Algobase, a System for Assessing Algorithms

Rym AIOUNI, Tahar BENSEBAA
Laboratory of Computer Serch (LRI)
University Badji Mokhtar Annaba
Algeria
aiouni.rym@gmail.com, t_bensebaa@yahoo.com

ABSTRACT: *An algorithm is a consequence of logical instructions that, once executed, reduce to a result and it is not always evident to good understand its functioning.*

The progression of the technologies of information and communication and its availability in the teaching and in the learning offer the possibility to put at the disposal of the learners the tools that facilitate their learning. The algorithmic is a very important field in computer science; it is currently studied using different intelligent environments for human learning dedicated to each discipline.

Algobase, is an Environment for assessment algorithms and learning resolution with decomposition of tasks. This is an editor that offers the learner opportunities to create, modify, evaluate algorithms and view mistakes. The assessment in this Environment is done through a matching of solutions.

Keyword: Algorithmic, Matching, Organigram, Assessment, Environment

Received: 20 July 2012, Revised 8 October 2012, Accepted 15 October 2012

1. Problem Statement

The algorithmic is discipline that used for long time but in a naïve way [1], without particular formalism. This discipline is often source of problem for both the teacher and students. For the teacher, because he must find adequate methods to let the students grasp some of the abstract aspects that are in their beginning level. For students, the problem is much bigger. In testifies the rate of failure or dismissal from the initiating level in programming in the first cycle of the university vary from 25 to 80% worldwide [2]. Works in cognitive psychology directly involves the nature of the taught discipline. These studies have identified the major axes of the intrinsic difficulties related to algorithmic:

• In the algorithmic, unlike other sciences such as physics, the beginning student has no model "*naïf*" of computer that might serve as a basis for building more sophisticated models.

opposite to, his experience with it seems to favor a modeling "*anthropomorphic* " which did not explain the sudden feedback which it is faced early in his practice of algorithmic [3].

• Another specific difficulty in algorithmic is the abstraction of the task: the learner must factorize in the algorithm, the overall behavior of the task. The result is a "*syndrome of the white page*", highlighted in particular by Kaasböll [2]. According to the students: "*... when the problem is presented ... it breaks it down like that, like it like that. Everything seems simple and logical, and that's you and Ouch*! *Where do I begin*? *Perhaps it is easy, but the problem is that you do not know where to start when it should solve the problem....*".

The following essential question arises:

- With witch pedagogic methods and tools could we improve the algorithmic learning?

Since some years, the integration of the technology of information and communication (TIC) has boosted the improvement of the quality of teaching and learning of different knowledge [4]. We support that appropriate use of TICs with innovative teaching methods and tools appropriated to the context, could be the solution to the problem of learning algorithmic.

The TLE has been known since some time considerable improvement efforts. Whatsoever in the manner of describe, to index pedagogic content but also to script pedagogic activities, formalisms imposed. In this evolution, evaluation is the poor relative. There is no particular formalism to specify the evaluation of learners. The evaluation, crucial dimension of the pedagogic activity, in its certificative role, formative, summative or normative, is widely found mistreated in the TLE.

This is in large part caused by difficulties of evaluation itself. Several tools and methods have been devoted but they are either ineffective (doubtful) or dedicated (they can't be applied to any field) [5], [6], [7], [8], [9].

Furthermore, the algorithmic activities of evaluation are among the most delicate, especially in a TLE, because the algorithmic is characterized by the multitude of solutions for a given problem. This feature increases the difficulty of evaluation in learning environments: the expert of the field finds difficulties in finding all possible solutions for a problem to integrate them into the database solutions. The localization of mistakes is an important fact of the progression of learners, is another problem caused by this feature. This makes the complex realization of these systems.

2. Our Proposal

To simplify the complex tasks, we need to decompose them into less complex ones and repeat this process until reaching a level of decomposition including basic operations and / or elementary ones. The algorithm solving the problem will be a composition of the late operations (basic and elementary). The number of levels of decomposition depends on the complexity of the problem: the more it's complex, the more the number of steps would be considerable.

This method of refining successive (also called top-down approach) allows a progressive movement with maximum chance of success, from the abstract description of the solution of problem (per a complex operation) to the algorithm that would allow its resolution. The algorithm is in the last level of refining when it contains only basic operations, elementary ones and the control structures.

We define a basic operation as being a known operation in algorithmic such as sorting an array, whereas, the elementary operation is a simple algorithmic one (e.g. assignment).

Thus, in the first level, the problem is decomposed into a set of basic operations, elementary ones and operations that can be decomposed, related by controlling structures. The number of levels depends on the complexity of the problem. Going down in levels, only decomposable operations are decomposed and the decomposition stops when we reach a level in which the constituents are only basic and elementary operations.

This approach prevents the learner to be drowning in the details at the outset and gradually reduces the complexity of the addressed problem. In addition to that, the learner can freely express its solution, without any influence or restriction, which favors the autonomy.

Our objective by this approach is to evaluate algorithmic solutions. However, the essential fallout is the learning by the learners of the decomposition. In fact, this is a bond-holder passage for the learner in the formulation of his solution.

3. Validation of the Solution

Recall that our objective is to reach a reliable evaluation for algorithmic solutions. Thus, when the learner has completed its decomposition problem that has been proposed, its solution is compared to those of the expert grouped into a plan of solutions.

A plan of solutions is a set of paths representing the different possible ways (solutions) for the one exercise. That can contain wrong and correct solutions. It is constituted by an expert and includes solutions judged pedagogically interesting.

This plan could suffer from an exhaustive problem of completeness of expected solutions. It was solved as the following, every unrecognized solution; its evaluation will remain pending until a human expert would add it to the plan of solutions. This progress in the plan of solutions guarantees an evaluation whatever the solution suggested by the learner is. Thus, by time the deferred evaluation will decrease to the benefit of the direct evaluation.

4. Recgnition of Solutions

To measure the similarity between two solutions, it is necessary to establish a correspondence between their components. More specifically, the point is to find the best possible matching: one that put in correspondence the components that are the most similar, the similarity of the components is being based on the features they have in common. A first important point of similarity measure we use in relation to existing measures is that is not only quantitative (evaluating the degree of similarity of two solutions) but also qualitative (explaining in what the solutions are similar and in what they are different).

A second important point of our measure is that it allows to define the relative importance of features, relative to each other and therefore to introduce knowledge in calculating similarity.

4.1 The Matching Method AMAS (Automatic Matching Algorithms Solutions)

To automate the comparison of the process of learning with those of the expert (plane solutions), and drawing on the works of [10] on the measurement of multi-labeled graphs, we propose a method for matching algorithmic solutions (Figure 1)

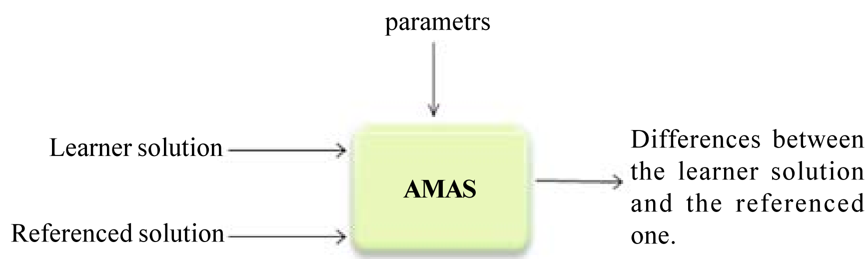

Figure 1. The matching method

This method involves comparing the solution of each learner with the expert solutions to measure the similarity between them. It is composed of two sequential steps (Figure 2):

1. Generate the description of the solution of the learner.

2. Measure the similarity between the solution of the learner and the solution referenced.

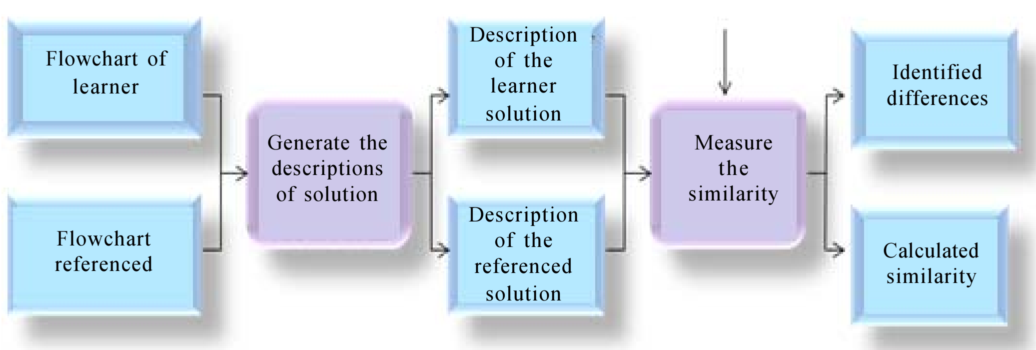

Figure 2. Steps of matching method AMAS

5. Algobase, A System for Assessing Algorithms

Our environment offers the learner a number of activities.

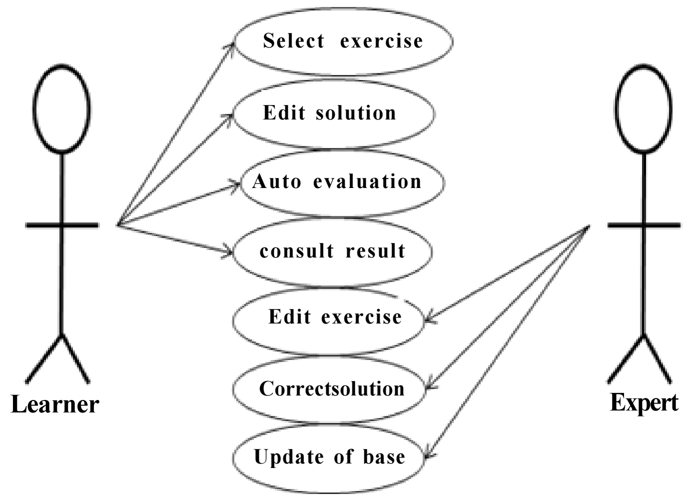

Figure 3. Use case Diagram of AlgoBase

Starting by our objectives, we decided to distinguish two main actors in our system: the learner and the expert (teacher).

The expert's role is to manage the databases, making changes on the evolution of exercises and evaluate solutions when necessary.

The learner can consult existing exercises to select one. It can edit exercises in flowchart form. He is also invited to make self-assessments to confirm and read his annotations in the form of notes or summaries.

Both actors can access various activities throughout the environment.

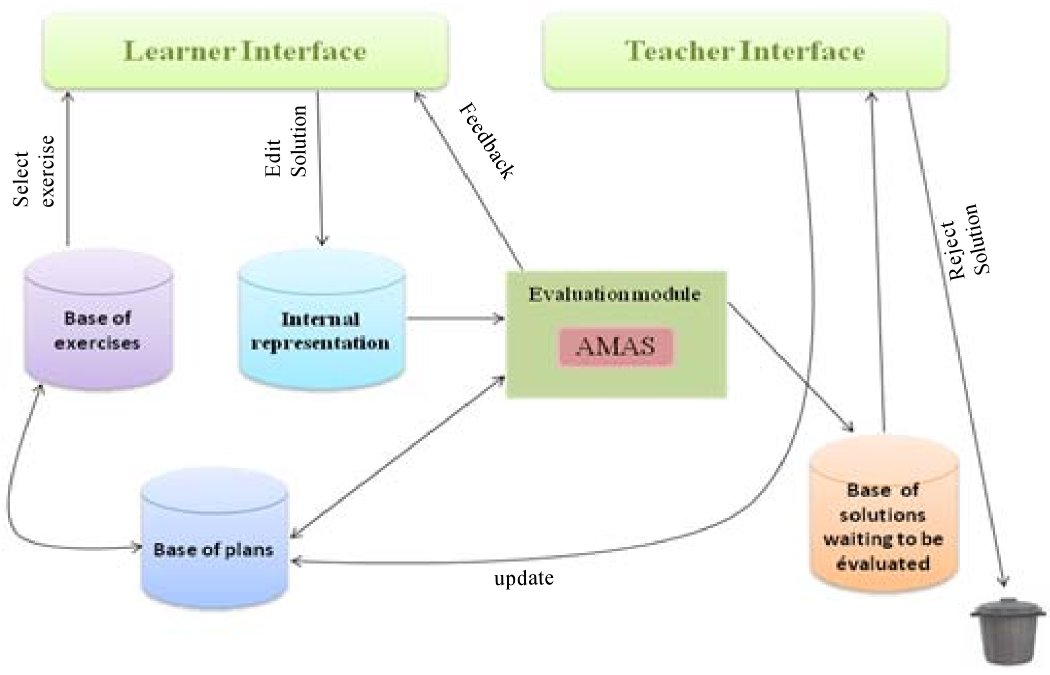

Figure 4. Architecture of AlgoBase

5.1 Algobase editor of solutions

This Editor offers the learner the opportunity to build its solution in flowchart form, it has at its disposal a library of operations and control structures.

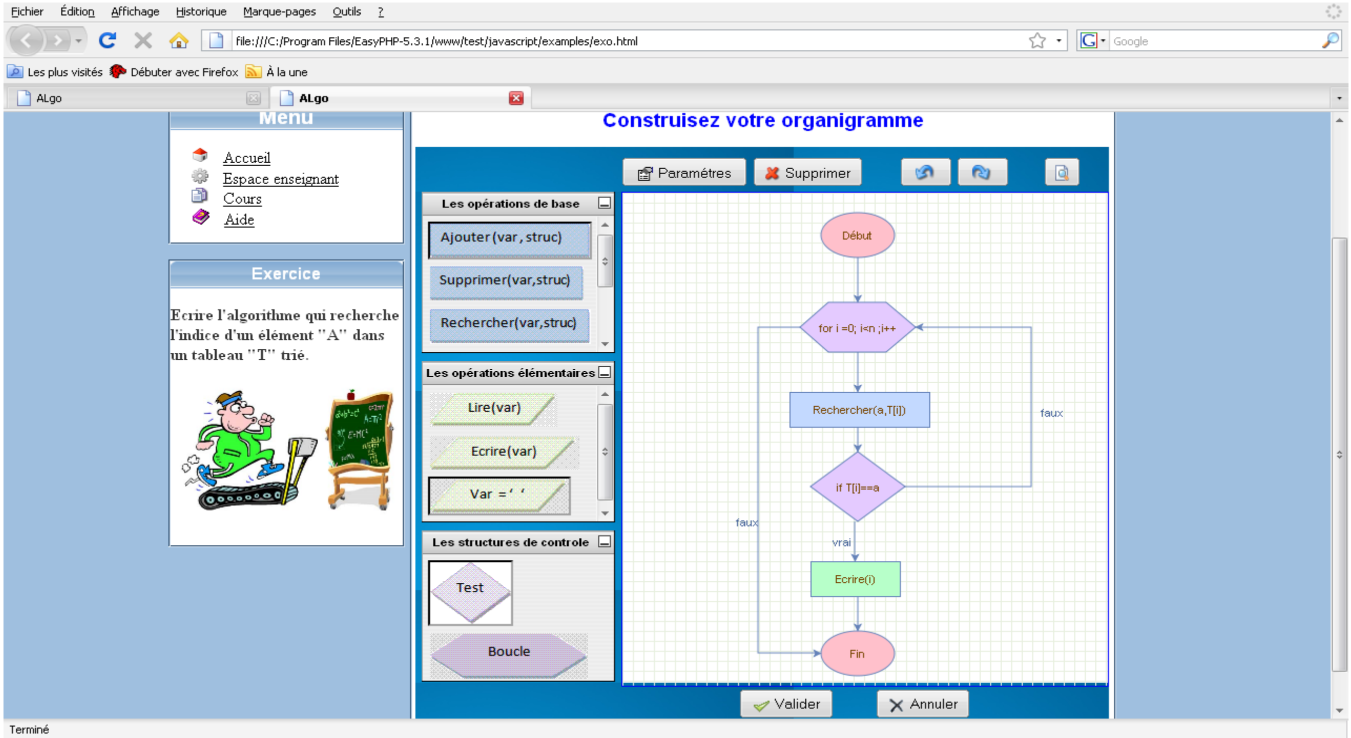

Figure 5. Interface of editor

6. Conclusion

The method of evaluation that we propose imposes, by its nature, to the learners to decompose his problem so that its solution is evaluated. This would *oblige* them to perform the decomposition.

This method is, in the other hand totally adequate for the field taught. As the result, it's trustful and not given in other suggested systems of evaluation of algorithmic, that are generally based on methods totally inadequate (MCQ...).

The feedback provided by the plan of solutions of a recognized solution is itself a source of learning. Thus, in addition to being summative, this evaluation is also formative.

Currently, a prototype is being tested with students from 2nd year LMD. The objective is to see after how much time the system acquires stability. By stability we mean reducing the maximum response of the human expert.

References

[1] Caignaert, C. (1988). Étude de l'évolution des méthodes d'apprentissage et de programmation. *Bulletin of EPI*, 50, 52-60.

[2] Kaasboll, J. (2002). *Learning Programming*. University of Oslo.

[3] Guibert, N., Guittet, L. Girard, P. (2005). Initiation à la Programmation « par l'exemple : concepts, environnement, et étude d'utilité. *conference proceeding EIAH'05*, Montpellier: 25-27 May, p. 461-466.

[4] Amerein, S. B., Proquin, M., Renaud, C., Trigano, P. (1998). De la réciprocité éducative dans le cadre d'une nouvelle pédagogie de l'enseignement supérieur : un didacticiel au service de l'informatique fondamentale. *conférence proceedingNTICF'98*. Rouen : 18-20 November.

[5] Jean-Daubias, S. (2003). Exploitation de profils d'apprenants. *conference proceeding EIAH'03*, Strasbourg: 15-17 april, p. 535-538.

[6] Allal, L. (2005). L'évaluation formative de l'apprentissage : revue de publication en langue française, Formative Assessment-Improving Learning in secondary classes.

[8] Labat, J. (2002). EIAH : Quel retour d'information pour le tuteur ?, Proceedings of the colloquium.

[7] Charle, H. (1977). l'évaluation démystifiée, ESF Editor.

[9] Benabbou Faouzia, Hanoune Mostafa. (2007). *Utilisation des NTICs pour l'apprentissage et l'autoévaluation de l'algorithmiqu. SETIT*.

[10] Sorlin, S., Sammound, O., Solnon, C., Jolion, J.-M. (2006). Mesurer la Similarité de Graphes, *In*: Proceedings of the 6^{th} day francophones Extraction and Knowledge Management 2006*, Atelier ECOI 2006*, January, p. 21-33.