\sum p(OSD)\right)\right)\left(\sum p(OS)\right) \end{pmatrix} \end{bmatrix} * Mark \;. \quad (3)$$

Where:

Mark: the mark of the reference plan,

$\sum p$ *(OS):* sum of the similar BO/EO weights,

$\sum p$ *(OSD):* sum of the similar disordered BO/EO weights,

When the proposed plan is not recognized (*SD< Threshold*) among SP of the DM, in this case, it will be reviewed by the teacher (*responsible*) which can add it as a new SP or simply ignored it.

During searching reference plan among SP of DM, and in order to reduce computation time, we introduce a filter before computing SD: a solution plan can't be a candidate only if (4) is verified.

$$\frac{Number\ of\ BO\ and\ EO\ of\ compared\ \ plan}{Number\ of\ BO\ and\ EO\ of\ the\ SP\ of\ DM} . \geq Threshold \quad (4)$$

Note: In all equations, $P_1, P_2 \in [0..10]$ (zero not penalty, 10 max of penalty) parameters determined by teacher. They represent penalties values for disordered BO/EO and useless BO/EO.

V. Conclusion

EASEL assures a trilateral assessment: summative (to quantify competences and knowledge really owned and to attribute a mark), formative (to aid learner to progress) and diagnostic assessment (to diagnose the gaps).

In addition, the advantage is that every step proposed by learner and recognized by system will be enriched by explanation; demonstration … The expertise (the evolution DMs) will get in some times (after a learning phase) a maximum of competences.

Though it has been conceived for learning algorithmic competences, this approach may be easily adapted in any field which manipulates procedural knowledge.

A prototype has been developed. We experiment it currently to validate both the method and formulas created for recognition and marking.

References

[1] Tomasz Müldner and Elhadi Shakshuki, 2003. Teaching Student to Implement Algorithms. *Jodrey School of Computer Science*, Acadia University. TR-2003-03 November, 2003.
[2] Kaasboll, J.,2002. *Learning Programming*. University of Oslo.
[3] Caignaert C, 1988. *Étude de l'évolution des methods d'apprentissage et de programmation*. Le bulletin de l'EPI N°50, Juin 1988.
[4] Stuart Garner, 2003. "Where Parallels Intersect" Learning Resources and Tools to Aid Novices Learn Programming. *Informing Science InSITE*. Edith Cowan University, Perth, Australia.
[5] Ben-Ari M.,1998. *Constructivism in Computer Science Education*. ACM IGCSE Bulletin, 30(1): pp.257-261.
[6] Dede, C., Salzman, M. C., Loftin, R. B., & Sprague, D. 1999. Multisensory immersion as a modelling environment for learning complex scientific concepts. In N. Roberts,W. Feurzeig, & B. Hunter (Eds.), *Computer modelling and simulation in science education* (pp. 282–319). New York: Springer-Verlag.

[7] Katai, Z., Juhasz, K., & Adorjani, A. K . 2008. On the role of senses in education. *Computers & Education*. 51(4), 1707–1717. Student performance in traditional mode and online mode of learning.

[8] Guibert, N., Guittet, L., & Girard, P. 2005. Initiation à la Programmation « par l'exemple »: concepts, environnement, et étude d'utilité. *Acte de colloque EIAH'05*, Montpellier: 25-27 Mai, 461-466.

[9] Lorrie A. Shepard, 2000. The Role of Classroom Assessment in Teaching and Learning. *CSE Technical Report 517 CRESST*. University of Colorado at Boulder.

[10] Guillaume Durand, 2006. Vers une scénarisation de l'évaluation en EIAH. L'évaluation comme activité scénarisable dans un dispositif de scénarisation pédagogique*. RJC-EIAH'2006.*