

Système AI-VT basé sur l'intelligence artificielle, générateur de listes d'exercices personnalisées et variées.

Françoise Greffier¹, Julien Henriet²

¹ Université de Bourgogne Franche-Comté, Laboratoire ELLIADD, EA 4661
Fédération de Recherche FR-Educ, UFC
F-25000, France
francoise.greffier@univ-fcomte.fr

² Université de Bourgogne-Franche-Comté, Institut FEMTO-ST / CNRS UMR 6174,
F-25000, France
julien.henriet@univ-fcomte.fr

Résumé. AI-Virtual Trainer (AI-VT) est un système d'aide à l'apprentissage basé sur l'intelligence artificielle proposant des exercices variés et personnalisés pour l'acquisition de compétences. AI-VT a été conçu pour des étudiants en difficulté devant s'entraîner à la conception d'algorithmes. AI-VT est conçu pour sélectionner des énoncés liés à une compétence choisie, personnalisés et à difficulté graduelle.

Mots-clés. Apprentissage par compétence, entraînement personnalisé, intelligence artificielle, raisonnement à partir de cas, système multi-agents.

Abstract. AI-Virtual Trainer (AI-VT) is an educative system based on artificial intelligence. AI-VT proposes varied and personalised questions to train to acquire any skill. AI-VT has been performed in order to help students with high difficulty in acquiring competences in the field of algorithm. AI-VT is designed to select gradual sets of appropriate and connected questions adapted to the trained skill.

Keywords. Skill-based learning, personalised training, artificial intelligence, case-based reasoning, multi-agent system.

1 Contexte et besoin

Le système AI-VT (*Artificial Intelligence Virtual Trainer*) basé sur l'intelligence artificielle, vise l'instrumentation d'une pédagogie par compétences soutenue par des entraînements. AI-VT est actuellement appliqué à l'enseignement à l'université des « bases de l'algorithmique » pour des étudiants de 1ère année aux niveaux hétérogènes et en grandes difficultés. Cette formation est particulière puisque les étudiants ont déjà suivi un enseignement en « bases pour la programmation » au premier semestre. Leur niveau en algorithmique est très hétérogène et une pédagogie tenant compte de leurs difficultés, est requise. Une étude [1] menée par le réseau PENSERA (Pédagogie de l'Enseignement Supérieur en Rhône-Alpes) sur les difficultés des étudiants en première année universitaire atteste que la formation universitaire en première année, devrait notamment évoluer pour développer davantage d'échanges afin de favoriser une prise de conscience des étudiants sur leurs méthodes de travail préexistantes et identifier leurs forces et leurs faiblesses. Egalement, travailler à partir d'exemples concrets issus des unités d'enseignement disciplinaires pour leur apprendre à adapter leurs méthodes de travail aux champs disciplinaires étudiés. Selon [2] la « La multiplication des occasions de se confronter à des tâches d'une même famille, particulièrement quand le processus est méthodique et raisonné, développe le niveau de compétence ou d'expertise par rapport à cette classe ou famille de tâches. ».

Notre hypothèse principale est donc que la répétition d'exercices traitant d'exemples concrets, et appartenant à une d'une même famille de compétences, conduira l'étudiant à maîtriser des habiletés ou compétences visées, pour produire des algorithmes.

Le système AI-VT génère pour chaque séance de cours une liste d'exercices personnalisée tenant compte des résultats de l'étudiant aux exercices précédents, de la durée de la séance et du choix d'une compétence à travailler. L'originalité d'AI-VT réside dans la résolution d'une tension entre répétition (entraînement) et variété. AI-VT peut en effet générer des exercices identiques (bien résolus mais sur un temps trop long ou résolus avec une note moyenne) sur plusieurs séances tout en garantissant une variété d'exercices dans les listes portant sur une même compétence travaillée.

2 Problématique et verrous scientifiques

La répétition est un outil incontournable de l'apprentissage par compétences [2]. Toutefois, de nombreux travaux montrent que la répétitivité doit être agrémentée de variabilité [3] [4] [5]. Enseigner par entraînement ne peut donc pas être réduit au seul fait de proposer des exercices associés à une compétence ciblée. Ainsi, la personnalisation pour un entraînement, portera à la fois sur le choix et sur l'ordre des exercices dans une liste. D'une part, les exercices les plus fondamentaux (par apport à la sous-compétence) et les plus simples (par rapport à la difficulté) sont proposés en premier. D'autre part, dans la mesure du possible les exercices d'un même thème (ex : géométrie) sont regroupés (pour proposer une sorte de « fil rouge »). Pour ce faire, l'ordre prend en compte une distance entre les exercices stockée dans la base de connaissances. Le système AI-VT est ainsi articulé autour des compétences (et sous-compétences) ciblées, selon une logique d'entraînement propice à l'acquisition de compétences présentée dans la section précédente. L'une des difficultés supplémentaires réside dans le fait qu'il n'existe pas de rapport bijectif entre problème et solution : l'acquisition d'une compétence pouvant être traitée de plusieurs manières différentes (plusieurs listes d'exercices possibles).

Il existe des logiciels de personnalisation pour l'apprentissage qui intègrent des stratégies pédagogiques [6], des connaissances sur les réponses aux exercices, d'aides à la résolution, des profils d'apprenant [7], une pédagogie par l'erreur [8]. Les deux principaux critères à la conception du système AI-VT ont été (i) de générer des listes de questions pour un apprentissage par entraînements et (ii) de proposer un système peu couteux en investissement pour l'enseignant.

3. Architecture et génération des listes d'exercices

Dans AI-VT, une question est composée d'un contexte et d'un énoncé. Par exemple, le contexte d'une question peut être « On souhaite calculer l'aire d'un cercle » et son énoncé « Quelles sont les variables et les constantes ? ».

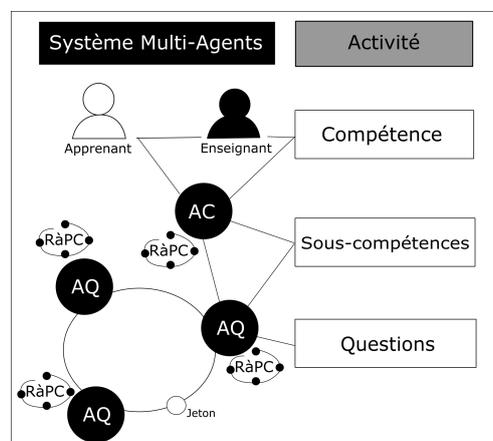


Fig. 1. L'architecture à base d'agents d'AI-VT.

Nous avons choisi de modéliser AI-VT par un Système Multi-Agents (SMA), dont l'architecture est représentée sur la figure 1. Le système est composé de quatre types d'agents : l'enseignant, l'apprenant, l'Agent Compétence (AC) et l'Agent Question (AQ). AC choisit les sous-compétences en regard de la compétence ciblée, et les AQ choisissent les questions en regard des sous-compétences. L'AC utilise le RàPC pour choisir les sous-compétences et envoie ensuite cette liste à un AQ. Les AQ collaborent pour sélectionner la liste de questions, chaque AQ choisissant les questions de l'une des sous-compétences. Ils sont ainsi organisés selon un anneau logique et ils ont besoin de partager leurs listes respectives de questions afin de détecter les conflits et de les arbitrer (lorsqu'une même question est choisie par plusieurs AQ). La cohérence des informations partagées est assurée par un jeton sur lequel les agents reportent les modifications qu'ils ont effectuées depuis leur dernière réception du jeton ainsi que les demandes d'autorisation de modification des informations partagées.

Après réception de la sous-compétence qui lui a été attribuée, chaque AQ sélectionne un ensemble de questions par RàPC. Durant la phase de remémoration, la similarité entre chaque cas source et le cas cible de chaque AQ. La phase d'adaptation répond à la logique d'ordonnancement pour l'ensemble des questions

proposées. Les questions sont sélectionnées dont leur *DPC* (Durée de Pratique Capitalisée) : si deux agents sélectionnent la même question, celui dont la *DPC* est la plus faible doit changer. La question et sa sous-compétence la moins complexe sont ensuite placées en premier. Des permutations sont finalement effectuées entre les questions d'une même sous-compétence afin de minimiser les distances entre les questions consécutives (« fil rouge »). Durant la phase de révision, l'enseignant et l'apprenant évaluent les résultats et la durée passée par l'apprenant sur chaque question. Durant la phase de capitalisation, les *DPC* sont mises à jour pour toutes les questions en fonction des durées effectivement passées par l'apprenant et en fonction d'une note évaluant la réponse de l'apprenant. Cette note sur 10 est donnée par l'apprenant en concertation avec l'enseignant.

4. Conclusion et perspectives

Le système AI-VT est actuellement testé dans le cadre d'un enseignement en « bases de l'algorithmique » pour des étudiants de 1ère année de licence. Cette phase de test nous permettra d'une part d'en évaluer les performances, et d'autre part d'avoir un retour des étudiants (en terme d'utilisation, d'ergonomie, de ressenti, ...). Pour ce faire, un questionnaire est rempli par les étudiants à l'issue de chaque séance.

Bien entendu, pour que le système propose des listes de questions variées, la base de données d'exercices prédéfinie par l'enseignant, doit être suffisamment fournie. L'enseignant doit par ailleurs avoir un découpage par compétences des énoncés et des questions, bien structurés par sous-compétences. En effet, le générateur d'exercices n'est pas automatisé comme dans d'autres systèmes existants tels que *Alfalex* [9] ou *Aplusix* [10]. Une perspective serait d'étudier la mise en place d'un système qui automatise (partiellement) la rédaction d'exercices d'une même famille (i.e. de sous-compétences visées identiques). Il existe en effet des systèmes qui assistent la rédaction d'exercices comme *ASKER* [11].

Nous envisageons également de compléter le système avec des aides à l'évaluation des algorithmes produits par les étudiants afin que ceux-ci s'autoévaluent plus facilement et s'interrogent sur leurs méthodes de résolution. Nous pourrions par exemple nous inspirer du logiciel *ELISE* [12] pour proposer des plans de solution ou bien une vue générale des solutions.

Références

1. Hoffmann, C., Douady, J., Billon, M., Bonvalot, M., Courtois, F., Lafourcade, P., Le Brun, I., Moraux, E., Rist, C., Soulage, M.F.: Deux approches pour une formation opérante des étudiants de l'Université Joseph Fourier (Grenoble, France) aux méthodes de travail universitaire.
2. Beckers, J.: Est-il possible de faire de la pédagogie par compétences un allié de l'équité à l'école. *Cah. Serv. Pédagogie Expérimentale* 2122. (2005) 41–63.
3. Magill, R.A., Hall, K.G.: A review of the contextual interference effect in motor skill acquisition. *Hum. Mov. Sci.* 9, (1990) 241–289.
4. Hall, K.G., Magill, R.A.: Variability of practice and contextual interference in motor skill learning. *J. Mot. Behav.* 27, (1995) 299–309.
5. Davids, K., Button, C., Araujo, D., Renshaw, I., Hristovski, R.: Movement models from sports provide representative task constraints for studying adaptive behavior in human movement systems, (2006).
6. Mandin, S., Guin, N., Lefevre, M.: Modèle de personnalisation de l'apprentissage pour un EIAH fondé sur un référentiel de compétences. In: 7ème Conférence sur les Environnements Informatiques pour l'Apprentissage Humain-EIAH'2015 (2015).
7. Nogry, S., Jean-Daubias, S., Guin, N.: How to combine objectives and methods of evaluation in iterative ILE design: lessons learned from designing *Ambre-add*. *Interact. Learn. Environ.* (2011).
8. Auzende, O., Giroire, H., Calvez, F.L.: Using Competencies to Search for Suitable Exercises. In: 2009 Ninth IEEE International Conference on Advanced Learning Technologies, (2009) 661–665.
9. Selva, T., Verlinde, S., Binon, J.: ALFALEX, un environnement d'aide à l'apprentissage lexical du français langue étrangère. Presented at the Technologies de l'Information et de la Connaissance dans l'Enseignement Supérieur et de l'Industrie October 21 (2004) 515-522.
10. Bouhineau, D., Nicaud, J.-F., Chaachoua, H., Bittar, M., Bronner, A.: Two Years Of Use Of The Aplusix System. Presented at the 8th IFIP World Conference on Computer in Education (2005).
11. Lefevre, M., Guin, N., Cablé, B., Buffa, B.: ASKER: un outil auteur pour la création d'exercices d'auto-évaluation. In: Atelier EAIE (Évaluation des Apprentissages et Environnements Informatiques)-Conférence EIAH 2015 (2015).
12. Delozanne, E., Vivet, M.: Explications en EIAO. Etude à partir d'Elise, un logiciel pour s'entraîner à une méthode de calcul de primitives = Explanations in a learning environment: studies about Elise, a software for the training of students in the use of a calculus method, <http://cat.inist.fr/?aModele=afficheN&cpsid=147290>.