

# Une approche itérative de conception et de mise en œuvre de générateurs de scénarios d'activités

Pierre Laforcade

Laboratoire d'Informatique de Le Mans Université (LIUM)  
Laval, France  
pierre.laforcade@univ-lemans.fr

## 1 Contexte d'adaptation et problématique

Dans le contexte des jeux sérieux d'apprentissage (ou *learning games*) proposant des activités répétitives adaptées à l'apprenant il n'est pas envisageable de concevoir et de réaliser toutes les activités possibles. L'utilisation d'un générateur est primordiale. Selon les critères définis dans [7] notre intérêt s'est porté sur ce besoin spécifique : l'adaptation par génération (technique) de scénarios d'activités (la cible) en fonction du profil de l'apprenant (source 1) et des activités possibles (source 2). Nous considérons ici les scénarios comme la spécification de la ou les activités d'apprentissage que devra réaliser l'apprenant pendant la session de jeu concernée par la génération ; chaque activité étant décrite en terme de configurations des éléments qui seront nécessaires au *player*. L'adaptation qui nous intéresse est initiée par l'apprenant lorsqu'il demande une nouvelle *session* de jeu. Elle est réalisée automatiquement (sans interaction avec l'apprenant ou une autre personne).

Une telle adaptation nécessite une caractérisation fine de la *cible*, des *sources* mais également des *règles* qui seront utilisées durant la génération. Certains éléments *source*, comme la définition du profil de l'apprenant, ont été largement traités dans le passé pour différents type d'EIAH (tuteur intelligent, hypermédia, etc.) [1] selon les points de vue modélisation, mise à jour, acquisition d'informations sur l'apprenant, etc. La modélisation des connaissances sur les objets qui composeront chaque scène de jeu a été également mise en avant [2]. L'adaptation est réalisée de manière *ad hoc* dans la plupart des travaux : les "modèles" des règles d'adaptation sont souvent implicites (dans le code) ou parfois explicites et décrits dans un format informatique (machines à états, règles logiques, etc.). Peu de travaux visent à supporter la modélisation des règles au niveau analyse (décrire l'adaptation, éventuellement avec l'aide d'experts<sup>1</sup>). Aussi, rares sont les travaux cherchant à guider la conception informatique. Les auteurs de [6] abordent cette conception de l'adaptation en proposant une architecture générique pour personnaliser un scénario de jeu sérieux en fonction des compétences de l'apprenant et de ses traces d'interaction. Des techniques de représentation et de manipulation d'ontologies sont utilisées pour la mise en œuvre concrète

---

1. L'expert du jeux d'apprentissage pouvant aussi être un enseignant selon les contextes.

de l'architecture. Toutefois peu d'informations sur la spécification des règles de génération sont disponibles.

La conception d'adaptation nécessite également que les règles d'adaptation soient connues, qu'elles aient déjà fait l'objet d'une description (analyse). Pourtant ceci n'est pas forcément trivial. Certains contextes d'adaptation nécessitent une approche itérative permettant de "tester" différentes adaptations possibles avec l'implication des enseignants/experts, afin de les valider ou de les modifier. Ceci nécessite que les règles d'adaptation identifiées soient mises en oeuvre et exécutées afin de fournir des retours aux experts pour qu'ils puissent réviser et expérimenter d'autres adaptations.

## 2 Proposition et premiers résultats

Notre positionnement général est de chercher à supporter une telle conception de générateurs de scénarios adaptés : une conception permettant de tester les règles d'adaptation pour aider les experts et les informaticiens à développer le générateur. Nous avons proposé pour cela une approche [4] et un processus d'application de cette approche [3]. L'approche consiste à considérer la conception du générateur selon 3 perspectives incrémentales pour les scénarios à générer, 3 dimensions de spécification des éléments du domaine, et 2 niveaux d'abstraction (modèle et méta-modèle). Les trois perspectives (objectifs, structure et caractéristiques) permettent de décomposer le problème de la conception de la génération en différentes étapes successives et incrémentales, chacune appréhendant un point de vue spécifique sur le scénario adapté à générer (adaptées de [6]). Chaque perspective consiste alors à définir sous forme de méta-modèles les 3 catégories d'éléments (éléments à générer, éléments contextuels, éléments décrivant le jeu d'apprentissage). La génération est alors traitée comme un problème de transformation de modèle classique en IDM<sup>2</sup>. Le processus de conception décrit les différentes activités d'analyse et de conception impliquant experts et informaticiens. Certaines activités comme la spécification des méta-modèles et des modèles incombent à l'informaticien mais l'expert n'est pas réduit à intervenir en phase d'analyse. Il participe à la modélisation des cas de tests et à l'analyse du modèle obtenu après la génération. Cette approche permet de se concentrer sur la génération des scénarios indépendamment du développement du jeu d'apprentissage. Elle est également *ouverte* à la prise en compte de nouveaux éléments dans le modèle des connaissances sur le jeu sans nécessiter des modifications sur le fonctionnement du générateur.

La première application de cette proposition a été réalisée dans le contexte du projet *Escape it!* dont l'objectif est de développer un *learning game* pour des enfants autistes. Ce jeu a pour but de renforcer les apprentissages et surtout d'encourager la généralisation des acquisitions en proposant de nombreuses situations variées pour une même compétence. Ces compétences concernent les performances visuelles (au sens du référentiel ABLLS-R) : appariement, tri, catégorisation, etc. Le jeu exploite un *gameplay* minimaliste où le joueur doit trouver

---

2. Ingénierie Dirigée par les Modèles

dans une pièce des objets, parfois cachés, et les déplacer vers des zones "solutions", de manière à débloquent l'ouverture de la porte et ainsi accéder au niveau suivant. Le scénario à générer (cible) correspond alors à une séquence ordonnée de scènes dont la configuration des différents éléments (objets, décors, cachettes, emplacements, etc.) est précisée. L'adaptation doit prendre en compte deux sources. Le profil de l'enfant (progression et difficultés pour les compétences) correspond aux *éléments contextuels* tandis que les informations sur l'arbre des compétences, les objets disponibles pour chaque scène, etc., représentent les *éléments du jeu*. Les règles de génération concernent principalement les relations entre niveaux de difficulté / compétences visées / agencement des scènes. La méta-modélisation capture les relations *structurelles* entre la cible et les différentes sources ainsi qu'entre les sources elles-mêmes mais les aspects *comportementaux* des règles de génération restent à spécifier et traiter informatiquement sous une forme non compréhensible par les experts.

### 3 Verrous et travaux en cours

Les expérimentations réalisées avec les experts autisme lors des différentes sessions de conception collaborative ont permis de vérifier : i/ que les 3 perspectives (à propos du scénario à générer) permettent bien d'aborder la complexité de l'analyse de la génération, ii/ qu'il était possible d'utiliser notre approche et le processus de conception du générateur pour valider / rejeter / modifier / faire émerger des règles d'adaptation. En revanche, il n'était pas possible de leur proposer de valider des règles d'adaptation pendant la session en cours à cause du temps requis pour leur mise en oeuvre.

En effet, les aspects comportementaux des règles étaient traduits sous la forme d'un algorithme procédural codé en Java/EMF. Ce verrou lié au coût/temps de mise en oeuvre des règles dans le générateur nous a amené à aborder la mise en oeuvre des règles de génération selon un nouveau paradigme. Nous avons ensuite considéré la spécification des règles d'adaptations comme un problème de satisfaction de contraintes exploitant les méta-modèles et modèles déjà spécifiés [5]. Cette nouvelle approche permet de minimiser le coût de mise en oeuvre pour l'informaticien. Elle permet également de faciliter la maintenance et la modification rapide des règles lorsque seulement quelques aspects sont modifiés. En revanche, lorsque les règles sont fortement modifiées, ou pour les nouvelles règles, un temps de réalisation différé est toujours nécessaire. Est-il envisageable d'identifier un formalisme qui permettrait de spécifier les règles d'adaptation facilement (*avec* ou *par* les experts métiers) pendant les sessions de conception et que ces règles soient interprétées par le générateur sans nécessité de développement supplémentaire ?

Nous avons également cherché à appliquer notre proposition dans le contexte du projet REVERIES. Celui-ci vise le développement d'une application permettant à tout citoyen de participer à l'inventaire botanique des arbres en France. L'application doit intégrer pour cela un générateur d'activités d'inventaire ludiques et offrant des opportunités d'apprentissage. L'activité principale générée

viser la réalisation de différents types de relevés. Les aspects "ludique" et "apprentissage" sont secondaires. Si l'on considère ces 3 aspects dans le même temps notre approche ne s'applique plus : il ne sera pas toujours possible de générer à la perspective *structure* une activité qui conviendra au triplet d'objectifs (inventaire / ludique / apprentissage) généré lors de la première itération du générateur. La manière dont ces différents aspects doivent s'articuler reste un défi d'analyse et de conception important à relever pour permettre la réalisation de tels générateurs.

Dans le cadre du projet *Escape it!* nous réalisons actuellement des expérimentations du jeu d'apprentissage auprès d'enfants autistes, en supervision avec leur parent. Le protocole de l'expérimentation a été préparé avec l'aide d'une ergothérapeute spécialisée autisme, qui conduit également les sessions d'expérimentations. Ces dernières ont principalement pour but d'analyser l'utilité (questionnaire spécifique) et l'utilisabilité (questionnaire SUS) du jeu sérieux qui a été développé. L'observation directe des sessions de jeu permet également de collecter des besoins de correctifs sur le développement du jeu.

## Références

1. Anouar Tadlaoui, M., Souhaib, A., Khaldi, M., Carvalho, R. : Learner modeling in adaptive educational systems : A comparative study. *International Journal of Modern Education and Computer Science* **8**, 1–10 (03 2016)
2. Bielikova, M., Diveky, M., Jurnecka, P., Kajan, R., Omelina, L. : Automatic generation of adaptive, educational and multimedia computer games. vol. 2, pp. 371—384. *Signal Image Video Process.* (2008)
3. Laforcade, P., Laghouaouta, Y. : Supporting the adaptive generation of learning game scenarios with a model-driven engineering framework. In : Pammer-Schindler, V., Pérez-Sanagustín, M., Drachsler, H., Elferink, R., Scheffel, M. (eds.) *Lifelong Technology-Enhanced Learning (ECTEL'18)*. pp. 151–165. Springer International Publishing, Cham (2018)
4. Laforcade, P., Loiseau, E., Kacem, R. : A model-driven engineering process to support the adaptive generation of learning game scenarios. In : *Proceedings of the 10th International Conference on Computer Supported Education - Volume 2 : CSEDU.*, pp. 67–77. INSTICC, SciTePress (2018)
5. Laghouaouta, Y., Laforcade, P., Loiseau, E. : A pattern-matching based approach for problem solving in model transformations. In : *Proceedings of the 13th International Conference on Software Technologies, ICSoft 2018, Porto, Portugal, July 26-28, 2018*. pp. 113–123 (2018)
6. Sebaha, K., Mahmood Hussaan, A. : Architecture et modèles génériques pour la génération adaptative des scénarios de jeux sérieux. application : Jeu d'évaluation et de rééducation cognitives. *Sciences et Technologies de l'Information et de la Communication pour l'Éducation et la Formation* **21**(1), 615–648 (2014)
7. Vandewaetere, M., Desmet, P., Clarebout, G. : The contribution of learner characteristics in the development of computer-based adaptive learning environments. *Computers in Human Behavior* **27**(1), 118 – 130 (2011)