

Approche d'évaluation de l'algorithmique dans un EIAH

AIOUNI Rym	BENSEBAA Tahar	BENSALEM Hana
<i>Université Badji Mokhtar</i>	<i>Université Badji Mokhtar</i>	<i>Université Badji Mokhtar</i>
<i>Annaba- Algérie</i>	<i>Annaba- Algérie</i>	<i>Annaba- Algérie</i>
aiouni.rym@gmail.com	t_bensebaa@yahoo.com	benhandz@yahoo.fr

2010

Résumé

L'évaluation dans un EIAH en général et dans un EIAH de l'algorithmique en particulier, pose un problème assez complexe. Dans ce sens, ce travail présente un processus d'évaluation de l'algorithmique visant un double objectif. Celui de l'apprentissage de la décomposition des problèmes et celui de la validation des solutions. Cette validation repose sur un appariement structurel.

Mots clés

Algorithmique, appariement structurel, évaluation, raffinement successif, plan de solutions ...

1. Problématique

L'algorithmique est une discipline longtemps utilisée de manière naïve [1], sans formalisme particulier. Cette discipline est souvent source de problème aussi bien pour l'enseignant que pour l'étudiant. Pour l'enseignant, parce qu'il doit trouver les méthodes adéquates pour faire assimiler des concepts assez abstraits à des étudiants qui ne sont qu'à leur phase d'initiation. Pour les étudiants, le problème est encore plus important. En témoigne le taux d'échec ou d'abandon aux cours d'initiation à la programmation en premier cycle universitaire varient de 25 à 80% de par le monde Kaasböll [2]. Des travaux en psychologie cognitive impliquent directement la nature même de la matière enseignée. Ces travaux ont dégagé les axes majeurs des difficultés intrinsèques liées à l'algorithmique :

- En algorithmique, contrairement à d'autres sciences comme la physique, l'étudiant débutant n'a pas de modèle "naïf" viable de l'ordinateur, qui pourrait lui servir comme base pour construire des modèles plus sophistiqués. A l'inverse, son expérience avec celui-ci semble favoriser une modélisation "anthropomorphique" qui ne lui permet pas de comprendre le retour d'erreur brutal auquel il est confronté aux débuts de sa pratique de l'algorithmique [3].
- Une autre difficulté spécifique à l'algorithmique est l'abstraction de la tâche : l'apprenant doit factoriser, dans l'algorithme, l'ensemble des comportements de la tâche. Il en résulte un « syndrome de la page blanche », mis en évidence notamment par Kaasböll. Selon les étudiants : « ... lorsque le problème est présenté ... on le décompose comme ça, comme ça, comme ça. Tout a l'air simple et très logique, et puis c'est à toi et Ouch ! Par quoi je commence ? Peut-être que c'est facile, mais le problème c'est que tu ne sais pas par quel bout commencer quand il faut résoudre le problème ... ».

Se posent alors la question essentielle suivante :

- Quelles méthodes pédagogiques et avec quels outils peut-on améliorer l'apprentissage de l'algorithmique ?

Depuis quelques années, l'intégration des technologies d'information et de communication (TIC) a relancé l'amélioration de la qualité de l'enseignement et de l'apprentissage des différents savoirs [9]. Nous soutenons qu'un

usage approprié des TIC, avec des méthodes pédagogiques innovatrices et des outils appropriés au contexte, pourrait constituer la solution au problème de l'apprentissage de l'algorithmique.

Les EIAH ont connu depuis quelque temps d'importants efforts d'amélioration. Que ce soit dans la manière de décrire, d'indexer les contenus pédagogiques mais aussi de scénariser les activités pédagogiques, des formalismes s'imposent. Dans cette évolution, l'évaluation est le parent pauvre. Il n'existe pas de formalisme particulier pour spécifier l'évaluation des apprenants. L'évaluation, dimension cruciale de l'activité pédagogique, dans sa fonction certificatrice, formative, normative ou sommative, se retrouve largement mal traitée en EIAHs.

Ceci est en grande partie du aux difficultés de l'évaluation elle-même. Plusieurs méthodes et outils y ont été consacrés mais ils sont tous soit inefficaces (douteux) soit dédiés (ils ne peuvent pas être appliqués à n'importe quel domaine). [4], [5], [6], [7], [8].

Par ailleurs, Les activités d'évaluation en algorithmique sont parmi les plus délicates et plus particulièrement dans un EIAH, car l'algorithmique est caractérisée par la multitude de solutions pour un problème donné. Cette caractéristique accroît la difficulté de l'évaluation dans les systèmes d'apprentissage : l'expert du domaine a du mal à trouver toutes les solutions possibles d'un problème afin de les intégrer dans la base de solutions. La localisation des erreurs qui est un facteur important pour la progression des apprenants est une autre difficulté engendrée par cette caractéristique. Cela rend complexe la réalisation de ces systèmes.

2. Notre proposition

Pour simplifier des tâches complexes, il faut les décomposer en des tâches moins complexes et répéter cette opération jusqu'à arriver à un niveau de décomposition comportant des opérations de base et/ou des opérations élémentaires. L'algorithmique solution du problème sera alors une composition de ces dernières opérations (de base et élémentaires). Le nombre d'étapes de décomposition dépend de la complexité du problème : plus ce dernier est complexe, plus le nombre d'étapes est important.

Cette méthode de raffinages successifs (dite également approche descendante) permet de passer progressivement, et avec un maximum de chances de réussite, de la description abstraite de la solution du problème (par une opération complexe) à l'algorithmique qui permettra sa résolution. L'algorithmique est au dernier niveau de raffinement lorsqu'il ne comporte que des opérations de base, des opérations élémentaires et des structures de contrôle.

On définit une opération de base comme étant une opération connue en algorithmique telle que le tri d'un tableau. Une opération élémentaire, quant à elle, est une opération algorithmique simple (exemple : l'affectation).

Ainsi, au niveau 1, le problème est décomposé en un ensemble d'opérations de base, d'opérations élémentaires et d'opérations décomposables qui peuvent être liées par des structures de contrôle. Le nombre de niveaux de décomposition dépend de la complexité du problème à résoudre. En descendant dans les niveaux, seules les opérations décomposables sont décomposées, et cette décomposition s'arrête lorsqu'on arrive à un niveau constitué uniquement d'opérations de base et d'opérations élémentaires (figure 1).

Cette approche évite à l'apprenant de se noyer dans les détails dès le départ et diminue graduellement la complexité du problème abordé. En plus, l'apprenant peut librement exprimer sa solution, sans aucune influence ou restriction, ce qui favorise l'autonomie.

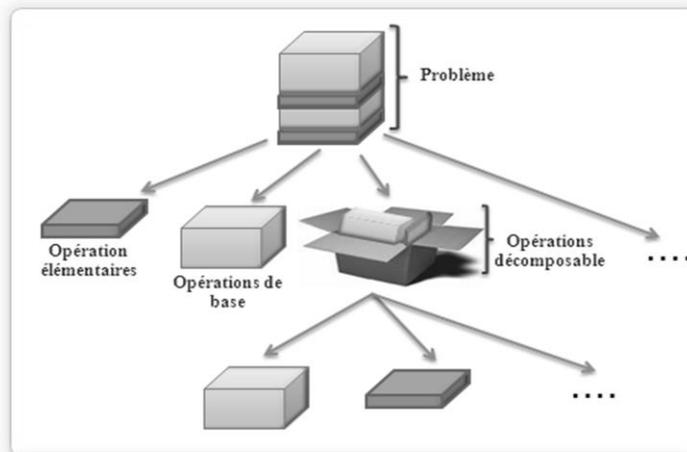


Figure 1 : schéma de décomposition d'un problème algorithmique

Notre objectif, par cette approche, est d'évaluer des solutions algorithmiques. Cependant, la retombée essentielle est l'apprentissage par l'apprenant de la décomposition. En effet, celle-ci est un passage obligé pour l'apprenant dans la formulation de sa solution.

3. Validation de la solution

Rappelons que notre objectif est d'arriver à une évaluation fiable pour les solutions algorithmiques. Ainsi, lorsque l'apprenant aura terminé sa décomposition du problème qui lui a été proposé, sa solution est comparée à celles de l'expert regroupées dans un plan de solutions.

Un plan de solutions est un ensemble de chemins représentant les différentes démarches possibles pour un même exercice. Il peut contenir aussi bien les démarches correctes que les démarches erronées. Il est constitué par un expert et comporte les démarches jugées pédagogiquement intéressantes.

Ce plan pourrait souffrir d'un problème d'exhaustivité des démarches attendues. Ceci a été résolu comme suit : toute démarche non reconnue verra son évaluation suspendue jusqu'à intervention de l'expert humain qui, s'il la juge intéressante, pourra la rajouter au plan de solutions. Cette évolutivité du plan de solutions garantit une évaluation quelque soit la solution proposée par l'apprenant. Ainsi, plus le temps passe plus l'évaluation différée diminuera au profit de l'évaluation directe.

4. Reconnaissance des solutions

Afin de mesurer la similarité entre deux solutions, il est nécessaire d'établir une correspondance entre leurs composants. Plus précisément, il s'agit de trouver le meilleur appariement possible : celui qui met en correspondance les composants les plus similaires, la similarité des composants étant fonction des caractéristiques qu'ils ont en commun. Un premier point important de la mesure de similarité que nous utilisons par rapport aux mesures existantes, est qu'elle n'est pas seulement quantitative (évaluant le degré de similarité de deux solutions) mais aussi qualitative (explicitant en quoi les solutions sont similaires et en quoi elles sont différentes).

Un second point important de notre mesure est qu'elle permet de définir l'importance relative des caractéristiques, les unes par rapport aux autres et par conséquent d'introduire la connaissance dans le calcul de similarité.

4.1 Principe

Afin d'automatiser la comparaison de la démarche de l'apprenant avec celles de l'expert (plan de solutions) nous proposons d'affecter une description pour chacune des démarches. Cette description est nécessaire pour l'appariement pour lequel nous nous sommes inspirés des travaux de Sorlin [10] qui proposent une mesure générique de similarité.

Pour mesurer la similarité entre deux démarches, il est nécessaire de mettre leurs descriptions en correspondance. Une solution algorithmique peut être vue comme un organigramme. A cet organigramme, on attribue une définition formelle $S = \langle O, r_o, r_t \rangle$. Etant donné L_o un ensemble fini d'étiquettes d'opérations et L_t un ensemble fini d'étiquettes de transitions, tels que :

- O est un ensemble fini d'opérations.
- $r_o \subseteq O * L_o$ est la relation associant opérations et étiquettes, i.e. r_o est l'ensemble des couples (o_i, l) tels que l'opération o_i est étiquetée par l .
- $r_t \subseteq O * O * L_t$ est la relation associant transitions et étiquettes, i.e. r_t est l'ensemble des tuples (o_i, o_j, l) tels que la transition (o_i, o_j) est étiquetée par l .

La description de la démarche S est l'ensemble de toutes ses caractéristiques d'opérations et de transitions, $Descr(S) = r_o \cup r_t$ (Figure 3) :

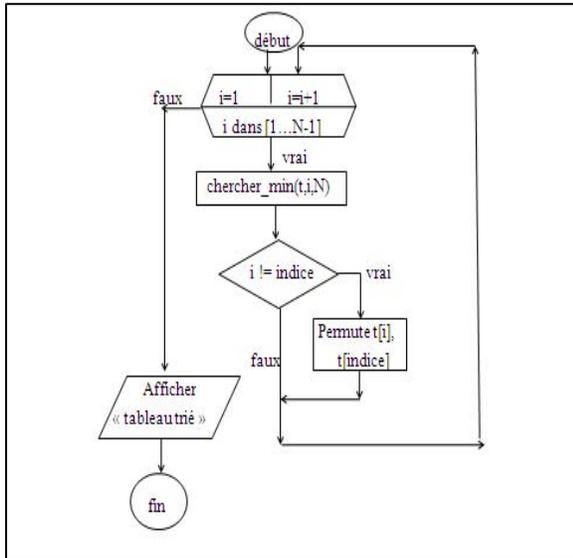


Figure 2 : Exemple d'organigramme.

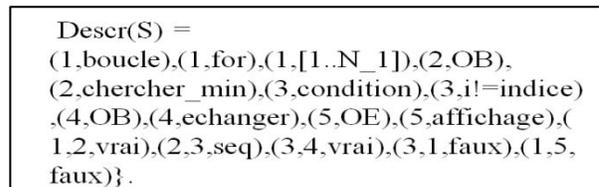


Figure 3 : Description de l'organigramme de la figure 2.

Un appariement entre deux démarches $S_1 = \langle O_1, r_{O1}, r_{t1} \rangle$ et $S_2 = \langle O_2, r_{O2}, r_{t2} \rangle$, est une relation : $m \subseteq O_1 * O_2$. Un tel appariement associe à chaque opération d'une solution de zéro à n opérations de l'autre solution. Pour mesurer la similarité entre deux démarches par rapport à un appariement m, nous proposons d'adapter la formule de similarité de Travesky [11] :

$$Sim1_m(S1, S2) = \frac{f(descr(S1) \cap descr(S2))}{f(descr(S1) \cup descr(S2))} \quad (1)$$

La fonction f définit l'importance relative des caractéristiques, les unes par rapport aux autres. Cette fonction est souvent définie comme une somme pondérée :

$$f(F) = \sum_{(o,l) \in F} poids(o, l) + \sum_{(o1,o2,l) \in S} poids(o1, o2, l) \quad (2)$$

L'attribution des poids aux différents descripteurs d'une solution est réalisée par l'expert, celui-ci tient compte de l'importance du descripteur, ainsi que de l'objectif de l'exercice résolu par l'apprenant.

Il est évident que la solution retenue sera celle présentant la plus grande similarité. Celle-ci doit impérativement dépasser un seuil d'acceptabilité.

Le degré de similarité calculé sert également à calculer la note à attribuer à l'apprenant.

5. Conclusion

La technique d'évaluation que nous proposons impose, de par sa nature, à l'apprenant de décomposer son problème pour que sa solution soit évaluée. Ceci aura comme retombée de *le forcer* à pratiquer la décomposition. Cette technique est, par ailleurs, en totale adéquation avec la matière enseignée. Il en résulte une fiabilité que n'offrent pas les systèmes d'évaluation de l'algorithmique jusqu'ici proposés et généralement basés sur des techniques totalement inadéquates (qcm, ...).

Le feedback fourni par le plan de solutions d'une démarche reconnue est lui-même source d'apprentissage. Ainsi, en plus d'être sommative, cette évaluation est également formative.

Actuellement, un prototype est en cours de test avec des étudiants de 2e année LMD. L'objectif est de voir au bout de combien de temps le système acquiert une stabilité. Par stabilité nous entendons diminuer au maximum les interventions de l'expert humain.

6. Références

- [1] Caignaert C. Étude de l'évolution des méthodes d'apprentissage et de programmation le bulletin de L'EPI N° 50, juin 1988.
- [2] Kaasboll, J., *Learning Programming*. 2002, Université de Oslo.
- [3] Nicolas Guibert, Laurent Guittet, Patrick Girard, Apprendre la programmation par l'exemple : méthode et système. Proceedings of the 17th international conférence on Francophone sur l'Interaction Homme-Machine 2005.
- [4] Jean-Daubias, S. (2000), «Pépite : un système d'assistance au diagnostic de compétences », Thèse de doctorat de l'Université du Maine.
- [5] Allal, L. (2005), « L'évaluation formative de l'apprentissage : revue de publication en langue française », L'évaluation formative- Pour un meilleur apprentissage dans les classes secondaires.
- [6] Charle, H. (1977), « l'évaluation démystifiée », ESF Editeur.
- [7] Labat, J. (2002), « EIAH : Quel retour d'information pour le tuteur ? », Actes du colloque.
- [8] Benabbou Faouzia, Hanoune Mostafa, Utilisation des NTICs pour l'apprentissage et l'autoévaluation de l'algorithmique.SETIT 2007.
- [9] Amerein S. B., Proquin, M., Renaud, C., Trigano, P. *De la réciprocité éducative dans le cadre d'une nouvelle pédagogie de l'enseignement supérieur : un didacticiel au service de l'informatique fondamentale*. NTICF'98, ROUEN.
- [10] Sorlin, S., Sammound, O., Solnon, C., Jolion, J.-M., « Mesurer la Similarité de Graphes », *Actes des 6e journées francophones Extraction et Gestion des Connaissances 2006, Atelier ECOI 2006*, janvier 2006, p. 21-3