

Vers la spécification de règles d'acceptation pour la génération de scénarios adaptés

Pierre Laforcade

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

1 Contexte : génération d'éléments adaptés

Notre intérêt concerne l'adaptation dans un EIAH lorsque celle-ci est réalisée par la machine. Plus particulièrement, nous nous intéressons à l'adaptation consistant à *générer des éléments adaptés* dans un EIAH ¹.

1.1 Caractérisation et travaux existants

La tâche de génération est souvent caractérisée [9] par son *objectif* (personnalisation, individualisation, recommandation, etc.), la *cible* (l'élément généré pouvant relever des catégories suivantes : contenu, présentation, navigation, structure, processus), la *source*, ou contexte, à laquelle sera adaptée la cible (incluant généralement tout ou partie du profil de l'apprenant), les nombreuses *connaissances* nécessaires à la réalisation de l'adaptation (modèles du domaine didactique, modèle du jeu, modèle pédagogique, etc. selon le type d'EIAH impliqué) et les *règles* décrivant la réalisation de la génération ².

L'adaptation dans un EIAH est une thématique ancienne [1]. Dans le contexte des tuteurs intelligents ou des hypermédias, l'apprenant était seul avec le système. L'adaptation visait, entre autres, la détection et le diagnostic des erreurs de l'apprenant (STI) ou le support à la navigation adaptative (hypermédias). Les enjeux étaient principalement de représenter les différentes connaissances, de raisonner avec elles, et même de produire des nouvelles connaissances. De nombreuses techniques d'IA étaient alors utilisées : nombreux formalismes de représentation et de raisonnement, moteurs d'inférence, *machine learning*, algorithmes génétiques, etc.

Plus récemment, les types d'EIAH plus variés (LMS puis plateforme MOOC, jeux sérieux, Environnement Virtuels, etc.) ont mis en exergue de nouveaux constats (difficultés à expliciter l'expertise de l'adaptation, à la valider, etc.) et de nouveaux besoins à prendre en compte (dimensions ludique, émotionnelle, motivationnelle, etc.). Les enjeux portent désormais sur l'explicitation des informations nécessaires à la modélisation de l'adaptation, la validation de l'adaptation,

1. Par la suite, le terme *adaptation* sera utilisé comme raccourci pour exprimer la génération d'éléments adaptés.

2. Les travaux du GT-adaptation ont identifié de nombreux autres critères permettant de décrire toute forme d'adaptation.

l'explicabilité de l'adaptation réalisée, la conciliation de différentes dimensions orthogonales dans l'adaptation, etc. Pourtant une analyse de la littérature [2] permet de constater que de des propositions sur ces thèmes sont encore rares et qu'elles portent encore davantage encore sur les techniques pour mettre en oeuvre concrètement des contextes particuliers d'adaptation (point de vue très informatique) et sur la modélisation de nombreux éléments de connaissances ou de contexte (point de vue didactique, pédagogique, psychologique, etc.).

Pour autant, peu de travaux ont abordé la modélisation des *cibles* ou bien des *règles* de l'adaptation. En effet, la spécification globale de l'adaptation (sa description), sans s'intéresser à sa mise en oeuvre informatique (le comment), implique la modélisation séparée de l'ensemble des éléments impliqués (cible, sources, connaissances) et la modélisation des règles d'adaptation (mise en relation des éléments impliqués afin de décrire le fonctionnement de l'adaptation).

1.2 Le besoin de spécification des règles de génération

La spécification de l'adaptation joue un rôle central intermédiaire très important car elle se situe entre l'explicitation de l'adaptation par les experts (exprimée en langage naturel) et la mise en oeuvre informatique de l'adaptation (exprimée dans un langage uniquement interprétable par la machine). Si ce travail global de formalisation est rarement abordé et qu'il n'existe pas de formalisme dédié c'est parce qu'il dépend du contexte considéré (EIAH), du paradigme approprié à exprimer l'adaptation considéré (CBR, pattern-matching, règles logiques / floues / conditionnelles, etc.), de la complexité des éléments impliqués dans l'adaptation, de la complexité des règles d'adaptation, mais également des usages qui seront faits de cette spécification. En effet, le choix d'un cadre de formalisation pour spécifier l'adaptation peut être utile, entre autres, pour permettre :

1. de diriger l'explicitation : en orientant les échanges avec les experts métier afin de les questionner précisément sur les informations requises ;
2. de diriger la modélisation des éléments impliqués : en assurant de ne modéliser que les informations nécessaires, c.-à-d. celles qui seront impliquées dans la réalisation de l'adaptation ;
3. de diriger la mise en oeuvre concrète de l'adaptation (les *spécifications fonctionnelles et techniques* telles que définies en Génie Logiciel) ;
4. de valider les éléments adaptés/générés : analyser le résultat de l'adaptation pour indiquer s'il est valide ;
5. de valider la bonne mise en oeuvre de l'activité d'adaptation : analyser la mise en oeuvre de l'adaptation pour indiquer si la réalisation est correcte (les *tests système* du GL) ;
6. de simuler l'adaptation et ainsi de valider les règles et/ou stratégies d'adaptation : la mise en oeuvre est ici davantage basée sur des heuristiques pour réduire les coûts de ré-ingénierie de la mise en oeuvre ;
7. expliquer l'adaptation réalisée (les règles / choix / etc. qui ont été appliqués).

2 Nos Travaux passés et futurs

Nous avons abordé la génération d'éléments adaptés dans le contexte du projet *Escape it!* (jeu sérieux d'apprentissage). L'objectif général était de permettre de faciliter la généralisation de l'apprentissage des compétences en évitant de produire des scènes de jeu fixes (coût de développement). La génération doit produire des scénarios de jeu adaptés à l'apprenant (dimension apprentissage seulement pour le profil) et en prenant en compte les connaissances sur le jeu (quelles scènes et objets sont disponibles). Nous considérons un scénario comme la spécification des activités d'apprentissage que devra réaliser l'apprenant pendant la session de jeu concernée par la génération ; chaque activité est décrite en terme de configurations des éléments qui seront nécessaires au *player*.

2.1 L'approche GPS

L'approche GPS (appelée approche 3x3x2 dans [5]) consiste à guider la conception du générateur en la considérant selon 3 perspectives incrémentales pour les scénarios à générer, 3 dimensions de spécification des éléments du domaine (**G**ame-**P**rofile-**S**cenario), 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 [8]). Chaque perspective consiste alors à définir sous forme de méta-modèles les 3 catégories d'éléments (éléments à générer du scénario, éléments contextuels du profil et éléments décrivant le jeu d'apprentissage).

La méta-modélisation capture la syntaxe et une partie de la sémantique *structurelle* entre la cible et les différentes sources ainsi qu'entre les sources elles-mêmes (i.e. ce qui est commun à toutes les cibles, quel que soit la génération). La génération est dirigée par la description de la cible. Cette approche participe au point (2) des objectifs possibles pour la spécification de l'adaptation (1.2). Notre approche est accompagnée également d'un processus détaillant, pour l'informaticien aidé des experts métier, les différentes étapes d'application de l'approche [3]. Ceci participe également partiellement à l'objectif (1) listé en (1.2).

La sémantique structurelle capturée par l'approche ne permet pas de décrire complètement l'adaptation. Notre approche est limitée à l'expressivité des méta-modèles : il n'est pas possible de spécifier les règles de génération des scénarios en fonction des contenus concrets du profil et des éléments du jeu. Dans le projet *Escape it!* la génération du modèle cible (le scénario) a été traitée comme un problème de transformation de modèle classique en IDM³ avec une approche algorithmique procédurale (code Java avec du code EMF généré à partir des méta-modèles) couplée à l'utilisation d'un solveur pour la résolution de la satisfaction des contraintes souhaitées par l'expert. Bien qu'opérationnelle cette mise en oeuvre de la génération posait le problème du coût de la réingénierie du code

3. Ingénierie Dirigée par les Modèles

lorsque les règles de génération devaient être modifiées, supprimées ou ajoutées [4]. En effet, les experts métier du jeu sérieux visé ne sont pas pour autant des experts en adaptation sachant précisément quelle adaptation doit être réalisée et comment.

2.2 L'architecture de spécification de la génération à réaliser

Au lieu d'aborder la piste de la validation *au plus tôt* des règles de génération, nous avons envisagé de traiter le verrou lié au coût/temps de mise en oeuvre des règles dans le code du générateur en proposant une architecture de spécification de la génération à réaliser basée sur un *modèle de configuration* [6][7]. La génération est réalisée en 2 étapes : reconnaissance d'un pattern-matching et production du scénario. Le modèle de configuration, à la charge de l'informaticien, décrit les correspondances entre la cible et la source en termes de *partial target pattern* (précisant chacun sa structure et ses contraintes). Une infrastructure de pattern-matching et de production est générée en exploitant les informations de ce modèle. Toutefois, certaines règles de production, celles relatives aux *contraintes*, nécessitent l'intervention de l'informaticien (ajouts exprimés en OCL). Cette proposition d'architecture participe à la spécification des règles de génération dans un formalisme interprétable pour diriger la mise en oeuvre (point (3) de 1.2).

Le principal avantage de l'architecture est qu'elle exploite les modèles et méta-modèles spécifiés en appliquant l'approche GPS. Aussi, l'informaticien n'a pas à prendre en charge le développement du générateur, ni à développer le pattern-matching ou la résolution de contraintes par backtracking. Il doit uniquement décrire les patrons et les contraintes. Modifier les règles de génération revient alors à modifier ce modèle. En revanche, l'expression de ce modèle nécessite des compétences fortes en IDM : l'*effort* de traduction des règles de génération en langage naturel vers ce formalisme n'est pas négligeable.

2.3 Constats sur les règles de génération

L'expérience du projet *Escape It!* a mis en évidence plusieurs constats. Les règles de génération sont difficiles à identifier et expliciter pour les experts, et lorsqu'elles le sont elles peuvent parfois être incomplètes, ambiguës, incohérentes, incertaines. Les aspects structuraux capturés dans le méta-modèle du scénario (la cible) n'ont pas été remis en cause, en revanche, les aspects dynamiques ont nécessité des ajustements (ajout, suppression, modification des règles) après avoir été mis en oeuvre dans le code du générateur. La validation de ces éléments n'a pas requis que le générateur soit intégré dans le jeu sérieux : il était directement utilisable, car développé comme un composant activable indépendant ; l'analyse des scénarios générés était possible sans avoir besoin d'une représentation interne au jeu (ceci ne sera pas forcément généralisable dans d'autres contextes). Le temps nécessaire à la ré-ingénierie du code ne permettait toutefois pas de valider les règles dans le même temps que leur explicitation. Aussi, la mise en oeuvre des règles de génération pouvait introduire des biais de traduction, des

choix inconscients de règles d'adaptation lors de la réalisation, etc. Il a fallu également *vérifier* que le générateur fonctionnait correctement avant de chercher à *valider* les règles d'adaptation qu'il mettait en oeuvre.

3 Positionnement

3.1 Un besoin de validation des règles d'adaptation

Dans des contextes où l'explicitation des règles de génération est difficile et complexe il est nécessaire de chercher à valider ces informations au plus tôt afin de réduire les temps et coûts de ré-ingénierie. Cette validation nécessite la formalisation et l'interprétation de ces règles (objectif de spécification du point (4) de 1.2).

Plusieurs perspectives sont envisageables : centrer la spécification sur la source pour décrire comment obtenir la cible ou bien sur la cible pour décrire comment la générer en fonction de la source. De manière similaire à l'approche GPS nous conservons l'approche de spécification de l'ensemble des éléments impliqués, dirigée par la cible. Toutefois il est encore possible de distinguer la spécification de ce que doit être la cible générée (*quoi* générer) et la spécification de comment générer cette cible (*comment* générer). Le premier cas revient à décrire les propriétés et caractéristiques attendues de la cible. Ces propriétés peuvent faire référence aux éléments sources et aux autres éléments de connaissances sur l'EIAH. Ces **règles d'acceptation** permettent de pouvoir accepter ou rejeter (et expliquer pourquoi) si les éléments générés sont valides. Le second cas, complémentaire au premier, décrit également comment la génération doit être réalisée pour les situations où la déclaration ne suffit pas : il s'agit souvent des moyens de prendre en compte les *préférences* et les *contraintes* exprimées dans les règles d'acceptation. L'expert métier peut expliciter les règles d'acceptation, avec de l'aide (point (1) 1.2), mais n'a pas les compétences informatiques nécessaires à exprimer comment résoudre les contraintes et autres choix de mise en oeuvre, souvent liés à un paradigme spécifique de représentation des connaissances et de raisonnement.

3.2 Proposition générale et hypothèses

Nous avons décidé d'orienter nos travaux, dans un premier temps, vers la formalisation des règles d'acceptation centrée sur la description de la cible (i.e. les scénarios dans un jeu sérieux). Dans le contexte de nos travaux existants nous proposons de compléter la démarche GPS avec une formalisation des règles d'acceptation du scénario à générer. L'objectif principal correspond donc au point (4) de 1.2 mais les points (1) et (2) seront aussi concernés indirectement.

Nous émettons différentes hypothèses que nos futures propositions et leur analyse devront chercher à confirmer ou infirmer :

- les règles d'acceptation aide à comprendre l'adaptation en capturant ses *intentions* tandis que les règles de génération sont uniquement des choix informatiques de mise en oeuvre basés sur les règles d'acceptation ;

- la validation des règles de génération se réduit à valider les règles d'acceptation : en effet, le *comment* de la mise en oeuvre n'impacte pas l'intention de l'adaptation capturée dans les règles d'acceptation ;
- la validation des règles d'acceptation peut être partiellement réalisée sans la mise en oeuvre du générateur en vérifiant des modèles effectués "à la main", i.e. non générés, en collaboration entre les experts et les informaticiens.

3.3 Questions de recherche

Dans l'objectif général de valider au plus tôt les règles de génération, la validation des règles d'acceptation pose alors les questions de recherche suivantes :

- Q1** Comment déterminer ce sur quoi devra porter les informations qui seront capturées par les règles d'acceptation ?
- Q2** Quel formalisme pour garantir la complétude⁴, la cohérence⁵, la clarté⁶ et la concision⁷ des règles d'acceptation ?
- Q3** Comment exploiter les règles d'acceptation formalisées pour aider les experts à lever les incertitudes et à valider les règles au plus tôt ?

3.4 Objectifs

Afin de répondre à **Q1** nous allons tout d'abord définir plus précisément ce que sont les règles d'acceptation dans un contexte large de générations d'éléments adaptés. Ensuite nous allons identifier et caractériser ce que sont ces règles d'acceptation dans le cadre de l'IDM (besoins d'expressivité) afin d'identifier les informations manquantes dans l'approche GPS. Nous envisageons également de fournir des moyens génériques (démarches, outils) pour aider l'informaticien à expliciter les besoins d'informations dans le contexte d'une adaptation spécifique, en collaboration avec les experts.

En ce qui concerne **Q2** nous allons tout d'abord analyser les formalismes existants dans les différents éco-systèmes du cadre outillé de l'IDM. Si l'un de ces formalismes couvre l'ensemble des besoins d'expressivité alors il sera expérimenté. Sinon nous aurons à concevoir un formalisme dédié. Nous utiliserons ensuite ce formalisme sur différents cas d'étude afin de spécifier des règles d'acceptation. Ces applications seront analysées afin de vérifier si les propriétés attendues (concision, complétude, clarté et cohérences) sont atteintes.

Pour **Q3** nous utiliserons l'interpréteur associé au formalisme IDM, si nous avons pu en identifier un qui convenait, sinon nous aurons à développer un tel interpréteur. Nous tenterons ensuite d'identifier et de proposer en complément une démarche de validation des règles d'acceptation, en collaboration entre informaticiens et experts métiers, utilisant cet interpréteur. Il s'agira de prouver que le suivi de cette démarche permet de valider, à faible coût, les règles d'acceptation en encourageant le test rapide de différentes stratégies d'adaptation.

4. Pas de connaissances manquantes.

5. Pas de connaissances en contradiction avec d'autres.

6. Au sens formel, pas d'ambiguïté sur l'interprétation.

7. Pas de connaissances inutiles à la génération.

3.5 Méthode de recherche

Afin de pouvoir conduire les différentes activités de recherche que nous avons mentionnées nous avons choisi de nous orienter vers l'utilisation de cas d'étude tel que définis par [10] : “*an empirical inquiry that investigates a contemporary phenomenon within its real-life context, especially when the boundaries between the phenomenon and context are not clearly evident*”. Les cas d'étude sont en effet pertinents dans des contextes de recherche empirique en Informatique. Ils sont basés sur les expérimentations, les observations réelles avec le public-cible, etc.

Dans un premier temps nous comptons réaliser une **étude de faisabilité** sur le cas d'étude d'*Escape it!* pour lequel nous avons déjà collecté de nombreuses informations et avons déjà appliqué la démarche GPS. Nous réaliserons ensuite une **étude pilote** pour appliquer et affiner les propositions sur un autre cas d'étude.

L'analyse des propositions et de leurs applications après les deux études devraient permettre d'apporter des éléments de réponses étayés aux questions de recherche identifiées mais également de confirmer ou infirmer les hypothèses sur lesquelles repose la problématique de ces travaux.

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. Hocine, N., Gouaich, A., Di Loreto, I., Abrouk, L. : Techniques d'adaptation dans les jeux ludiques et sérieux. *Revue des Sciences et Technologies de l'Information - Série RIA : Revue d'Intelligence Artificielle, Lavoisier* **25**(2), 253–280 (2011)
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., Laghouaouta, Y. : Génération de scénarios adaptés dans un jeu d'apprentissage selon une approche dirigée par les modèles. In : *Environnements Informatiques pour l'Apprentissage Humain (EIAH19)*, Jun 2019, Paris, France. pp. 169–180 (2019)
5. 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)
6. Laghouaouta, Y., Laforcade, P. : A practical approach for constraint solving in model transformations. In : *Software Technologies, 1077*, Springer, pp.104-123, 2019, *Communications in Computer and Information Science*. pp. 104–123 (2019)
7. Laghouaouta, Y., Laforcade, P. : Dealing with uncertainty in model transformations. In : *The 35th ACM/SIGAPP Symposium on Applied Computing (SAC'20)*, Mar 2020, Brno, Czech Republic. pp. 1595–1603104 (2020)

8. 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)
9. 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)
10. Yin, R.K. : Case study research : design and methods. Thousand Oaks, Calif : Sage Publications (2003)