



## Un outil auteur pour une génération semi-automatique d'exercices d'auto-évaluation

Baptiste Cablé\*, Nathalie Guin\*, Marie Lefevre\*

\* Université de Lyon, CNRS  
Université Lyon 1, LIRIS, UMR5205, F-69622, France  
{baptiste.cable, nathalie.guin, marie.lefevre}@liris.cnrs.fr

---

*RÉSUMÉ.* Cet article présente un générateur semi-automatique d'exercices d'auto-évaluation. Ce travail s'inscrit dans le cadre du projet CLAIRE dont l'objectif est la conception d'une plateforme d'édition collaborative de contenus pédagogiques. Le générateur proposé permet à l'auteur (généralement un enseignant) de créer un modèle d'exercice selon ses choix pédagogiques. Ce modèle est ensuite automatiquement instancié pour donner lieu à un grand nombre d'exercices différents évaluant tous les mêmes compétences. Les réponses aux exercices peuvent être évaluées automatiquement et instantanément par le système, ce qui permet à l'apprenant d'avoir un retour immédiat sur son niveau de maîtrise. La particularité de ce générateur est de proposer des types d'exercices indépendants du domaine, ce qui permet de l'utiliser à différents niveaux scolaires et dans de nombreuses disciplines. Par ailleurs, l'utilisation de connaissances du domaine permet d'alléger la charge de travail nécessaire à l'auteur pour définir les modèles d'exercices et les connaissances de diagnostic associées.

*MOTS-CLÉS:* génération semi-automatique d'exercices, outil auteur, auto-évaluation, diagnostic automatique.

---

## 1. Introduction

Au fur et à mesure de l'avancement des technologies de l'information et de la communication pour l'enseignement (TICE), il est de plus en plus fréquent que des apprenants travaillent en autonomie à l'aide de contenus pédagogiques en ligne. Afin de les supporter dans cet apprentissage, nous souhaitons offrir à l'auteur (généralement un enseignant) la possibilité d'ajouter facilement au contenu de cours des exercices d'auto-évaluation à diagnostic automatique. Ces exercices permettent à l'apprenant de tester en autonomie son niveau de maîtrise de ce qu'il a appris dans le cours.

Les apprenants peuvent échouer lors des premières tentatives de réponse aux exercices si les connaissances ne sont pas maîtrisées. Il est donc possible qu'un apprenant soit amené à répondre plusieurs fois à un même exercice avant d'obtenir un succès. Afin que l'apprenant ne se laisse pas influencer par ses précédentes résolutions, il est nécessaire que l'exercice d'auto-évaluation soit différent d'une fois sur l'autre, tout en évaluant la même connaissance. Toutefois, il ne semble pas raisonnable de demander à l'auteur de rédiger de nombreuses versions de l'énoncé. Nous proposons donc de recourir à un générateur d'exercices que l'auteur puisse facilement utiliser quel que soit le domaine.

Différents générateurs d'exercices existent déjà mais aucun ne dispose à la fois de toutes les propriétés que nous souhaitons : l'exercice est différent d'une fois sur l'autre ; l'auteur est maître du contenu de l'exercice et a la garantie qu'il correspond à ses attentes en matière de contenu pédagogique ; le générateur d'exercices est utilisable dans un grand nombre de domaines et de niveaux scolaires ; le diagnostic de la réponse est fait automatiquement et en temps réel ; la construction d'un exercice ne prend pas un temps démesuré à l'auteur ; la création d'exercices ne demande pas de compétences techniques particulières.

Cet article décrit la solution proposée et son implémentation dans le cadre d'une plateforme de rédaction collaborative de contenus d'enseignement. Cette solution s'appuie sur l'approche GEPPETOp [LEVEVRE 10] et consiste en un générateur semi-automatique d'exercices à diagnostic automatique où l'auteur crée un modèle d'exercice qui est ensuite automatiquement instancié pour donner lieu à un grand nombre d'exercices et à leurs diagnostics. Cette proposition ne fait pas référence à un modèle théorique de l'apprenant car aucun n'existe actuellement dans les contextes applicatifs concernés.

L'article est structuré de la manière suivante : le contexte applicatif est présenté dans la section 2. La section 3 est un état de l'art des générateurs d'exercices. La quatrième section présente l'approche GEPPETOp sur laquelle s'appuient nos travaux. La cinquième section décrit la proposition que nous implémentons et un exemple de scénario d'utilisation est donné dans la sixième section. Nous achevons par une conclusion et les perspectives de travail en section 7.

## 2. Contexte : le projet CLAIRE

Le projet CLAIRE (Community Learning through Adaptive and Interactive multichannel Resources for Education) est mené par l'entreprise Simple IT (gestionnaire du Site du Zéro, [SDZ]) en collaboration avec le LIRIS (Lyon) et l'INRIA (Grenoble). Il a pour objet la création d'une plateforme de rédaction collaborative open-source pour le monde de l'éducation. En plus des dernières avancées dans ce domaine, elle dispose d'un générateur d'exercices d'auto-évaluation permettant l'exécution et le diagnostic des exercices. L'objet du travail présenté dans cet article est la création de ce générateur d'exercices.

La plateforme CLAIRE est destinée à plusieurs contextes d'utilisation. Entre autres, elle sera déployée dans des contextes post-bac (universités, écoles d'ingénieur) et pour le Site du Zéro dont la toute dernière version s'appuiera sur CLAIRE. Dans les deux cas, les auteurs auront la possibilité de joindre à leurs contenus de cours des exercices d'auto-évaluation.

En contexte post-bac, ces exercices seront destinés à des étudiants. Ils leur permettront d'appliquer les connaissances de la partie de cours concernée et de vérifier qu'elles sont maîtrisées. Les domaines d'étude sont très variés, allant des langues au droit, en passant par les sciences. Les auteurs des exercices seront alors des enseignants, généralement ceux qui rédigent les cours. Ils n'ont pas nécessairement de compétences en informatique.

Sur le Site du Zéro, le contenu est rédigé sous forme de tutoriels par des membres expérimentés appelés auteurs. Le public est composé de personnes de tous âges et de toutes formations dont le point commun est de souhaiter apprendre un sujet lié à l'informatique en partant de zéro. Pour se former, les apprenants suivent le tutoriel mais n'ont pas d'interaction avec un enseignant pour les guider ou évaluer leur niveau. Des exercices d'auto-évaluation sont actuellement proposés à la fin de chaque chapitre sous la forme de questionnaires à choix multiples (QCM). Cela permet à l'apprenant de voir s'il a compris et retenu les connaissances essentielles du chapitre. Les QCM sont rédigés par l'auteur du tutoriel.

Sur la plateforme CLAIRE, nous souhaitons proposer différents types d'exercice<sup>1</sup>. L'objectif est d'offrir du choix et de la liberté aux auteurs dans la création des exercices tout en proposant des types d'exercice indépendants du domaine. Après avoir choisi un type d'exercice, l'auteur pourra créer un modèle d'exercice qui décrit le contenu et la forme de l'exercice, mais sans forcément le contraindre totalement. À l'aide de ce modèle, un générateur d'exercices sera capable de mettre à disposition de l'apprenant un grand nombre d'exercices différents évaluant la même compétence. Chaque instance d'exercice ainsi générée sera interactive : l'apprenant y répondra en ligne et obtiendra un diagnostic de sa réponse.

### 3. État de l'art sur les générateurs d'exercices

De nombreux générateurs d'exercices existent dans la littérature. Si l'on considère la place de l'auteur dans le processus de création des exercices et la capacité du système à générer automatiquement des variantes d'un exercice, on peut répartir les générateurs en trois grandes familles dont la première regroupe les **générateurs automatiques**. Avec ce genre de générateur, un grand nombre d'exercices est créé automatiquement sans que l'auteur puisse influencer les choix du système. Il peut simplement choisir la catégorie de l'exercice (forme, thème, connaissances abordées) mais ne peut pas agir sur le contenu ni sur des contraintes précises. Un exemple d'un tel générateur se trouve dans le projet ALFALEX où Selva propose une solution pour créer automatiquement des exercices à trous contextuels pour l'apprentissage du français langue étrangère [SELVA 02]. Ce générateur s'appuie sur un dictionnaire et sur un corpus de texte comme ressources. Dans le système Reading Tutor [MOSTOW 04], un générateur automatique d'exercices propose des questions de compréhension au fur et à mesure qu'un apprenant lit un texte. Le micromonde APLUSIX [BOUHINEAU 06], destiné à l'apprentissage de l'algèbre, contient également un

---

<sup>1</sup> Nous appelons « type d'exercice » la nature de l'exercice. On trouve, par exemple, les types suivants : « vrai ou faux », « texte à trous » ou « traduction ». Le type d'exercice est, *a priori*, indépendant du domaine. Cependant, certains types peuvent y être intimement liés, comme par exemple « diagonalisation de matrice » qui est un type uniquement utilisable dans des contextes comprenant des matrices.

générateur automatique d'exercices. Un très grand nombre de modèles d'exercice a été codé dans cet EIAH, ce qui laisse une certaine liberté dans le choix de l'activité à générer automatiquement, mais il est impossible pour l'auteur de paramétrer précisément un de ces modèles.

Prenons un exemple concret dans notre contexte : un auteur veut créer un exercice d'auto-évaluation pour évaluer la maîtrise des mots-clés du langage de programmation C. Il souhaite pour cela créer des textes à trous à partir de programmes en C où les trous porteraient sur les mots-clés. S'il dispose d'un générateur automatique permettant de créer de tels exercices, l'auteur pourra facilement en générer un très grand nombre. Par contre, il n'aura pas la possibilité de choisir lui-même les textes (programmes C) sur lesquels porteront les exercices. Selon la finesse du générateur, il pourra ou ne pourra pas préciser les mots-clés sur lesquels il souhaite ou ne souhaite pas travailler. Dans tous les cas, il n'aura pas une totale liberté sur le contenu de l'exercice.

Ce type de générateur répond donc à notre besoin de pouvoir créer un grand nombre d'exercices évaluant une même connaissance. Ainsi, un apprenant qui échoue peut recommencer l'exercice sans être influencé par la résolution d'une instance précédente de l'exercice. Toutefois, le projet CLAIRE requiert que les exercices soient en accord avec les choix pédagogiques de l'auteur. Il est donc nécessaire que ce dernier puisse choisir précisément le contenu de ses exercices. Les générateurs automatiques ne sont donc pas adaptés car ils n'offrent pas cette flexibilité.

La deuxième grande famille de générateurs d'exercices est celle des **générateurs manuels**. Souvent utilisés dans le cadre des outils auteurs, ces générateurs laissent une grande liberté à l'auteur. Ce dernier définit précisément le contenu de l'exercice ainsi que toutes les options de mise en forme. Un tel générateur se trouve par exemple dans le projet ARIADNE avec l'outil auteur GenEval [COGNE 98] où les exercices créés sont des hypermédias. La résolution de l'exercice se fait sur papier puis l'apprenant compare sa réponse avec celle donnée par le système. La solution peut aussi n'être que partiellement affichée afin de simplement constituer une aide à la résolution. Certains outils auteurs issus du commerce tels que Articulate Quizmaker [AQM] ou Hot Potatoes [HOT POTATOES] sont couramment utilisés pour créer, de manière assistée, des exercices papier ou informatisés. Les plateformes d'apprentissage en ligne Claroline [CLAROLINE] et Moodle [MOODLE], couramment utilisées dans l'enseignement supérieur, proposent également leurs propres outils d'édition d'exercices. Les types d'exercice proposés sont le texte à trou, le QCM, le vrai ou faux et l'appariement d'éléments. Il s'agit ici aussi de générateurs manuels où les questions doivent être créées mot par mot par l'auteur.

Si l'on reprend l'exemple des textes à trous sur des programmes en C, avec un générateur manuel, l'auteur devra créer un à un ses textes à trous. Pour chaque instance, il choisit un programme C puis indique les trous un à un. S'il veut proposer 25 instances différentes de texte à trous, il devra répéter 25 fois ce processus.

Avec ce type de générateur, l'auteur est assuré d'avoir un exercice correspondant précisément à ses attentes. Cela répond à l'un de nos besoins. Par contre, l'auteur doit créer chaque instance d'exercice une par une en précisant son contenu. Ce type de générateur n'est pas capable de créer automatiquement un grand nombre d'exercices évaluant la même compétence. On ne peut donc pas utiliser un générateur manuel d'exercices pour répondre à notre besoin d'un grand nombre d'exercices interactifs d'auto-évaluation.

Les **générateurs semi-automatiques** d'exercices combinent les avantages des deux catégories précédentes. Ceux-ci proposent à l'auteur de définir un modèle d'exercice qui est ensuite instancié pour donner lieu à un grand nombre d'exercices. Le générateur PepiGen [PREVIT 07] crée des exercices en déclinant des modèles paramétrés d'exercice. Ces modèles sont créés par un informaticien à partir de spécifications établies par un spécialiste

en didactique. Tous les exercices issus d'un même modèle paramétré évaluent la même compétence. Le modèle est créé par un informaticien et est lié au domaine, ce qui par rapport à nos besoins représente deux faiblesses. Nous souhaitons que les modèles d'exercice puissent être créés par des auteurs, quelque soit leur discipline. Le générateur GenAmbre [DUCLOSSON 05] permet la création d'exercices pour l'apprentissage de méthodes, essentiellement dans des domaines scientifiques. Ces exercices sont générés en fonction des contraintes définies par l'auteur. Si ce dernier n'a rien précisé, l'exercice est généré de manière totalement aléatoire. S'il a précisé un grand nombre de paramètres, la génération est fortement contrainte. Ici encore, pour chaque nouveau domaine où l'on souhaite utiliser le générateur, un grand nombre de connaissances doit être décrit par un informaticien. De plus, le type de problèmes que peut générer GenAmbre est très précis. Il est par exemple impossible de générer un exercice d'appariement d'images. Ces premiers générateurs semi-automatiques ne répondent pas à nos besoins car ils sont conçus pour des applications ou domaines spécifiques.

Moodle propose, en plus de ses générateurs manuels, le type d'exercice *question calculée*. Ce type de question numérique contient des paramètres tels que  $\{x\}$ ,  $\{y\}$  qui sont remplacés par des valeurs tirées au hasard lorsque le test est passé par un étudiant. Le générateur lié à ce type de question est donc semi-automatique. Ce type de générateur d'exercices est également proposé par WIMS [WIMS] et par EULER [EULER] pour des activités pédagogiques dans le domaine des mathématiques. Des travaux de proposition d'extension à IMS-QTI [AUZENDE 07] s'intéressent également à ce type d'exercice afin de traiter le cas de variables interdépendantes. Ce type d'exercice répond partiellement à notre besoin et nous l'implémentons dans notre proposition. Toutefois, il se limite aux domaines nécessitant du calcul, or nous souhaitons des générateurs semi-automatiques adaptés à de nombreux domaines et incluant d'autres types d'exercice.

De manière générale, les travaux les plus pertinents par rapport à nos besoins sont ceux de Lefevre [LEFEVRE 10] qui propose l'approche GEPPEToP qui permet, entre autres, de définir et générer des exercices de manière semi-automatique. Les générateurs qui y sont décrits répondent particulièrement à nos attentes car ils permettent de créer des exercices de plusieurs types et indépendamment du domaine d'enseignement. Nous discutons dans la section suivante de la manière d'utiliser cette approche dans notre contexte.

## 4. L'approche GEPPEToP

### 4.1. Principe de l'approche

L'approche GEPPEToP (GENeric models and Processes to Personalize learners' PEDagogical activities according to Teaching Objectives – Paper) [LEFEVRE 10] propose des modèles et processus pour personnaliser les activités pédagogiques en fonction des souhaits de l'auteur. Pour créer son activité papier, l'auteur ne va pas directement travailler sur l'instance finale de l'exercice mais sur un modèle de l'exercice ici appelé *structure d'exercices*.

Un patron d'exercice (Figure 1) contient des connaissances relatives à un type d'exercice. Ce méta-modèle décrit toutes les informations que doit contenir une structure d'exercices. Chaque type d'exercice dispose de son propre patron. Dans notre exemple de texte à trous portant sur des mots-clés de programmes en C, le patron d'exercice « texte à trous » pourrait être utilisé.

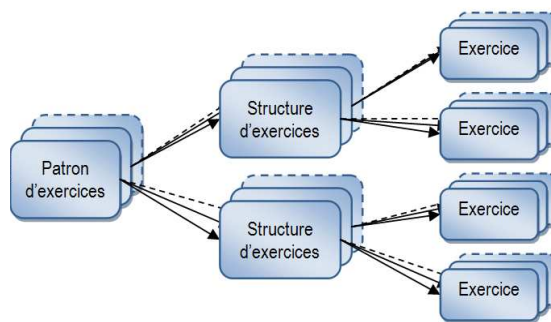


Figure 1. L'approche GEPPEToP [LEFEVRE 10]

Le patron d'exercice est instancié par l'auteur pour créer une structure d'exercices. Elle contient l'information permettant de créer des exercices : on y trouve essentiellement la déclaration des ressources sur lesquelles l'exercice se fonde, la consigne de l'exercice et les préférences de l'auteur en matière de présentation de l'exercice. Dans notre exemple, définir la structure d'exercices consisterait à préciser les informations suivantes : « En ressource, je veux un programme C de longueur comprise entre 100 et 200 mots, de difficulté facile ou moyenne. Dans le texte ainsi choisi, je veux supprimer 6 mots parmi *for*, *while*, *if*, *then*, *else*, *case*, *end*. Je veux que la liste des mots supprimés soit affichée à l'apprenant dans l'ordre alphabétique ».

La structure d'exercices est exploitée par le générateur d'exercices associé au patron qui crée alors des instances d'exercice. L'instance d'exercice, appelée exercice (Fig. 1), consiste en la version finalisée de l'activité telle qu'elle est proposée à l'apprenant. L'exercice inclut également une correction (pour l'auteur enseignant). Dans notre exemple, l'exercice est un texte présentant un programme C dans lequel des mots-clés ont été enlevés. L'exercice précise la liste de ces mots classés dans l'ordre alphabétique. Il contient également la correction ; il s'agit du texte d'origine.

Dans cette approche, l'auteur dispose donc d'un cadre précis pour construire ses structures d'exercice mais il dispose d'une grande liberté sur le contenu de ses exercices. De plus, s'agissant d'un générateur semi-automatique, un grand nombre d'exercices différents peut être créé à partir d'une même structure.

#### 4.2. Limites de l'approche dans notre contexte

L'approche GEPPEToP est fondée sur l'étude d'un corpus d'exercices allant de l'école primaire à l'université. L'approche revendique donc de couvrir la génération d'exercices pour tous les niveaux scolaires (primaire, secondaire et supérieur). Toutefois, les types d'exercice proposés s'apparentent à des activités que l'on retrouve essentiellement à l'école primaire, parfois au collège. Nous nous interrogeons donc sur la possibilité de suivre cette approche dans notre contexte applicatif post-bac et sur le Site du zéro. Sur le Site du zéro, les exercices d'auto-évaluation ont pour but de permettre au lecteur de s'assurer qu'il a compris et retenu l'essentiel du chapitre du tutoriel. Il ne s'agit donc pas de proposer des exercices extrêmement techniques ni spécialisés. Des exercices génériques tels que des textes à trous, des ordonnancements ou des appariements d'éléments sont donc adéquats. Dans l'enseignement supérieur, les domaines étudiés sont très variés et l'on trouve pour chacun d'entre eux des exercices spécialisés dont la résolution et la création requièrent un très grand nombre de connaissances du domaine. Chacun de ces types d'exercice spécialisés

nécessite presque un générateur à lui seul, tant la démarche est spécifique. Ces exercices sortent donc de notre problématique car nous nous intéressons à des exercices génériques pouvant être employés dans de nombreux domaines pour de l'auto-évaluation. Même si l'usage veut que, dans l'enseignement supérieur, les connaissances techniques d'un domaine soient travaillées et évaluées à l'aide d'exercices spécialisés, les exercices génériques (textes à trous, appariement, etc.) pourraient permettre également une évaluation efficace. L'une des différences entre les exercices spécialisés et génériques est que l'exercice spécialisé permet, habituellement, de guider l'apprenant lors de la résolution d'un problème complexe. Avec un exercice générique, l'apprenant n'est pas guidé et seule la réponse est attendue. Il doit donc effectuer la démarche de résolution de manière autonome. Bien que cela ne soit pas pertinent dans un contexte d'entraînement, cela semble toutefois adapté en contexte d'auto-évaluation car cela permet de voir si la compétence est acquise. Un exemple de ce type d'exercice générique est le QCM qui est un classique lors des processus d'évaluation. Nous proposons les exercices suivants comme moyens d'auto-évaluation indépendants du domaine :

- identification de parties du texte (inclut le texte à trous)
- organisation d'éléments (ordonnancement, groupement, appariement)
- annotation d'illustration
- questionnaire à choix multiple (QCM)
- questionnaire à réponses ouvertes et courtes (QROC).

Ces types d'exercice sont tous proposés dans l'approche GEPPEOp. Ainsi, les types d'exercice qu'elle propose semblent adaptés à tous les niveaux scolaires dans le cadre d'évaluation (ou auto-évaluation) et en particulier à ceux du projet CLAIRE, à savoir le Site du Zéro et l'enseignement supérieur.

L'approche GEPPEOp est prévue pour produire des exercices papier. Dans notre contexte, nous souhaitons produire des exercices informatisés où l'apprenant répond en ligne et obtient un diagnostic automatique de sa réponse. Ce diagnostic est plus complexe pour des exercices informatisés. En effet, pour la version papier, il suffit de produire une solution juste pour l'auteur. Dans le cas d'exercices où plusieurs solutions (ou plusieurs syntaxes) sont justes, l'auteur fait l'effort d'adaptation de la solution pour estimer si la réponse de l'apprenant est juste. Pour des exercices informatisés à diagnostic automatique, il n'est pas suffisant de produire une correction juste : il faut pouvoir reconnaître toutes les bonnes réponses. L'approche GEPPEOp ne propose pas de solution pour reconnaître toutes les bonnes réponses. De manière générale, l'approche GEPPEOp n'est pas prévue pour des exercices informatisés à diagnostic automatique.

## 5. Proposition

GEPPEOp décrit des outils et processus pour la création d'activités selon les souhaits de l'auteur. Nos travaux consistent à nous appuyer sur cette approche pour (1) proposer un générateur semi-automatique d'exercices d'auto-évaluation interactifs pour la plateforme CLAIRE, (2) proposer une solution de diagnostic automatique de la réponse de l'apprenant, et (3) proposer une solution d'aide aux auteurs, lors de la création d'exercices, fondée sur les connaissances du domaine. Nous nous appuyons sur les niveaux de représentation des exercices employés dans GEPPEOp. Nous renommons le patron d'exercice en *connaissances sur le type d'exercice* et la structure d'exercices en *modèle d'exercice*. Les types d'exercice que nous implémentons sont ceux listés dans la section précédente auxquels nous ajoutons la *question calculée*. Cette liste n'est pas exhaustive et une personne ayant les compétences peut ajouter un type d'exercice répondant à de nouveaux besoins.



### 5.1. Architecture

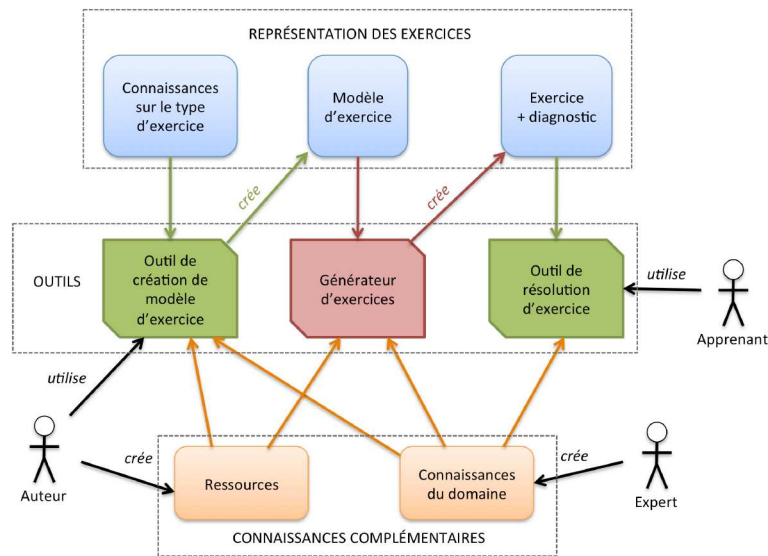


Figure 2. Architecture de notre approche

La figure 2 présente l'architecture de la solution que nous proposons. Le bloc du haut est constitué des différents niveaux de représentation des exercices que nous avons décrits en nous inspirant de l'approche GEPPETOp. Dans le bloc central se trouvent les trois mécanismes manipulant ces représentations des exercices. Le bloc inférieur contient les ressources et connaissances utilisées dans le processus de création des exercices.

Les ressources sont les éléments de base qui sont utilisés pour construire les exercices. Il s'agit, par exemple, de textes, d'images ou de séquences<sup>2</sup>. Chaque ressource dispose de métadonnées la caractérisant ainsi que de métadonnées enrichissant la ressource. Il s'agit par exemple de la légende d'une image ou d'annotations sur des zones de l'image.

Les connaissances du domaine (cf. 5.2) sont des connaissances relatives à une matière et indépendantes du type d'exercice. Il s'agit par exemple de la liste des mots-clés du langage C. Ces connaissances sont définies par un expert.

L'auteur crée le modèle d'exercice à l'aide d'un outil dédié qui s'appuie sur les connaissances sur le type d'exercice. Cet outil de création de modèles d'exercice facilite la création du modèle, notamment en s'appuyant sur les connaissances du domaine. Il permet également de générer quelques instances d'exercice pour tester si le modèle donne bien lieu aux exercices attendus. Cet outil facilite l'accès et le choix des ressources. Il permet même à l'auteur d'en créer de nouvelles.

Le générateur reçoit en entrée un modèle d'exercice qu'il instancie pour donner lieu à un exercice en sortie. L'exercice est conforme au modèle et donc aux choix de l'auteur qui l'a créé. Le générateur ne nécessite aucune intervention humaine. Il dispose de toutes les

<sup>2</sup> Cette ressource est une liste ordonnée d'éléments pouvant être utilisée pour construire un exercice d'ordonnement d'éléments.

informations nécessaires dans le modèle d'exercice et fait appel, si nécessaire, aux ressources et aux connaissances du domaine. Il est activé à chaque fois que l'on veut obtenir une nouvelle instance d'exercice (incluant le diagnostic) à partir d'un modèle.

L'exercice est présenté à l'apprenant par un outil de résolution d'exercice. Ce dernier met en forme l'exercice, recueille la réponse de l'apprenant et en fournit le diagnostic.

### 5.2. Connaissances du domaine

D'une manière générale, les connaissances du domaine sont des connaissances directement liées au domaine et indépendantes du type d'exercice dans lequel elles sont utilisées. Il s'agit par exemple des mots-clés du langage C, d'une méthode pour détecter le genre d'un nom, de la valeur d'une constante, etc. Elles peuvent prendre différentes formes selon le type de connaissance, dont valeur constante, formule de calcul, règle, énumération. Elles sont saisies manuellement par un expert du domaine à l'aide d'un outil dédié (qui n'apparaît pas sur la Figure 2 pour l'alléger).

Ces connaissances sont utilisées à deux niveaux. D'une part, elles simplifient la création des modèles d'exercice car il est possible d'utiliser ces connaissances plutôt que de les redéfinir pour chaque exercice du même domaine. Dans notre exemple, elles permettent de demander : « crée des trous sur les mots-clés du C » plutôt que : « crée des trous sur les mots *for*, *while*, *if*, *then*, *else*, *case*, *end* ». En plus de simplifier la création du modèle d'exercice, cela limite les risques d'erreur de la part de l'auteur car en saisissant l'énumération, il peut en oublier ou faire plus facilement des erreurs. D'autre part, ces connaissances peuvent être utilisées lors du diagnostic de la réponse de l'apprenant.

La création des connaissances du domaine est une tâche indépendante de la création de modèles d'exercice et tous les auteurs ne souhaiteront pas s'y atteler. Cette mission est menée à bien par un expert qui a la connaissance du domaine ainsi que la maîtrise technique nécessaire pour créer des connaissances. Cet expert peut être (et sera souvent) un auteur.

### 5.3. Diagnostic

La principale différence entre le contexte de l'approche GEPPETOp et notre implémentation est l'aspect informatisé des exercices générés. L'apprenant résout ces exercices en ligne et obtient un diagnostic automatique de sa réponse. Dans le cas d'un exercice où la réponse est unique, il suffit que cette solution soit intégrée à l'exercice (mais invisible pour l'apprenant, bien entendu). Le diagnostic consiste alors à analyser la réponse de l'apprenant pour voir si elle est identique à l'unique bonne réponse ou si il s'agit d'une réponse provoquant un feedback particulier. Dans le cas où plusieurs bonnes réponses sont possibles, et ce parfois en très grand nombre, cette solution n'est pas envisageable : il n'est parfois pas possible de préciser dans l'exercice toutes les réponses possibles. Dans notre approche, cette difficulté est gérée à deux niveaux.

Les problèmes simples d'ordre syntaxique sont traités au niveau de l'interface. Par exemple, si l'étudiant répond 3.14 au lieu de 3,14 (ou 3E4 au lieu de  $3 \cdot 10^4$ ), ce n'est pas au niveau du contenu de l'exercice que l'équivalence de ces deux réponses est précisée. L'interface fait automatiquement cette conversion. Pour tous les autres problèmes, majoritairement d'ordre sémantique, la solution générée avec l'exercice est en réalité un modèle de solution qui rend plusieurs réponses acceptables. Ces informations sur les réponses acceptables proviennent soit du modèle d'exercice où l'auteur a précisé les variations et tolérances autour de la réponse, soit des ressources employées où les différentes solutions associées sont précisées.



Figure 3. Solution d'exercice à plusieurs variantes

À titre d'exemple, les ressources de type *séquence* utilisées pour les exercices d'ordonnement contiennent des informations sur les blocs interchangeables qui ne faussent pas la solution (Figure 3). Dans cet exemple, les blocs CD, E, FGH et I peuvent être permutés au sein du grand bloc gris pour donner une solution juste. Cela évite de devoir lister les 24 solutions justes. Cette méthode est implémentée pour tous les exercices d'ordonnement, par exemple les exercices consistant à remettre dans l'ordre un programme C où les lignes ont été mélangées. En effet, il peut exister plusieurs solutions car certaines lignes sont interchangeables sans pour autant perturber le fonctionnement du programme.

De plus, un expert peut formaliser des connaissances du domaine spécialement dédiées au diagnostic. Comme les autres connaissances du domaine, elles sont communes (non incluses dans un modèle d'exercice) et allègent la tâche de l'auteur. Il pourrait s'agir, par exemple, d'un outil permettant de comparer l'équivalence de deux formules logiques<sup>3</sup>.

## 6. Scénario d'usage

Afin d'illustrer l'ensemble de notre proposition, prenons un exemple d'utilisation du générateur d'exercices de CLAIRE. Supposons qu'un auteur souhaite créer un exercice pour évaluer la capacité à reconnaître visuellement des cellules cancéreuses sur des observations microscopiques. Supposons que l'auteur sache que parmi les ressources disponibles se trouvent des clichés de telles observations et que ces images sont annotées (type de cellules, pathologie observée, stade de la pathologie, etc.). Il choisit son type d'exercice : groupement d'éléments. Il précise ses contraintes sur le choix des ressources : il veut des images d'observations microscopiques de cellules, contenant une information sur la pathologie observée, et avec un facteur de grossissement compris entre x400 et x1000. Il pourrait également, s'il le souhaitait, préciser davantage de contraintes pour avoir un choix d'images plus homogènes. Il définit ensuite les options propres aux exercices de groupement : le groupement doit porter sur la pathologie observée et il veut deux catégories : cancéreuses et saines. Cela restreint automatiquement le choix de ressources en excluant les autres pathologies. L'identification automatique de la catégorie de ces images fait appel aux connaissances du domaine qui, à partir des métadonnées de l'image, déterminent s'il s'agit de cellules saines, cancéreuses ou autre. Le modèle d'exercice ainsi créé peut ensuite être instancié par le générateur pour donner lieu à des exercices. Avant de valider son travail, l'auteur peut générer quelques instances pour voir si le modèle produit des exercices répondant à ses attentes.

Lorsque l'apprenant souhaite s'auto-évaluer, un exercice de groupement est créé et lui est présenté. Il le résout puis demande l'évaluation de sa réponse. Quel que soit son résultat, il peut continuer à s'auto-évaluer en demandant d'autres instances de l'exercice.

Notons que les ressources employées dans cet exemple peuvent être réutilisées dans des exercices de groupement sur d'autres critères ou même dans d'autres types d'exercice.

<sup>3</sup> Il s'agit d'un exemple fictif de connaissance du domaine. Actuellement, l'utilisation des connaissances du domaine pour le diagnostic n'a pas encore été implémentée.

## 7. Conclusion et perspectives

Nous présentons dans cet article le générateur d'exercices d'auto-évaluation de la plateforme CLAIRE. Ce générateur est semi-automatique et crée des exercices informatisés à diagnostic automatique. Il s'agit d'une implémentation basée sur l'approche GEPPEOp proposée par Lefevre [LEFEVRE 10]. Ce générateur permet à un auteur de créer un modèle d'exercice où il décrit avec beaucoup de liberté les ressources à utiliser et les contraintes sur l'activité. En accord avec ces contraintes, le générateur peut créer un grand nombre d'instances d'exercice permettant d'évaluer les mêmes compétences chez l'apprenant. Ce dernier répond à l'exercice *via* une interface informatique et obtient instantanément un diagnostic automatique de sa réponse. Cette solution est un compromis intéressant entre les outils auteurs permettant de générer un exercice respectant précisément les choix de l'auteur et les générateurs automatiques capables de créer un grand nombre d'exercices du même type sur un thème.

Les premiers tests réalisés en laboratoire nous permettent de valider le fonctionnement du générateur d'exercices. Il sera testé par le public du Site du Zéro quand la première version de CLAIRE sera mise en place au printemps 2013. Cela permettra une validation en condition réelle et à plus grande échelle.

Le générateur d'exercices étant indépendant du domaine, les connaissances du domaine n'existent pas au départ. Leur acquisition représente donc un enjeu important. Cette tâche est actuellement effectuée par un expert qui crée manuellement les connaissances du domaine. Cela peut se révéler efficace, mais dans de nombreux cas, il s'agit d'une tâche fastidieuse à laquelle peu de personnes acceptent de se consacrer. L'utilisateur principal de l'outil de création de modèles d'exercice étant l'auteur, il serait intéressant que ce soit lui qui construise les connaissances du domaine. Le plus convivial pour lui serait qu'il crée les connaissances du domaine requises au fur et à mesure qu'il en ait besoin ou, encore mieux, qu'elles soient déduites des informations qu'il renseigne pour créer ses modèles d'exercice. Pour le moment, seule la création manuelle des connaissances est supportée. A l'avenir, nous souhaitons assister l'utilisateur dans la définition de ces connaissances au fur et à mesure de la création des activités.

Par ailleurs, il serait intéressant de coupler la proposition avec des informations sur les apprenants afin d'adapter le contenu des exercices d'auto-évaluation au profil de chaque apprenant.

Enfin, nous souhaitons travailler sur le diagnostic de la réponse de l'apprenant afin d'identifier des misconceptions ou de l'aider à comprendre son erreur. Cela permettrait de guider son apprentissage.

## Bibliographie

- [AUZENDE 07] Auzende, O., Giroire, H., Le Calvez, F., « Propositions d'extensions à IMS-QTI 2.1 pour l'expression de contraintes sur les variables d'exercices mathématiques. », *EIAH*, 2007.
- [BOUHINEAU 06] Bouhineau, D., Nicaud, J.-F., « Aplusix, un EIAH de l'algèbre. », Chapitre 15, *Environnements Informatiques pour l'Apprentissage Humain*, sous la direction de Monique Grandbastien et Jean-Marc Labat, p. 333–350, Hermès, Paris, 2006.
- [COGNE 98] Cogne, A., David, J.-P., Lacombe, C., « Production d'exercices hypermédias et mise en œuvre pédagogique. », *8ième Journée Informatique Et Pédagogie Des Sciences Physiques*, p. 31–39, Montpellier, 1998.
- [DUCLOSSON 05] Duclosson, N., Jean-Daubias, S., Riot, S., « AMBRE-enseignant : un module partenaire de l'enseignant pour créer des problèmes. », *EIAH*, p. 353–358. Montpellier, 2005.

- [LEFEVRE 10] Lefevre, M., « GEPPETO: une approche générique permettant d'adapter les activités des apprenants aux intentions pédagogiques de chaque enseignant. », *RJC EIAH*, 2010.
- [MOSTOW 04] Mostow, J., Beck, J.-E., Bey, J., Cuneo, A., Sison, J., Tobin, B., Valeri, J., « Using automated questions to assess reading comprehension, vocabulary and effects of tutorial interventions. » *Technology, Instruction, Cognition and Learning*, (2), p. 97–134, 2004.
- [PREVIT 07] Prévité, D., Delozanne, E., Grugeon, B., « Génération d'exercices de diagnostic de compétences en algèbre. », *EIAH*, 2007.
- [SELVA 02] Selva, T., « Génération automatique d'exercices contextuels de vocabulaire. », *Traitement Automatique du Langage Naturel*, p. 185–194. Nancy, 2002.

### **Références sur le WEB**

- [AQM] Site web de Articulate, page de Quizmaker.  
<http://www.articulate.com/products/quizmaker.php> (consulté en janvier 2013)
- [CLAROLINE] Documentation Claroline. <http://doc.claroline.net/fr/index.php> (consulté en janvier 2013)
- [EULER] Euler. <http://euler.ac-versailles.fr/> (consulté en janvier 2013)
- [HOT POTATOES] Site web de Hot Potatoes <http://hotpot.uvic.ca/> (consulté en janvier 2013)
- [MOODLE] Documentation de Moodle. <http://docs.moodle.org/2x/fr> (consulté en janvier 2013)
- [SDZ] Le Site du Zéro. <http://www.siteduzero.com/> (consulté en janvier 2013)
- [WIMS] WWW interactive multipurpose server. <http://wims.unice.fr/wims/> (consulté en janvier 2013)