

Modélisation des parcours pédago-ludiques pour l'adaptation des jeux sérieux

Bertrand Marne, Thibault Carron, Jean-Marc Labat

► **To cite this version:**

Bertrand Marne, Thibault Carron, Jean-Marc Labat. Modélisation des parcours pédago-ludiques pour l'adaptation des jeux sérieux. Christophe Choquet; Philippe Dessus; Marie Lefevre; Julien Broisin; Olivier Catteau; Philippe Vidal. 6e Conférence sur les Environnements Informatiques pour l'Apprentissage Humain, May 2013, Toulouse, France. IRIT Press 2013, pp.55-66, 2013. <hal-00839749>

HAL Id: hal-00839749

<https://hal.sorbonne-universite.fr/hal-00839749>

Submitted on 29 Jun 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Modélisation des parcours pédago-ludiques pour l'adaptation des jeux sérieux

Conférence EIAH 2013 (Environnements Informatiques pour l'Apprentissage Humain)

Bertrand Marne*, Thibault Carron*, Jean-Marc Labat*

* LIP6

Université Pierre et Marie Curie

4 place Jussieu

75005 Paris

bertrand.marne@lip6.fr

thibault.carron@lip6.fr

jean-marc.labat@lip6.fr

RÉSUMÉ. Pour pouvoir exprimer tout leur potentiel, les jeux sérieux doivent être acceptés et assimilés par les enseignants. Dans cette optique, nous présentons ici nos contributions à la résolution de la problématique de l'adaptation des jeux sérieux. Nous proposons un modèle de scénarisation nommé MoPPLiq qui est capable de représenter à la fois les aspects pédagogiques et les aspects ludiques de jeux sérieux qui peuvent être découpés en étapes. Nous proposons également un système de contrôle et de compensation des incohérences lors des manipulations des parcours pédago-ludiques modélisés dans notre outil auteur (APPLiq). Notre modèle a fait l'objet de mises à l'épreuve dans deux contextes qui montrent que son expressivité convient à la modélisation des scénarios discontinus de jeux sérieux.

MOTS-CLÉS: jeux sérieux, scénarisation, modélisation, adaptation, outil auteur.

1. Introduction

De manière croissante, les jeux sérieux sont utilisés dans un contexte d'enseignement traditionnel [SQUIRE 03]. En effet, leurs apports en matière de motivation et d'attractivité pour la génération actuelle d'apprenants et les plus-values potentielles d'ordre pédagogique qui ont été identifiées expliquent cet engouement [BOGOST 07]. Afin de pouvoir exprimer son potentiel, le jeu sérieux ne doit pas rester un objet pédagogique statique, mais doit être assimilé par l'enseignant qui va ainsi le modeler, l'adapter en fonction de ses besoins, du contexte [EGENFELDT-NIELSEN 04]. Nous avons constaté qu'un certain nombre d'enseignants en maîtrisent l'utilisation, mais expriment le besoin d'avoir la même flexibilité qu'avec leurs supports pédagogiques habituels. Ils souhaitent notamment intervenir sur la scénarisation des jeux sérieux [MARNE & LABAT 12]. En effet, ces derniers sont fréquemment composés d'étapes (niveaux, études de cas, exercices, activités, sous-activités, etc.) sur lesquelles les enseignants veulent être en mesure d'agir, essentiellement sur les paramètres pédagogiques, les paramètres plus fonctionnels, ou ludiques ne relevant généralement pas de leur expertise. Cependant, lors d'interventions sur les paramètres pédagogiques, il faudra veiller à maintenir automatiquement les contraintes ludiques, voire fonctionnelles. Car ces trois types de paramètres sont étroitement imbriqués dans un jeu sérieux. La difficulté qui en résulte est de pouvoir représenter tous ces éléments imbriqués sur dans même modèle. Actuellement, il n'existe pas de modèle de référence, et lorsqu'un effort de modélisation a été fait, il y a presque autant de représentations possibles que de jeux sérieux.

L'objectif de ce travail est donc de proposer un modèle capable de représenter de façon intelligible la scénarisation pédo-ludique des jeux sérieux et un outil logiciel pour qu'elle puisse être manipulée par les enseignants.

Dans la première partie de cet article, nous présentons notre contribution à cette problématique sous la forme d'un nouveau modèle nommé *MoPPLiq* et de son implémentation (l'outil auteur nommé *APPLiq*). Dans une seconde partie, nous présentons les systèmes que nous avons mis en place dans *APPLiq* pour mesurer et compenser les incohérences pédagogiques ou ludiques qui apparaissent lorsque qu'un enseignant adapte la scénarisation initiale à ses besoins. Puis, avant de conclure, nous proposons dans la troisième partie une mise à l'épreuve de notre modèle de scénarisation par le biais de la modélisation de plusieurs types jeux sérieux, et par l'interfaçage de notre outil auteur avec d'autres d'outils auteurs de jeux sérieux.

2. Modélisation des Parcours Pédago-Ludiques (*MoPPLiq*)

Pour répondre au besoin d'adaptation observé chez les enseignants [MARNE & LABAT 12], nous avons conçu un modèle : *MoPPLiq* : **Modélisation des Parcours Pédago-Ludiques**. Ce modèle et son implémentation ont été conçus pour les jeux sérieux dont le

scénario peut être fractionné en étapes. Il s'agit par exemple des jeux sérieux ayant des niveaux, ou qui permettent de traiter des études de cas différentes, ou encore qui proposent plusieurs types d'exercices.

Pour construire un modèle capable de représenter des activités avec leurs enchaînements et qui puisse être manipulé par les enseignants, nous nous sommes appuyés sur l'état de l'art actuel des modèles de scénarisation des jeux sérieux, mais aussi des EIAH et des jeux vidéo et avons isolé trois aspects fondamentaux exemplifiés par des patrons de conception : en premier lieu, le patron de conception « *Levels* » [BJÖRK & HOLOPAINEN 05] met en évidence que les scénarios sont en général découpés par buts ou intentions, notamment pédagogiques, mais aussi ludiques dans le cas des jeux vidéos et sérieux. En second lieu, nous avons mis en évidence dans le patron de conception « *Objectifs hiérarchisés et pédagogiques* »^[web 1] que la plupart des EIAH, des jeux sérieux et des jeux vidéo qui proposent des scénarios non linéaires fondent cet arbre de branchements sur les buts atteints par l'apprenant-joueur et sur la hiérarchie de ces buts. Enfin, nous avons retenu un troisième aspect pour notre modélisation, qui est la capacité de certains EIAH, jeux sérieux et jeux à proposer une adaptation dynamique fondée sur l'historique des actions de l'apprenant-joueur. Par exemple, en donnant de l'aide à ceux qui ont du mal à résoudre les problèmes ou au contraire en augmentant la difficulté pour les apprenants-joueurs les plus performants. Plus concrètement, les principales particularités du modèle *MoPPLiq* sont présentées dans les sections suivantes.

2.1. Un découpage des scénarios en « boîtes noires » caractérisées par leurs buts.

Dans l'offre existante de jeux sérieux dont le scénario peut être découpé, certains comme *Refraction*¹, proposent des étapes qui sont construites sur le même modèle (ex. : niveaux, études de cas). Par conséquent, la modélisation des mécanismes de l'une de ces étapes suffit à modéliser l'ensemble d'entre elles. Leurs particularités se situent dans les différents paramètres à leur appliquer. En revanche, d'autres types de jeux sérieux comme *Science en jeu*² ont un scénario composé d'étapes aux mécanismes très variés (ex. : quêtes, activités). Dans ces cas-là, nous ne pouvons pas isoler de mécanisme générique et par conséquent, pour notre modèle de scénarisation, il n'était pas envisageable de modéliser les mécanismes de chaque étape.

Ne pouvant pas décrire de façon générique le contenu des étapes du scénario d'un même jeu sérieux, nous avons donc choisi de considérer ces étapes comme des « boîtes noires » que nous appelons *activités*. En effet, dans le contexte de l'adaptation du scénario d'un jeu sérieux, nous n'avons pas besoin de connaître les mécanismes internes des activités. Ce qui nous importe est de connaître les conséquences de ces mécanismes sur l'évolution du

¹ *Refraction* est un jeu sérieux de type puzzle, destiné à l'apprentissage de la manipulation des fractions^[web 2].

² *Science en jeu* est un jeu sérieux de rôle, massivement multi-joueurs, visant différents aspects de l'apprentissage des sciences dans de nombreuses quêtes découpées en activités^[web 3].

4 Environnements Informatiques pour l'Apprentissage Humain, Toulouse 2013

```
<activity id_activity="11">
  <name>Niveau 6.2</name>
  <description>Tutoriel pour comprendre que l'on peut additionner des lasers
  de même dénominateurs.</description>
  [...]
  <input_state id_input="9" />
  [...]
  <output_state id_output="13"/>
  [...]
</activity>
<goals>
<goal id_goal="24" type="edu">
  <name>Comprendre qu'une fraction est une proportion </name>
  <goal_links>
  <goal_link object="output_state" id_object="13" />
  </goal_links >
</goal>
<goal id_goal="38" type="peda">
  <name>Adding Additionner des fractions de dénominateurs différents
  </name>
  <goal_links>
  <goal_link object="output_state" id_object="13" />
  </goal_links >
</goal>
  [...]
</goals>
```

Figure 1. Extrait du fichier XML décrivant le niveau 6.2 du jeu sérieux *Refraction*.

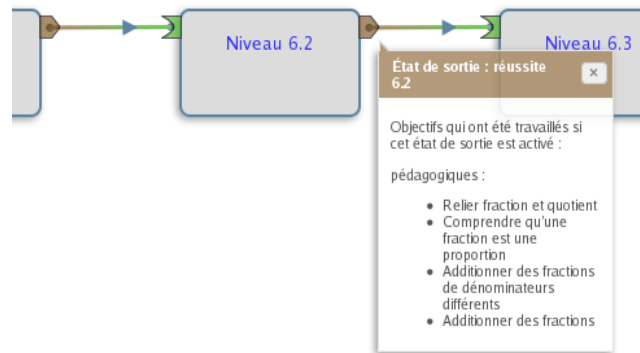


Figure 2. Représentation graphique du formalisme de MoPPLiq : l'activité (boîte aux coins arrondis) est caractérisée par des objectifs pédagogiques (décrits dans la bulle d'information).

scénario. Autrement dit, nous avons besoin de savoir ce que le déroulement de l'activité change dans le modèle de l'apprenant-joueur et dans les paramètres du jeu. C'est pour ces raisons que pour décrire les activités dans *MoPPLiq*, nous avons choisi de les caractériser par les objectifs (buts) pédagogiques et ludiques qu'elles permettent d'atteindre. Notamment quand ceux-ci ont un impact sur le modèle de l'apprenant-joueur et sur les paramètres du jeu sérieux. Ces objectifs correspondent alors aux paramètres de sortie de nos activités.

Par exemple, le niveau 6.2 de *Refraction* permet d'atteindre plusieurs objectifs pédagogiques comme « Comprendre qu'une fraction est une proportion » ou « Additionner des fractions de dénominateurs différents », etc. Les figures 1 et 2 ci-après montrent comment nous modélisons ce niveau 6.2 avec le formalisme de *MoPPLiq*.

Nous utilisons l'élément `<activity>` pour décrire l'activité elle-même. Avec les éléments `<goal>` (but) associés à la sortie (`<output_state>`) de l'activité, nous caractérisons ses objectifs pédagogiques (l'attribut `type` différencie les buts pédagogiques et ludiques : `peda` ou `ludo`). Les éléments `<goal>` sont aussi associés à des éléments `<input_state>` (correspondant au début de l'activité). Ils permettent de décrire les prérequis qui sont nécessaires pour commencer une activité. Par exemple, `<goal_link object="input_state" id=object="42" />` signifie que l'objectif est associé à l'entrée n° 42.

Tous les jeux sérieux n'ont pas comme *Refraction* un parcours linéaire. Pour ceux qui proposent des branchements, nous avons introduit dans *MoPPLiq* le principe des *états de sortie* multiples que nous décrivons dans la section suivante.

2.2. Construire une arborescence d'activités à partir des choix de l'apprenant-joueur

Dans certains jeux sérieux, comme *Les ECSPER*³, le branchement des activités aux suivantes dépend d'événements qui se sont produits au cours de leur déroulement. Ces événements sont des conséquences des choix des apprenants-joueurs. Par exemple, dans l'une des activités du jeu *Les ECSPER*, l'apprenant-joueur doit déterminer si le mode de rupture d'une vis est fragile ou ductile. En fonction de son choix, l'activité qui suivra sera différente. Si la réponse est fautive, il aura une activité de remédiation, sinon il avancera à l'activité suivante. Dans le cas d'une bonne réponse, et en l'absence d'un système de suivi plus élaboré⁴, on peut considérer que le but pédagogique « reconnaître un mode de rupture ductile » a été atteint.

Dans *MoPPLiq*, pour prendre en compte des scénarios non-linéaires, nous matérialisons les choix de l'apprenant-joueur et leurs conséquences dans l'activité par des états de sorties différents : d'une part, chaque état de sortie peut être branché à une activité différente, d'autre part chaque état de sortie peut être caractérisé par le travail par l'apprenant-joueur sur des objectifs (<goal>) pédagogiques ou ludiques différents. La figure 3 montre comment nous formalisons cela en XML : l'activité contient plusieurs <output_state> qui sont à leur tour reliés (par l'élément <output_input_link>) à d'autres activités. La bonne réponse (« ductile ») mène à l'activité suivante et est liée à l'objectif pédagogique « reconnaître un mode de rupture ductile » (<goal_link object="output_state" id_object="35" />). Cet objectif est également requis pour l'activité « Examen de la surface de rupture ». La figure 4 est la représentation graphique de ce modèle.

³ *Les ECSPER* (Études de Cas Scientifiques et Pratiques pour l'Expertise en Rupture) est un jeu sérieux destiné à faire étudier à des élèves ingénieurs la mécanique de la rupture^[web 4].

⁴ Sachant qu'en l'absence d'un système de suivi il est rare de pouvoir déterminer automatiquement qu'un objectif pédagogique travaillé a pu être atteint, nous avons choisi de considérer dans *MoPPLiq* qu'à la fin d'une activité les objectifs étaient seulement « travaillés ».

2.3. Prendre en compte les possibilités d'adaptations dynamiques au modèle de l'apprenant-joueur

Dans certains jeux sérieux, nous avons identifié une autre caractéristique importante : les activités peuvent avoir un fonctionnement différent en fonction des choix et performances passés de l'apprenant-joueur et non plus seulement dans l'activité en cours. Ces choix et ces

```

<activity id_activity="24">
  <name>Analyse du mode de rupture </name>
[... ]
  <output_states>
    <output_state id_output="34" name="Fragile" />
    <output_state id_output="35" name="Ductile" />
  </output_states>
  <activity>
    <activity id_activity="25">
      <name>Examen Examen de la surface du rupture </name>
[... ]
      <input_state id_input="40" />
[... ]
    </activity>
    <activity id_activity="30">
      <name>Observation détaillée de la surface de rupture </name>
[... ]
      <input_state id_input="32" />
[... ]
    </activity>
  <links>
    <output_input_link id_output="34" id_input="28" />
    <output_input_link id_output="35" id_input="32" />
  </links>
  <goals>
    <goal id_goal="23" type="peda">
      <name>Reconnaitre un mode de rupture ductile </name>
      <goal_links>
        <goal_link object="output_state" id_object="35" />
        <goal_link object="input_state" id_object="40" />
      </goal_links>
    </goal>
  </goals>

```

Figure 3. Extrait du fichier XML décrivant le scénario non linéaire du jeu sérieux Les ECSPER.

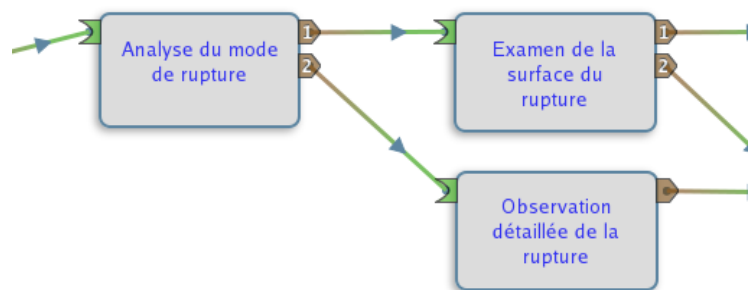


Figure 4. Représentation graphique d'un scénario non linéaire décrit avec MoPPLiq : les pointes brunes numérotées 1 et 2 sont les états de sortie qui symbolisent les choix des joueurs et mènent à des activités différentes.

performances sont collectés par le jeu dans le « modèle de l'apprenant-joueur » qui est ensuite interrogé lorsqu'une activité capable de s'adapter s'exécute. C'est le cas pour *Science en jeu* ou encore *Learning Adventure* [MARTY et al. 12]. Par exemple, l'une des activités de *Science en jeu* demande à l'apprenant-joueur d'analyser de l'eau pour détecter son niveau de pollution. Si le joueur n'a jamais fait d'analyse de l'eau auparavant, un tutoriel lui est fourni, sinon il devra faire l'analyse sans aide.

Pour modéliser cette prise en compte dynamique du modèle de l'apprenant-joueur, nous avons introduit dans *MoPPLiq* la notion d'état d'entrée (<input_state>) : chaque état d'entrée correspond à un mode de fonctionnement de l'activité, et est caractérisé par les prérequis nécessaires à l'exécution de ce mode (voir figures 5 et 6).

```

<activity id_activity="15">
<name> Analyser des échantillons d'eau (pollution)</name>
<input_states>
<input_sate id_state="17" name="Débutant" />
<input_sate id_state="18" name="Expert" />
</input_states>
<output_states>
<output_state id_output="25" name="Réussite du Tutoriel" />
<output_state id_output="26" name="Réussite autonome " />
</output_states>
</activity>
<goals>
<goal id_goal="41" type="peda">
<name> Avoir fait une première analyse de l'eau (pollution)</name>
<goal_links>
<goal_link object="output_state" id_object="25" />
<goal_link object="input_state" id_object="18" />
</goal_links >
</goal>
</goals>

```

Figure 5. Extrait du fichier XML décrivant grâce à plusieurs états d'entrée les différents comportements d'une activité du jeu sérieux Science en jeu.

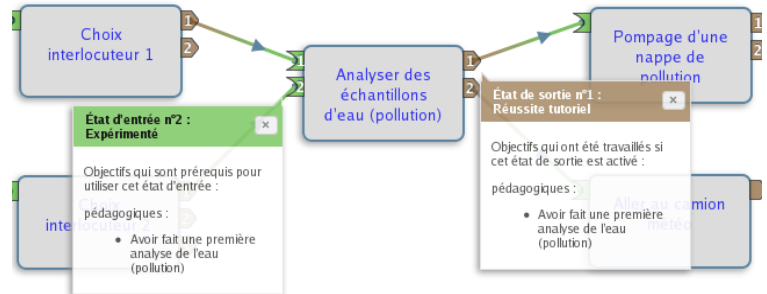


Figure 6. Formalisation graphique d'une activité proposant des fonctionnements adaptés au profil de l'apprenant-joueur. Les chevrons verts numérotés 1 et 2 sont les états d'entrée qui symbolisent ces différents comportements.

L'objectif (<goal>) « Avoir fait une première analyse de l'eau » est lié à l'état d'entrée (<input_state>) « Expert », car c'est un prérequis. L'apprenant-joueur novice commence par l'<input_state> « Débutant » qui n'a pas de prérequis et propose le tutoriel.

2.4. Un outil auteur pour adapter les parcours pédago-ludiques modélisés avec MoPPLiq

Les représentations graphiques des figures 2, 4, 6, 8 et 9 sont issues de notre prototype permettant de manipuler les scénarios modélisés avec *MoPPLiq*. Cet outil auteur se nomme *APPLiq* : **A**daptation des **P**arcours **P**édago-**L**udiques. Ce prototype est une application web est développé en PHP, MySQL, JavaScript. Son objectif principal est de permettre aux enseignants de visualiser et de manipuler la scénarisation des jeux sérieux. Il s'agit de leur permettre de construire des parcours pédagogiques répondant à leurs besoins en tenant compte des contraintes ludiques des jeux sérieux. L'objectif secondaire d'*APPLiq* est de permettre aux concepteurs de jeux sérieux de modéliser leurs scénarios avec *MoPPLiq*.

D'un point de vue méthodologique, nous avons initié le développement d'*APPLiq* avec une démarche centrée utilisateur et des itérations les plus courtes possible. Dans un premier temps, les utilisateurs avec lesquels nous travaillons ont tous le profil de concepteurs de jeux sérieux désirant modéliser leur logiciel avec *APPLiq*. Dans un second temps, nous testerons le profil d'enseignants qui cherchent à construire ou à modifier un parcours pédagogique avec les activités d'un jeu sérieux.

APPLiq permet donc aux enseignants de consulter et surtout de modifier des parcours pédago-ludiques pour les adapter à leur contexte pédagogique et donc à leurs besoins. Cependant, au cours de ces modifications, des incohérences (de nature pédagogique ou ludique) peuvent apparaître. Dans la section suivante, nous présentons les fonctionnalités que nous avons développées dans *APPLiq* pour détecter et résorber les incohérences issues de ces adaptations.

3. Maintenir la validité des modèles adaptés

Quand les enseignants construisent ou modifient un parcours pédagogique dans *APPLiq*, des inadéquations entre les prérequis et les buts qui ont pu être travaillés par les apprenants-joueurs dans les activités précédentes peuvent survenir. C'est pourquoi nous avons doté *APPLiq* d'un système de détection de ces incohérences : au moment de la connexion d'un état de sortie à un état d'entrée, il détecte les éventuelles incohérences en deux temps. Dans un premier temps, *APPLiq* détermine et stocke l'ensemble des objectifs susceptibles d'être travaillés par l'apprenant-joueur dans chacun des chemins possibles du parcours jusqu'à l'état de sortie désigné. Puis à partir de ces données, il calcule l'ensemble d'objectifs qui est le plus petit dénominateur commun des ensembles d'objectifs de chaque chemin qui peut être parcouru par l'apprenant-joueur jusqu'à l'état de sortie. Cet ensemble calculé (voir figure 7) contient seulement les objectifs dont on peut garantir qu'ils seront travaillés par l'apprenant-joueur, quel que soit le chemin emprunté dans le parcours. Le calcul de cet ensemble est nécessaire, car la cohérence est ici calculée a priori : avant que l'apprenant-joueur ne commence à jouer. Dans un second temps, *APPLiq* vérifie que les objectifs prérequis par l'état d'entrée sont bien tous présents dans l'ensemble d'objectifs précédemment calculé.

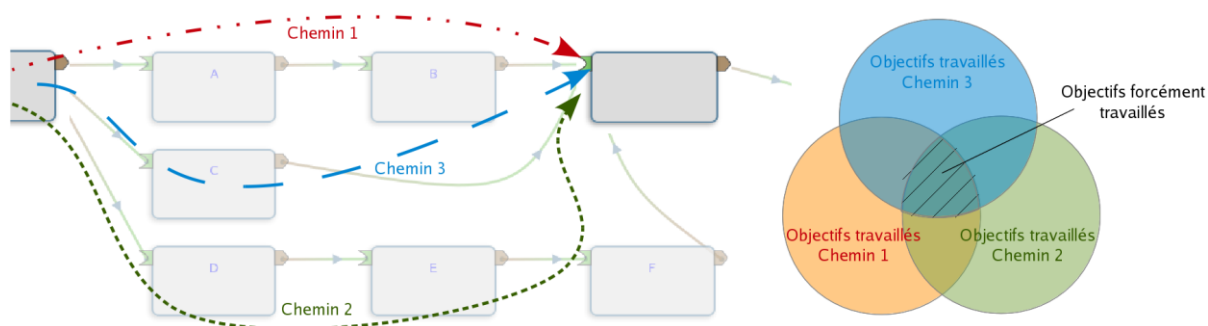


Figure 7. À gauche, un exemple de parcours non linéaire avec plusieurs chemins possibles (chemins 1 à 3). À droite, chaque cercle représente l'ensemble des objectifs travaillés pour un chemin donné du parcours. À l'intersection de ces trois ensembles (cercles) se trouve l'ensemble des objectifs qui seront forcément travaillés par l'apprenant-joueur, quel que soit le chemin qu'il choisisse dans le parcours.

Quand au cours d'une construction ou d'une modification d'un parcours, un enseignant crée une incohérence pédagogique entre les objectifs travaillés et ceux qui sont prérequis, *APPLiq* la lui signale. Cet utilisateur pourra donc agir en connaissance de cause. Nous avons choisi de ne pas créer de contrainte empêchant l'enseignant de construire le parcours pédagogique qu'il souhaite, y compris avec des incohérences pédagogiques, car il peut éventuellement avoir de bonnes raisons de construire un parcours pédagogiquement « incohérent ». En effet, ce parcours peut trouver sa cohérence dans son contexte d'utilisation : lors d'une formation mixte (jeu sérieux accompagné d'une formation plus classique), en fonction de la présence d'un tuteur lors des sessions de jeu, ou en fonction des acquis antérieurs des apprenants, etc.

A contrario, il est nécessaire d'empêcher les enseignants de construire des parcours qui ont des incohérences ludiques, car elles pourraient rendre le jeu sérieux impossible à jouer par les apprenants-joueurs. Les enseignants que nous avons interrogés nous ont expliqué qu'ils ne voulaient pas avoir à gérer d'autres aspects que les aspects pédagogiques lors de la re-scénarisation des jeux sérieux. Par conséquent, et contrairement aux aspects pédagogiques que nous laissons les enseignants gérer, nous avons dû mettre en place dans *APPLiq* un automatisme pour supprimer les incohérences ludiques. Cet automatisme se fonde sur des activités particulières que nous appelons *activités tampons*.

La figure 8 est un exemple du fonctionnement des *activités tampons* : lorsqu'un enseignant essaie de relier deux activités, *APPLiq* vérifie que les prérequis ont pu être atteints par l'apprenant-joueur avec les activités qui précèdent (voir aussi figure 7). Ici, ce n'est pas le cas puisque l'activité 2 utilise de l'électricité. Elle a donc comme prérequis que le courant est branché, mais cet objectif ne peut être atteint dans les activités qui précèdent. Dans ce cas-là, nous avons choisi de ne pas empêcher le branchement de ces deux activités, car leur enchaînement peut correspondre aux besoins pédagogiques de l'enseignant qui construit le parcours. Mais en revanche, *APPLiq* lui propose une solution pour compenser l'incohérence ludique du parcours. Cette compensation prend la forme d'une activité qui s'intercale automatiquement et qui permet d'atteindre l'objectif manquant (« brancher l'électricité »).



Figure 8. Les activités tampons (ici en vert, au contour pointillé) sont intercalées automatiquement par *APPLiq* pour compenser les incohérences ludiques.

Ces activités tampons ont deux caractéristiques principales : d'une part, elles n'ont pas de prérequis afin de pouvoir être placées à n'importe quel endroit du parcours. D'autre part, elles n'ont jamais de but pédagogique (seulement des buts ludiques) afin de ne pas perturber les progressions construites ou modifiées par les enseignants. Ainsi, lors de la conception et la réalisation du jeu sérieux, il faut avoir prévu le développement de ces activités tampons.

Le système des activités tampons adjoint au modèle *MoPPLiq* et à son implémentation *APPLiq* permettent aux enseignants d'adapter la séquence pédagogique du scénario des jeux sérieux à leurs besoins, sans pour autant craindre d'introduire à leur insu des incohérences pédagogiques ou de produire des jeux dysfonctionnels. Nous avons entrepris une démarche d'évaluation de ces outils, qui commence par la vérification de *MoPPLiq* et notamment de sa capacité à modéliser les jeux sérieux existants.

4. Mesurer la généricité du modèle

Valider le modèle *MoPPLiq* revient à vérifier qu'il est capable de représenter les différents aspects des scénarios des jeux sérieux qui sont nécessaires pour comprendre et pour modifier l'ordre de leurs activités. Pour faire cette vérification, nous avons suivi deux axes. Le premier axe est de travailler à la modélisation de nombreux jeux sérieux parmi ceux qui peuvent être décomposés en étapes. Le second axe est d'essayer de permettre à *APPLiq* d'importer des scénarios exprimés dans le formalisme d'autres outils auteurs de jeux sérieux.

4.1. Modélisation de jeux sérieux à étapes

Ce travail de modélisation systématique ne peut être exhaustif, mais nous a déjà permis de classer en trois catégories les jeux sérieux sur la modélisation desquels nous avons travaillé.

La première catégorie, celle des jeux sérieux à scénario linéaire (à l'image de *Refraction*), représente presque la totalité des jeux sérieux que nous avons pu modéliser, car ils sont les plus fréquents sur le marché. Nous pouvons citer : *CellCraft*, *Donjons & Radon*, *Prog & Play*, *StarBank*, *FoodForce*, *Ludiville*, etc. De fait, la linéarité de leurs scénarios facilite grandement leur modélisation, ainsi que la lisibilité des graphes produits par *APPLiq*. La caractérisation des activités par les objectifs atteints et prérequis associés à leur état d'entrée et leur état de sortie permet bien de décrire le contenu de façon à le réorganiser. Toutefois, certains jeux sérieux comme *CellCraft*^[web 5] sont plus complexes à modéliser, et nous parvenons mal à le faire par le biais de *MoPPLiq*. En effet, *CellCraft* est un jeu sérieux dont le découpage en niveaux est rendu cohérent par une histoire qui sert de fil rouge. Nous éprouvons des difficultés à exprimer les éléments clés de cette histoire sous forme de buts et de prérequis ludiques qui puissent par la suite être compensés par des activités tampons.

La seconde catégorie contient les jeux sérieux à scénarisation non linéaire, à l'image du jeu *Les ECSPER*. Ces jeux sérieux sont beaucoup plus rares, mais nous pouvons citer pour illustrer *Mécagénus* ou *Défenses Immunitaires*. L'utilisation de plusieurs états de sortie s'est montrée parfaitement adaptée à la modélisation de ces jeux sérieux qui proposent des alternatives dans leurs scénarios en fonction des choix de l'apprenant-joueur.

La troisième catégorie de jeux sérieux correspond à ceux qui sont capables de prendre en compte le modèle de l'apprenant-joueur pour modifier le comportement de leurs activités. Le seul jeu sérieux de cette catégorie sur la modélisation duquel nous avons travaillé est *Science en jeu*. Nous n'avons pas rencontré de difficulté de modélisation, sinon la complexité des graphes qui en résultent.

4.2. Import des modèles d'autres outils auteurs dans APPLiq

Afin d'éprouver l'expressivité de notre formalisme, nous avons aussi décidé d'ajouter une fonction d'import des fichiers produits par des outils auteurs de jeux sérieux dans *APPLiq* et donc faisant la transformation de leur modèle vers *MoPPLiq*. Pour confronter *MoPPLiq* à des formalismes très différents, nous avons choisi d'utiliser deux outils auteurs proposant des concepts et des fonctionnalités très dissemblables. Dans un premier temps, nous avons travaillé avec l'outil auteur *Legadee* [MARFISI-SCHOTTMAN 12] qui est destiné à la conception de jeux sérieux et dont le formalisme repose sur la dualité entre les aspects pédagogiques et ludiques des jeux sérieux. D'autre part, nous avons travaillé avec *eAdventure* [MORENO-GER et al. 05] qui est un outil auteur de développement de jeux sérieux de type « point & click » parmi les plus aboutis.

4.2.1. Import depuis Legadee

Dans *Legadee*, les scénarios conçus sont découpés en « screens » [MARFISI-SCHOTTMAN 12 ; MARFISI-SCHOTTMAN et al. 10] qui correspondent dans *MoPPLiq* aux activités. Ainsi, nous avons pu aisément récupérer la plupart des éléments que le modèle de *Legadee* exprimait (voir le tableau 1). Le modèle *MoPPLiq* s'est même montré plus formel sur certains aspects. Ainsi, dans *Legadee* les activités de départ et de fin ne sont pas explicitement déclarées, les contraintes et buts ludiques ne sont pas décrits ailleurs que dans les textes descriptifs des « screens », et de manière générale il n'y a pas vraiment de système de prérequis (même si les enchainements de « screens » peuvent être conditionnés). Cette différence de formalisme s'explique facilement par les buts de *Legadee* qui ne cherche pas à fournir un formalisme opérationnalisable, mais capable de décrire la scénarisation à des développeurs [MARFISI-SCHOTTMAN 12]. Toutefois, il y a un aspect du modèle de scénarisation de *Legadee* que nous n'avons pas été en mesure de transférer dans *MoPPLiq* : dans les scénarios conçus avec *Legadee* il y a des éléments gigognes ou imbriqués (les « screens » sont contenus dans des « sequences », elles-mêmes contenues dans des « missions »). C'est une caractéristique répandue des modèles de scénarisation (voir aussi *ScenLRPG* [MARIAIS 12], *LAMS* [DALZIEL 08] ou *Collage* [HERNÁNDEZ-LEO et al. 06]) et qui n'existe pas encore dans *MoPPLiq* et *APPLiq*.

4.2.2. Import depuis eAdventure

Le modèle sous-jacent de *eAdventure* est très bien décrit dans la littérature [MORENO-GER et al. 07], bien qu'il ait bien évolué depuis. À l'inverse de *Legadee*, le modèle de scénarisation de *eAdventure* diffère fondamentalement de celui de *MoPPLiq*. Car, si comme dans *Legadee* les scénarios sont bien découpés (cette fois-ci en « scenes »), les moyens de brancher les « scenes » entre-elles sont très variés et peuvent être causés par des objets contenus dans les scènes, des conversations tenues avec les personnages présents ou même des événements indépendants du joueur. Bien que très complexe et difficile à développer, le traitement automatique de la transformation du modèle de *eAdventure* vers celui de *MoPPLiq* a été réussi et est capable de prendre en charge la plus grande partie des éléments exprimés dans le modèle d'*eAdventure* (voir le tableau 1). Nous rencontrons néanmoins deux difficultés. La première est la taille et la complexité des graphes des scénarios issus de l'import dans *APPLiq*. Cette complexité nous ramène à la question des activités gigognes évoquée plus haut. La seconde difficulté est la présence dans les scénarios de *eAdventure* d'événements « trans-activités ». Par exemple, dans le jeu sérieux *Fire Protocol*^[web 6] réalisé avec *eAdventure*, un compte à rebours (timer) décompte le temps à partir du commencement du jeu et interrompt la partie lorsqu'il atteint zéro (incendie). Ce compte à rebours est donc indépendant des activités, et peut arriver à son terme alors que l'apprenant-joueur est occupé à l'une d'elles. Nous n'avons pas encore vraiment modélisé cela avec *MoPPLiq* (voir la figure 9).

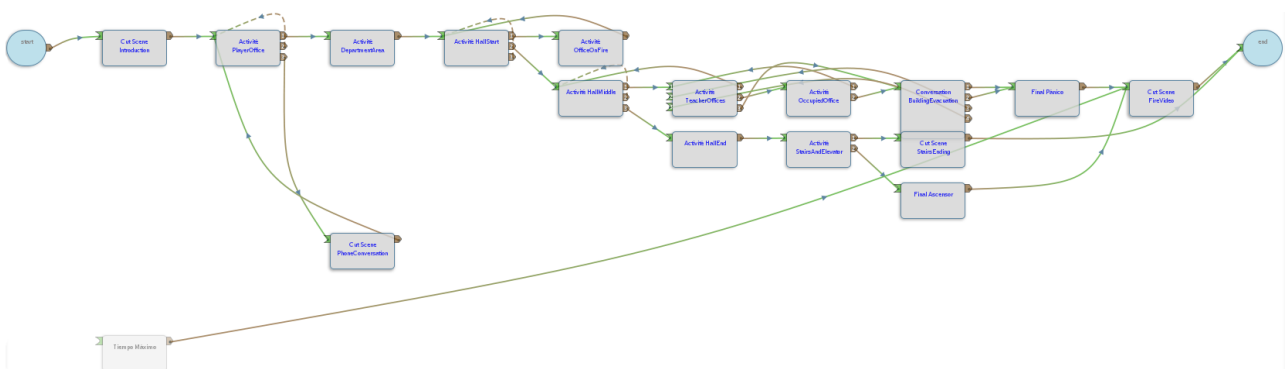


Figure 9. Parcours par défaut du jeu sérieux *Fire Protocol* réalisé avec *eAdventure*, puis importé dans *APPLiq* par transformation de modèle. On remarque une activité plus claire (située en bas du graphe), non connectée à « start » (début du jeu) et donc considérée comme non parcourue dans *MoPPLiq* (ce qui est symbolisé par la couleur plus claire). Il s'agit de la scène de coupe « *Tiempo Máximo* » qui se déclenche dans le jeu à la fin du compte à rebours.

Nous avons aussi constaté que le formalisme de *eAdventure* décrit très peu d'éléments pédagogiques, et notamment que les buts pédagogiques n'y figurent pas. En conséquence, nous ne sommes en mesure d'importer que des objectifs ludiques (« flags » de *eAdventure*).

Ces premiers éléments de mise à l'épreuve nous montrent que *MoPPLiq* est assez robuste pour permettre la modélisation d'une grande partie des jeux sérieux, et qu'il permet aussi de modéliser la plupart des aspects des scénarios issus de deux outils auteurs génériques de jeu sérieux (voir le tableau 1). Ces vérifications nous donnent aussi quelques pistes d'amélioration du modèle, comme la mise en place de méthodologies pour découper les scénarios trans-activités ou comme la prise en compte d'événements trans-activités.







	 Activités <activity>	 États d'entrée <input_state>	 États de sortie <output_state>	 Objectif <goal>
 Legadee	Les activités correspondent aux éléments <screen>	Les états d'entrée correspondent aux points d'arrivée des éléments <Connector>. (Pour chaque élément <screen>, les états d'entrée sont distingués par les <condition> contenus dans les éléments <Connector>)	Les états de sortie correspondent aux points de départ des éléments <Connector>.	Les objectifs travaillés correspondent aux éléments <competency>, <knowledge> et <behaviors> qui sont liés à chaque <screen>. Les objectifs prérequis correspondent aux éléments <condition> des éléments <Connector> entrants.
 eAdventure	Les activités correspondent le plus souvent aux éléments <scene>, mais peuvent aussi correspondre à des scènes de coupe (<slidescene>, <videoscene>, <graph-conversation>) ou à des conversations, quand celles-ci peuvent déclencher une nouvelle scène ou conversation.	Les états d'entrée correspondent aux cibles des éléments <next-scene> (contenus dans des <exit>) et <trigger-xxxx> (où xxxx correspond au type de scène comme « slidescene », « videoscene » ou « conversation »). (Pour un même élément cible, les états d'entrée sont distingués par les attributs DestinyX et Y qui leurs sont associés ou pas les éléments <condition>).	Les états de sortie correspondent à chaque élément <exit> ou <trigger-xxxx> contenus dans une <scene> et qui pointent vers une cible distincte (Les cibles sont distinguées par les éléments <condition> et <effect>, et par les attributs DestinyX et Y).	Les objectifs ludiques correspondent aux « flags » <condition> utilisés pour relier les objets. Les « travaillés » sont contenus dans les éléments <effects>. Les « prérequis » sont contenus dans les éléments <condition>.

Tableau 1. Comparaison du formalisme de MoPPLiq avec ceux de Legadee et eAdventure.

5. Conclusion

Dans cet article, nous avons présenté nos travaux dont le but est de répondre à un besoin des enseignants utilisant des jeux sérieux : pouvoir agir sur la scénarisation pédagogique en fonction des besoins ou du contexte.

Par conséquent, nous avons construit un formalisme capable de représenter la scénarisation pédaogo-ludique des jeux sérieux (*MoPPLiq*), ainsi qu'un mécanisme pour qu'il puisse être compris et manipulé par les enseignants (*APPLiq*). En faisant une analyse de l'existant, nous sommes parvenus à identifier 3 principes pour fonder notre modèle : le découpage pédaogo-ludique, les choix (et donc performances) des apprenants-joueurs et la prise en compte dynamique du modèle de l'apprenant-joueur.

Ce modèle a été expérimenté en s'appuyant sur des jeux sérieux existants et bien connus ce qui a permis de vérifier sa robustesse pour représenter la scénarisation de plusieurs types d'entre eux. En outre, nous avons complété ces expérimentations sur ce modèle et son implémentation en faisant de la transformation de modèles, par l'import de scénarios existants. Les difficultés mises en lumière tracent les perspectives de travail pour mener à terme ces travaux de recherche. Il s'agit notamment de prendre en compte les éléments continus (ex. : compte à rebours, histoire, etc.) qui coexistent avec une scénarisation discontinue (découpée en activités).

6. Bibliographie

[BJÖRK & HOLOPAINEN 05] Björk S., Holopainen J. *Patterns in game design*. Cengage Learning, 2005. 452 p.

- [BOGOST 07] Bogost I. *Persuasive Games: The Expressive Power of Videogames*. Cambridge, UK : MIT Press, 2007. 651 p.
- [DALZIEL 08] Dalziel J. « Using LAMS Version 2 for a game-based Learning Design ». *Journal of Interactive Media in Education*. 19 décembre 2008. Vol. 2008, n°2.
- [EGENFELDT-NIELSEN 04] Egenfeldt-Nielsen S. « Practical barriers in using educational computer games ». *On the Horizon*. 1 mars 2004. Vol. 12, n°1, p. 18- 21.
- [HERNÁNDEZ-LEO et al. 06] Hernández-Leo D., Villasclaras-Fernández E., Jorrín-Abellán I., Asensio-Pérez J., Dimitriadis Y., Ruiz-Requies I., Rubia-Avi B. « Collage, a Collaborative Learning Design Editor Based on Patterns ». janvier 2006.
- [MARFISI-SCHOTTMAN et al. 10] Marfisi-Schottman I., George S., Frank T.-B. « Tools and Methods for Efficiently Designing Serious Games ». In : *Proceedings of ECGBL 2010 The 4th European Conference on Games Based Learning. 4th ECGBL*. Danish School of Education Aarhus University, Copenhagen, Denmark, 2010. p. 226- 234.
- [MARFISI-SCHOTTMAN 12] Marfisi-Schottman I. *Méthodologie, modèles et outils pour la conception de Learning Games*. Thèse de Doctorat en Informatique. Lyon, France : Université Claude Bernard Lyon 1, 2012. 339 p.
- [MARIAIS 12] Mariais C. *Modèles pour la conception de Learning Role-Playing Games en formation professionnelle*. Thèse de Doctorat en Informatique. Université de Grenoble, 2012.
- [MARNE & LABAT 12] Marne B., Labat J.-M. « Implémentation de patrons de conception pour l'adaptation des parcours pédagogique-ludiques dans les jeux sérieux ». In : *Actes du 8ème Colloque Technologies de l'Information et de la Communication pour l'Enseignement. TICE 2012*. Lyon, France, 2012. p. 69- 79.
- [MARTY et al. 12] Marty J., Carron T., Pernelle P. « Observe and react: interactive indicators for monitoring pedagogical sessions ». *International Journal of Learning Technology*. 1 janvier 2012. Vol. 7, n°3, p. 277- 296.
- [MORENO-GER et al. 05] Moreno-Ger P., Martinez-Ortiz I., Fernández-Manjón B. « The <e-Game> Project: Facilitating the Development of Educational Adventure Games ». In : *Proceedings of the IADIS International Conference on Cognition and exploratory learning in digital age (CELDA 2005). Cognition and exploratory learning in the digital age (CELDA 2005)*. Porto, Portugal: IADIS. Porto, Portugal, 2005.
- [MORENO-GER et al. 07] Moreno-Ger P., Sierra J. L., Martínez-Ortiz I., Fernández-Manjón B. « A documental approach to adventure game development ». *Science of Computer Programming*. 1 juin 2007. Vol. 67, n°1, p. 3- 31.
- [SQUIRE 03] Squire K. « Video Games in Education ». *International Journal of Intelligent Simulations and Gaming*. 2003. Vol. 2, p. 49-62.

7. Références sur le WEB.

- [web 1] <http://seriousgames.lip6.fr/DesignPatterns/> consulté en janvier 2013
- [web 2] <http://centerforgamescience.com/site/games/refraction> consulté en janvier 2013
- [web 3] <http://www.scienceenjeu.com/> consulté en janvier 2013
- [web 4] <http://campus-douai.gemtech.fr/course/view.php?id=934> consulté en janvier 2013
- [web 5] <http://www.cellcraftgame.com> consulté en janvier 2013
- [web 6] <http://e-adventure.e-ucm.es/mod/resource/view.php?inpopup=true&id=268> consulté en janvier 2013