



HAL
open science

Génération dynamique de dilemmes en environnement virtuel à partir de modèles de connaissances

Azzeddine Benabbou, Domitile Lourdeaux, Dominique Lenne

► **To cite this version:**

Azzeddine Benabbou, Domitile Lourdeaux, Dominique Lenne. Génération dynamique de dilemmes en environnement virtuel à partir de modèles de connaissances. 8ème Conférence sur les Environnements Informatiques pour l'Apprentissage Humain (EIAH 2017), Jun 2017, Strasbourg, France. hal-01538619

HAL Id: hal-01538619

<https://hal.science/hal-01538619>

Submitted on 13 Jun 2017

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.

Génération dynamique de dilemmes en environnement virtuel à partir de modèles de connaissances

Azzeddine Benabbou, Domitile Lourdeaux et Dominique Lenne

Sorbonne universités, Université de technologie de Compiègne
CNRS UMR 7253 Heudiasyc
57 avenue de Landshut – 60203 COMPIEGNE Cedex
{azzeddine.benabbou – domitile.lourdeaux – dominique.lenne}@hds.utc.fr

Résumé. Ecrire des scénarios pour la formation ou l'entraînement en environnement virtuel nécessite un travail d'écriture très coûteux. Nous proposons donc un système de génération dynamique de scénarios. Dans le cadre du projet MacCoy Critical nous souhaitons former les individus à la gestion des situations critiques. Les dilemmes en font partie. Ils correspondent aux situations où il n'existe pas de bonne solution, c'est-à-dire où l'individu est amené à faire un choix sacrificiant. Nous nous sommes intéressés à la question suivante : comment générer des situations dilemmatiques sans avoir à les identifier et les écrire au préalable ? Dans cet article, nous détaillons notre approche et expliquons comment nous les générons dynamiquement à partir de modèles de connaissances. Nos propos sont illustrés par un exemple en conduite automobile.

Mots-clés. Scénarisation, Dilemme, Modèles de connaissances, Situations critiques

Abstract. Writing scenarios for training in virtual environments represents a huge amount of work. To tackle this issue, we propose a system that generates them dynamically. As part of the project MacCoy Critical, we would like to train individuals to handle critical cases such as dilemma situations. These latter refer to situations where there is no "right" solution, in other words, situations that lead to negative consequences whichever was the choice made by the protagonist. In our work, we are interested in how to generate dilemma situations dynamically without having to script them beforehand. In this article, we present our approach and explain how we generate dilemmas using knowledge models.

Keywords. Scenario orchestration, Dilemma, Knowledge models, critical situations

1 Introduction

Une situation critique peut être définie comme une situation dynamique complexe, où des facteurs internes et externes à l'individu font qu'elle s'écarte des situations a priori maîtrisées. Pour gérer ce genre de situations, l'individu est amené à mobiliser, en plus des compétences « métiers », des compétences transversales dites non-techniques. En effet, dès les années 70, l'analyse des catastrophes aériennes et industrielles a mis en évidence le rôle majeur du facteur humain dans la survenue de ces accidents [1]. Dans

le domaine de la santé, plus de 70% des erreurs médicales sont le résultat d'un problème lié à une ou plusieurs compétences non-techniques [2]. Pour éviter des conséquences désastreuses de se produire, il est donc important d'entraîner les individus aux compétences non-techniques utiles pour gérer ce genre de situations. Dans cette optique, l'apprenant doit être confronté à des situations variées où il devra comprendre son environnement et agir, parfois en urgence, développant ainsi les compétences attendues. Cependant la production de situations propices à l'entraînement n'est pas toujours évidente. En effet, la dangerosité, le manque de ressources et les problèmes d'accessibilité font que les situations d'apprentissage ne sont pas toujours reproductibles en conditions réelles. La réalité virtuelle peut pallier ce problème en offrant des outils et des techniques de simulation permettant d'immerger les personnes dans des environnements virtuels « fidèles » aux représentations réelles. Elle permet un apprentissage situé et constructiviste : l'apprenant restructure ses connaissances par l'expérience, par la confrontation à des situations variées, nouvelles et plus ou moins courantes. Elle permet de le confronter, à des situations d'un niveau de criticité variable en fonction de ses actions et de son profil. Ces situations doivent être suffisamment difficiles tout en restant abordables. Elles doivent permettre de confronter les compétences acquises mais aussi de mettre en jeu des compétences nouvelles et proches de celles acquises pour favoriser le développement [3]. Pour supporter ce type d'apprentissage, il est nécessaire de pouvoir générer un large spectre de scénarios. L'écriture de tels scénarios est un travail conséquent qui conduit, lors du passage à l'échelle, à ce qu'on appelle l'authoring bottleneck [4]. Il est alors nécessaire de mettre en place des systèmes de scénarisation permettant de créer des environnements adaptables, sans avoir à définir explicitement l'intégralité des scénarios possibles. Pour conserver la liberté d'action de l'utilisateur et assurer l'adaptabilité des scénarios, nous faisons l'hypothèse qu'il est possible de générer dynamiquement et automatiquement des situations critiques à partir de modèles de connaissances qui sous-tendent la simulation. Nous nous intéressons au processus de scénarisation de situations critiques en environnement virtuel. La scénarisation est un processus comprenant à la fois la spécification du ou des déroulements possibles ou souhaitables de la simulation, et le contrôle (exécution et/ou suivi et correction) du déroulement des événements en temps interactif. Un système de scénarisation est composé d'un ou plusieurs langages de scénarisation permettant de modéliser le contenu scénaristique et/ou les objectifs scénaristiques, et d'un moteur de scénarisation permettant de gérer de manière dynamique la réalisation du scénario [5]. Le verrou lié à ces travaux naît de l'incompatibilité entre ces différents objectifs : le contrôle s'oppose à l'adaptabilité, la liberté d'action va à l'encontre du contrôle, l'alliance du contrôle et de l'adaptabilité met en péril la cohérence, et ainsi de suite. Notre système de scénarisation d'environnements virtuels doit permettre d'assurer à la fois la liberté d'action de l'apprenant, le contrôle dynamique de la simulation, la cohérence des comportements présentés et l'adaptabilité de l'environnement virtuel.

Dans le cadre de l'entraînement en situation critique, nous avons identifié, avec les membres du projet, plusieurs dimensions critiques : l'ambiguïté, le dilemme, la charge sociocognitive, la fréquence d'occurrence des événements, la gravité, la nouveauté et la maîtrise [6]. Dans cet article, nous présentons nos travaux sur la génération dynamique de situations mettant en jeu l'une de ces dimensions : le dilemme. Nous exposons tout d'abord quelques travaux connexes avant de présenter notre architecture globale. Nous détaillons ensuite notre contribution et l'implémentation de nos travaux.

2 Travaux connexes

Plusieurs travaux de l'ICT¹ [7] implémentent le scénario suivant : jouant le rôle d'un lieutenant de l'armée américaine, l'utilisateur doit venir en renfort à un peloton en difficulté. Pendant le trajet, le lieutenant et sa troupe passent par un village où ils croisent le chemin d'un garçon blessé en situation critique. Un véhicule de la troupe du lieutenant est impliqué dans l'accident. Le lieutenant est confronté à l'alternative suivante : poursuivre son chemin pour venir en renfort à son peloton ou sécuriser une zone d'atterrissage pour permettre à un hélicoptère médical de se poser et venir en secours au garçon blessé. Gratch et Marsella [8] modélisent le comportement émotionnel d'un médecin face au cas médical classique de l'accompagnement des mourants par l'administration de médicaments. Dans le scénario présenté, le médecin souhaite prolonger le plus possible la vie de son patient âgé de 11 ans. La famille du patient quant à elle pourrait s'opposer à cette décision à cause de la souffrance que le patient va devoir endurer. Que fera le médecin dans ce cas-là ? Céder aux exigences de la famille ou ignorer leur décision et accomplir son devoir ? Nous trouvons aussi dans la littérature des travaux sur le célèbre dilemme du tramway. Dans sa version originale, énoncé par Philippa Foot [9], un tramway roulant sur la voie A se dirige droit sur cinq ouvriers. La seule issue possible est de dévier le tramway de sa trajectoire et le diriger vers la voie B où se trouve une seule personne. Que faire dans ce cas ? Sacrifier la personne sur la voie B pour sauver les cinq autres ou ne pas intervenir et laisser le tramway suivre son chemin et tuer les cinq personnes ? Ce dilemme et ses variantes ont fait l'objet de plusieurs études sur papier [10, 11], et en environnements virtuels [12, 13]. Dans les travaux cités précédemment, les dilemmes sont écrits en amont de l'exécution de la simulation. Cette approche scriptée permet de décrire d'une manière fine et précise les situations dilemmatiques.

A l'opposé, nous trouvons les approches génératives, qui consistent à générer dynamiquement des dilemmes au fur et à mesure qu'on avance dans la simulation. Nous avons identifié un seul système qui adopte une telle approche. GADIN [14] est un moteur narratif interactif, qui confronte l'utilisateur, au fur et à mesure qu'il avance dans l'histoire, à des situations de dilemmes en utilisant des techniques de planification. L'histoire évolue en fonction des décisions de l'utilisateur face à ces dilemmes. Afin de les générer, les auteurs proposent cinq catégories distinguées selon les conséquences qu'aura l'action de l'utilisateur sur lui, ses amis et/ou ses ennemis : « Trahison », « Sacrifice », « Bien commun », « Mal commun » et « Faveur ».

L'approche scriptée permet de décrire en amont d'une manière fine et précise le dilemme à présenter à l'utilisateur. Bien que cette approche soit pertinente pour des expérimentations en sciences humaines ou pour l'entraînement à des situations bien spécifiques, elle reste non adaptée à nos besoins d'entraînement en situations critiques. En effet, nous souhaitons confronter l'apprenant à diverses situations de dilemmes qui varient en fonction des consignes pédagogiques. La nécessité d'une variabilité de situations ainsi que la liberté d'action de l'utilisateur dans l'environnement virtuel rend difficile voire impossible l'écriture exhaustive de toutes les situations de dilemmes. Pour remédier à ce problème, GADIN propose une approche générative. Cependant, il présente deux limites majeures. Premièrement, le système n'est pas implémenté dans un

¹ University of Southern California

environnement virtuel et la liberté d'action de l'utilisateur est réduite. Deuxièmement, toutes les catégories proposées dépendent nécessairement des relations sociales (amies et ennemies) qu'entretient l'utilisateur avec les autres personnages. Il est donc impossible de générer un dilemme si l'utilisateur est seul dans l'environnement ou n'entretient pas de relations particulières avec les autres personnages. L'approche que nous proposons permet de générer dynamiquement des dilemmes sans avoir à les écrire en amont. Elle permet d'ajuster la criticité au fur et à mesure de la simulation, et donc d'adapter la nature des dilemmes à présenter, en fonction des actions de l'utilisateur et de son profil. De plus, elle repose sur un modèle de dilemme qui ne se restreint pas aux relations sociales qu'entretient l'utilisateur avec les autres. Dans la suite, nous présentons l'architecture globale qui décrit l'ensemble des modules du projet, puis nous détaillons notre approche sur la génération dynamique des dilemmes.

3 Architecture globale

La présente l'ensemble des modules de notre architecture. La génération des dilemmes se fait *via* le module de « spécification du scénario ». Ce module cohabite avec un « planner » au sein du système de scénarisation. Ce dernier reçoit en entrée des consignes pédagogiques de la part du module du « diagnostic de l'apprenant ». Ces consignes sont composées 1) d'intentions pédagogiques comme vérifier, renforcer, déstabiliser une ou des compétence(s) (non-)technique(s) et 2) un degré de criticité à respecter. Elles sont traduites en objectifs scénaristiques par le module de « spécification de scénario », qui les envoie ensuite au « planner ». Ce dernier communique directement avec l'environnement virtuel et réalise les ajustements adéquats afin de diriger la simulation vers la situation qui respecte les objectifs scénaristiques. L'environnement virtuel et le module de diagnostic sont gérés par nos partenaires dans le projet.

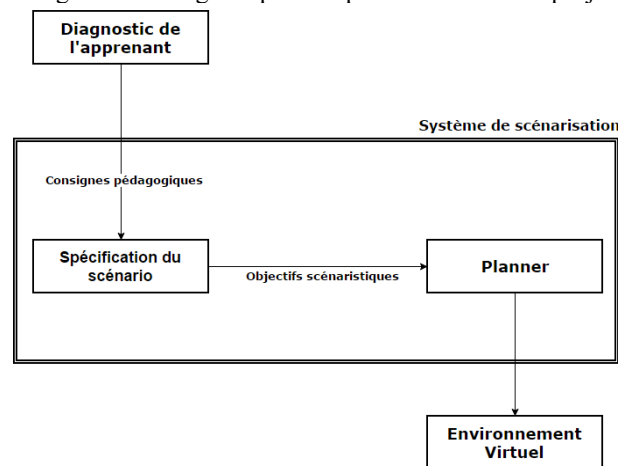


Figure 1 - Architecture globale

4 Génération des dilemmes

Dans la littérature, un dilemme est défini comme une situation où il n'existe pas de « bonne » solution. C'est une situation où l'individu est confronté à un choix difficile, sacrificiant, mettant en conflit dans certains cas, ses valeurs. Ce genre de situations peut être identifié par un humain. Par conséquent, un formateur, disposant d'une interface appropriée, pourrait facilement identifier une situation dilemmatique et la proposer à l'apprenant durant sa session d'entraînement. Cependant dans une approche de scénarisation automatique, ce processus d'identification et de sélection de situations doit se faire automatiquement par un système informatique et cette identification est difficile sans sémantique. Nous nous sommes alors intéressés aux questions suivantes : Comment modéliser informatiquement un dilemme ? Comment construire une sémantique associée ? Et comment permettre à un moteur de scénarisation de générer dynamiquement un dilemme à partir de cette sémantique ? La définition du dilemme évoquée ci-dessus met l'accent sur deux propriétés importantes auxquelles répondent les situations dilemmatiques : (1) Le choix présenté à l'utilisateur est *exclusif*; (2) les conséquences sont toujours *négatives*. Nous pensons donc qu'en identifiant des situations respectant ces deux propriétés, notre système de scénarisation serait capable de générer dynamiquement des situations de dilemmes.

Pour répondre à la première propriété, nous proposons de générer des situations qui mettent en jeu des actions contradictoires. Deux actions sont contradictoires, si elles sont opposées/incompatibles et sollicitées simultanément. Pour pouvoir les identifier, nous devons disposer d'un modèle de représentation de tâches qui permet de renseigner les relations temporelles et d'inférer les liens sémantiques entre elles. De plus, nous devons être capables de savoir si la réalisation ou non d'une tâche entraîne des conséquences négatives, afin de respecter la deuxième propriété. Pour ce faire, nous devons disposer d'un modèle qui permet de représenter, à différents niveaux d'abstraction, les causalités entre les événements ainsi que les tâches qui pourraient prévenir leur survenue. Dans la section suivante, nous présentons ces modèles et nous les illustrons par des exemples.

4.1 Modèles de connaissances

Dans cette partie, nous présentons notre modèle de connaissances. Il comporte trois parties : un modèle du monde, un modèle de tâches et un modèle de causalité.

4.1.1 Modèle du monde

Le modèle du monde est destiné à être conçu par les experts du domaine qui ne sont pas nécessairement informaticiens. Il doit donc être intelligible tout en étant interprétable par un système informatique. Ce modèle doit en plus offrir une large expressivité. Il doit permettre de renseigner les objets du monde, leurs propriétés ainsi que les relations entre eux, tout en permettant de raisonner à différents niveaux d'abstraction afin de permettre une plus grande variabilité de situations. Pour répondre à ces besoins,

notre modèle [15] s'appuie sur une représentation ontologique. Il permet d'interroger la base de connaissances du monde afin d'extraire des informations pertinentes pour la scénarisation (ex. Quels sont tous les feux de circulation qui se trouvent, au plus, à un rayon de 100m du véhicule de l'apprenant ?). Ces connaissances peuvent être couplées à des règles de fonctionnement pour permettre de gérer dynamiquement l'évolution du monde. Un exemple de ce dernier est illustré dans la Figure 2.

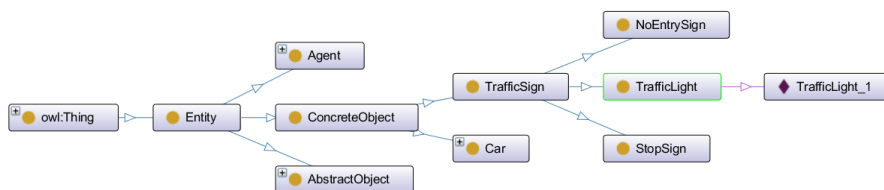


Figure 2 - Exemple d'une partie du modèle du monde

4.1.2 Modèle de tâches

Le modèle de tâches est destiné à être rempli par des experts ergonomes. Par conséquent, il est utile qu'il repose sur une représentation qui opérationnalise des principes issus de langages utilisés en ergonomie cognitive [15, 16]. Par exemple, dans certaines représentations hiérarchiques, les tâches mères sont composées de sous-tâches filles sur plusieurs niveaux. Ces dernières sont liées par des constructeurs qui permettent de renseigner les relations logiques et temporelles entre elles. Les tâches peuvent avoir des préconditions et des postconditions exprimées, par exemple, par des agrégats d'assertions sur le monde sous forme : (sujet prédicat objet). Cette formulation nous paraît très pertinente pour un couplage avec une représentation ontologique du monde. Certaines représentations distinguent, entre autres, les préconditions contextuelles et les préconditions favorables. Les premières sont les conditions qui rendent pertinente la réalisation de la tâche. Les dernières sont les conditions qui rendent la réalisation de la tâche préférable à d'autres. Les postconditions, quant à elles, appelées aussi conditions de satisfaction, renseignent sur l'état que le monde doit satisfaire pour qu'une tâche soit considérée comme réalisée. La Figure 3 montre une partie d'un modèle qui décrit la tâche « gérer feu rouge » composée de deux sous-tâches.

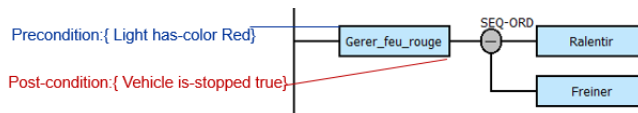


Figure 3 - Extrait d'un modèle de tâches (gérer feu rouge)

4.1.3 Modèle de causalité

Le modèle de causalité est un graphe acyclique et orienté qui exprime les chaînes de causalité pertinentes de l'environnement. Les nœuds du graphe font référence aux événements. Ils sont reliés entre eux par des liens de subsomption ou de causalité. Le déclenchement d'un événement peut être empêché par une barrière de prévention qui peut être une tâche du modèle de tâches. La Figure 4 montre un exemple d'un modèle de causalité qui décrit le risque d'accident suite à un aquaplaning. La perte de contrôle du véhicule peut être évitée grâce à la barrière « gérer aquaplaning » qui correspond à une tâche dans le modèle de tâches.

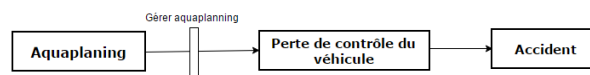


Figure 4 - Exemple d'un modèle de causalité

4.2 Génération dynamique des dilemmes

Rappelons que pour la génération des dilemmes, nous avons identifié deux propriétés importantes : le choix présenté à l'apprenant doit être exclusif et les conséquences doivent être négatives quel que soit le choix. Dans la suite, nous expliquons comment nous extrayons les tâches ayant des conséquences négatives depuis le modèle de causalité. Ensuite nous montrons comment nous générons des situations à choix exclusif qui mettent en jeu les tâches sélectionnées.

La modélisation à différents niveaux d'abstraction, que permet le modèle de causalité, nous permet de distinguer les événements qui conduisent à des conséquences négatives des autres. En ce qui concerne les conséquences négatives, nous distinguons des conséquences en terme de :

- gravité : blessure, nombres de victimes ou de dégâts matériels (ex. percuter un piéton, un véhicule, un arbre ...)
- normes/lois/valeurs : violation des lois, non-respect des consignes, normes et réglementations (ex. griller un feu rouge) ;
- score : perte de points (ex. points de vie, points de performance).

Ce qui nous intéresse précisément ce ne sont pas les événements en eux-mêmes, mais les tâches dont la (non)-réalisation provoque ces événements. Les barrières de prévention, qui sont des tâches, peuvent fournir cette information. En effet, la non-réalisation d'une barrière entraîne le déclenchement des événements postérieurs à celle-ci. Nous sélectionnons donc toutes les barrières de prévention, dont les événements postérieurs conduisent à un ou plusieurs type(s) de conséquence cité(s) plus haut. La Figure 5 illustre nos propos. Selon la consigne pédagogique, nous pouvons affiner l'ensemble des tâches sélectionnées en se restreignant à un type de conséquence en particulier et/ou en respectant un degré minimum ou maximum de criticité (ex. gravité_maximum = 2, point_de_performance = -5).

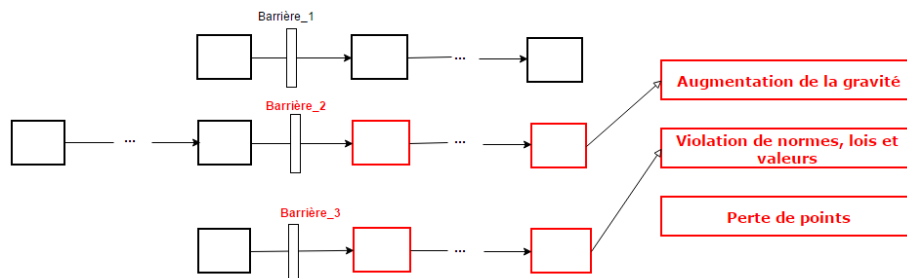


Figure 5 - Sélection de barrières

Lors d'un choix exclusif, l'alternative présentée à l'utilisateur, implique forcément la non-réalisation d'une des deux tâches requises, et par conséquent le déclenchement systématique des conséquences négatives qui lui sont rattachées. Afin de générer des situations où le choix présenté à l'utilisateur est exclusif, nous devons générer des situations où l'utilisateur est amené à réaliser deux tâches contradictoires. La contradiction survient lorsque deux tâches incompatibles sont sollicitées simultanément. Nous distinguons deux types d'incompatibilité entre les tâches :

- Nomologique : se réfère aux actions incompatibles par nature (ex. avancer/reculer, augmenter/diminuer une dose de médicament, ouvrir/fermer une porte).
- Scénaristique ou réglementaire : se réfère à toutes les actions qui sont nomologiquement compatibles mais qui, dans un contexte particulier, ne le sont plus suite à une contrainte scénaristique ou réglementaire (lois, normes, ordre, consigne...). Par exemple, les actions « parler au téléphone » et « conduire » ne sont pas opposées par nature. Cependant, dans un contexte particulier (le code de la route en France par exemple), les deux actions deviennent incompatibles.

Cependant, pour la génération d'une situation à choix exclusif, seul le premier type d'incompatibilité peut être retenu. En effet, lorsque deux actions sont incompatibles du point de vue scénaristique ou réglementaire, rien n'empêche pour autant l'individu d'accomplir les deux. Le choix présenté n'est donc plus exclusif.

L'extraction de tâches nomologiquement opposées se fait à partir du modèle de tâches. Ces tâches doivent de plus répondre à deux critères pour que l'instanciation d'un état du monde propice à leur réalisation soit possible : elles doivent être contextuellement et temporellement compatibles. La Figure 6 présente les étapes de la génération de dilemmes.

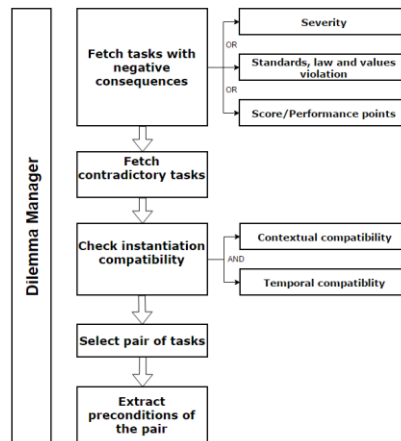


Figure 6 - Modèle de génération de dilemmes

Tout d'abord, nous parcourons, deux par deux, l'ensemble des tâches sélectionnées depuis le modèle de causalité (tâches dont la non-réalisation conduit à des conséquences négatives). Pour chaque paire de tâches, nous examinons leurs conditions de satisfaction (postconditions). Si elles sont incompatibles, les tâches le sont aussi. Deux conditions, sont considérées comme incompatibles, si pour un même couple (sujet prédicat) l'objet est différent.

Les paires de tâches doivent être contextuellement compatibles. Pour cela, nous examinons les préconditions contextuelles des paires de tâches sélectionnées. Si elles sont incompatibles, la paire de tâches est alors écartée.

Exemple :

Tâche 1 : « Open_door » ; précondition : « door has-state close » ; postcondition : « door has-state open ».

Tâche 2 : « Close_door » ; précondition « door has-state open » ; postcondition : « door has-state-closed ».

Les tâches « open_door » et « close_door » sont nomologiquement incompatibles (analyse des postconditions), par conséquent elles peuvent potentiellement servir à générer un dilemme. Cependant, elles sont contextuellement incompatibles (analyse des préconditions), car l'instanciation d'une telle situation nécessite que la porte soit fermée et ouverte en même temps, ce qui est impossible. La paire de tâches est donc écartée.

Par ailleurs, les tâches doivent être temporellement compatibles. C'est-à-dire que la réalisation d'une tâche ne doit pas dépendre de la réalisation préalable de l'autre. Dans notre modèle de tâches, l'incompatibilité temporelle entre deux tâches se réfère à celles qui ont un ancêtre commun dont le constructeur temporel est séquentiel. Par conséquent, seules les tâches temporellement indépendantes ou parallèles sont retenues.

Ensuite, nous sélectionnons une paire dans la liste des tâches retenues. A ce stade de nos travaux, la sélection est aléatoire. Nous pensons dans la suite des travaux attribuer un score à chaque paire de tâches, calculé à partir de différents facteurs (préconditions favorables, degrés de gravité des conséquences, consignes pédagogiques...). Ce score nous permettra d'ordonner les tâches par préférence « scénaristique » et/ou pédagogique.

Finalement nous extrayons les préconditions de la paire de tâches sélectionnée. Ces préconditions constituent un état partiel du monde. Cet état est ensuite transmis à l'environnement virtuel afin d'instancier la situation.

5 Implémentation et exemple

L'approche décrite ci-dessous a fait l'objet d'une première implémentation. Nous avons développé une première version de notre module de génération de dilemme. Nous avons aussi développé un environnement virtuel de test, pour la conduite automobile, sous Unity. Nous avons utilisé le modèle de tâches présenté dans la Figure 7. Le modèle décrit trois tâches indépendantes qui sont « gérer_aquaplaning », « gérer_feu_rouge » et « gérer_stop ». Par ailleurs, nous avons utilisé le modèle de causalité présenté dans la Figure 8. Le modèle décrit trois événements centraux : « griller stop », « griller feu rouge » et « perte de contrôle du véhicule » qui peuvent survenir en cas de non réalisation des barrières. L'exécution de l'algorithme sur les modèles retourne deux paires de tâches candidates : ({Gérer_aquaplaning, Gérer_stop}, {Gérer_aquaplaning, Gérer_feu_rouge}). En effet, les tâches de chacune des paires sont nomologiquement incompatibles, et contextuellement et temporellement compatibles. Elles peuvent donc générer une situation de dilemme. Comme expliqué précédemment, à ce stade des travaux, l'algorithme sélectionne aléatoirement une paire de tâches. Si l'on suppose que la paire de tâches choisie est la suivante : {Gérer_aquaplaning, Gérer_feu_rouge}, l'algorithme extrait ensuite leurs préconditions : {(Vehicle has-state aquaplaning) AND (Light has-color Red)}. Cet état du monde est ensuite envoyé au moteur de planification qui se charge de diriger la simulation vers une situation où le véhicule de l'apprenant est en aquaplaning et où le feu est rouge. Le dilemme dans cette situation est : Faut-il freiner pour s'arrêter au feu rouge au risque de perdre le contrôle du véhicule ? Une vidéo de démonstration est disponible sur le lien en bas de page².

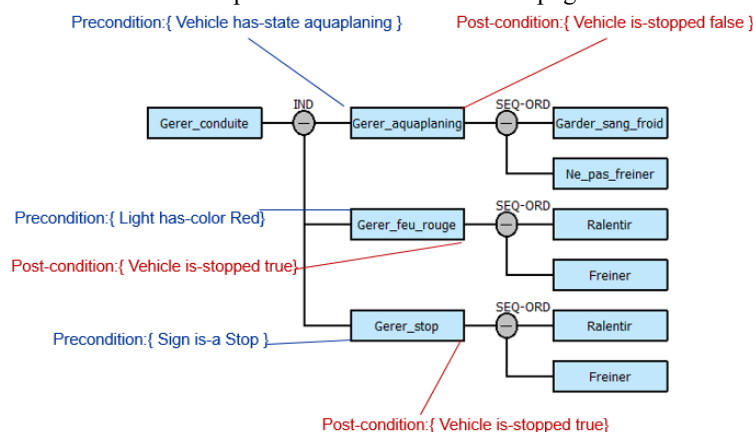


Figure 7 - Modèle de tâches simplifié (conduite automobile)

² <https://www.youtube.com/watch?v=ccNQ7brKa0M>

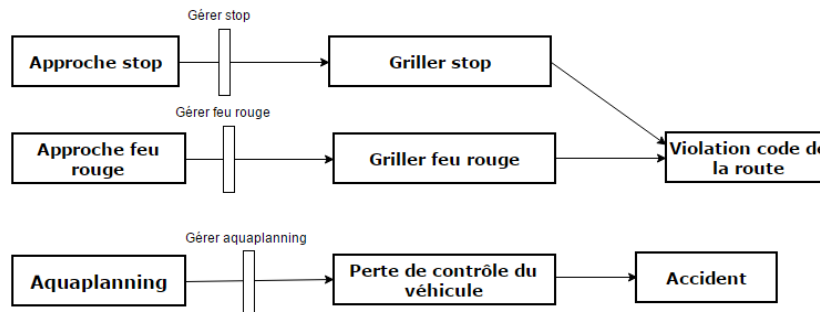


Figure 8 - Extrait du modèle de causalité pour la conduite

6 Discussion et conclusion

La génération dynamique des situations d'apprentissage constitue une solution intéressante pour palier le problème de « l'authoring bottleneck ». Dans le cadre du projet, nous adoptons cette approche pour la génération de la criticité. Nous avons identifié avec nos partenaires différentes dimensions de criticité dont le dilemme. Cette notion nous a confrontés à la question de recherche suivante : « Comment générer des situations de dilemme sans avoir à les écrire à l'avance ? ». Nous avons proposé une approche qui repose sur la génération dynamique à partir de modèles de connaissances. Cette approche a fait l'objet d'une première implémentation qui a montré des résultats encourageants. Nous travaillons actuellement sur l'écriture de modèles de connaissances plus fournis afin de pouvoir générer suffisamment de situations de dilemme pour faire des expérimentations. Nous envisageons par la suite, l'intégration des dilemmes moraux une fois que nous aurons identifié le modèle théorique le plus pertinent pour un contexte d'apprentissage. Nous tenons aussi à préciser que les travaux présentés dans cet article concernent la partie de spécification d'objectifs scénaristiques (module spécification du scénario) seulement. Il est important de souligner le rôle majeur que joue le « planner » dans le processus de génération des dilemmes. En effet, la temporalité des évènements est très importante car elle pourrait mettre en péril le dilemme. Dans l'exemple présenté précédemment, le dilemme est mis en péril si le feu ne passe au rouge qu'après le passage du véhicule.

Remerciement. Ces travaux de recherche s'inscrivent dans le cadre du projet ANR MacCoy Critical (n° ANR CE14-24 0021).

Références

1. Brami, J., & Amalberti, R. (2010). *La sécurité du patient en médecine générale*. Springer Science & Business Media.
2. Flin, R., Winter, J., & Cakil Sarac, M. R. (2009). Human factors in patient safety: review of topics and tools. *World Health, 2*.

3. Vygotsky, L. S. (1978). *Mind in Society*. Cambridge, {MA}: Harvard University Press.
4. Spierling, U., & Szilas, N. (2009). Authoring issues beyond tools. In *Interactive Storytelling* (pp. 50–61). Springer.
5. Barot, C. (2014). *Scénarisation d'environnement virtuel. Vers un équilibre entre contrôle, cohérence et adaptabilité*. Université de Technologie de Compiègne, Compiègne.
6. Burkhardt, J.-M., Corneloup, V., Garbay, C., Bourrier, Y., Jambon, F., Luengo, V., ... Lourdeaux, D. (2016). Simulation and virtual reality-based learning of non-technical skills in driving: critical situations, diagnostic and adaptation. *IFAC-PapersOnLine*, 49(32), 66–71.
7. Rickel, J., Marsella, S., Gratch, J., Hill, R., Traum, D., & Swartout, W. (2002). Toward a new generation of virtual humans for interactive experiences. *IEEE Intelligent Systems and Their Applications*, 17(4), 32–38.
8. Gratch, J., & Marsella, S. (2004). A Domain-independent Framework for Modeling Emotion. *Journal of Cognitive Systems Research*, 5(4), 296–306.
9. Foot, P. (1967). The problem of abortion and the doctrine of double effect.
10. Valdesolo, P., & DeSteno, D. (2006). Manipulations of Emotional Context Shape Moral Judgment. *Psychological Science*, 17(6), 476–477.
11. HAUSER, M., CUSHMAN, F., YOUNG, L., KANG-XING JIN, R., & MIKHAIL, J. (2007). A Dissociation Between Moral Judgments and Justifications. *Mind & Language*, 22(1), 1–21.
12. Navarrete, C. D., McDonald, M. M., Mott, M. L., & Asher, B. (2012). Virtual morality: emotion and action in a simulated three-dimensional “trolley problem”. *Emotion (Washington, D.C.)*, 12(2), 364–70.
13. Skulmowski, A., Bunge, A., Kaspar, K., & Pipa, G. (2014). Forced-choice decision-making in modified trolley dilemma situations: a virtual reality and eye tracking study. *Frontiers in Behavioral Neuroscience*
14. Barber, H. (2008). Generator of Adaptive Dilemma-based Interactive Narratives, (October), 1–18.
15. Van Der Veer, G. C., Lenting, B. F., & Bergevoet, B. A. J. (1996). GTA: Groupware task analysis - Modeling complexity. *Acta Psychologica*, 91(3 SPEC. ISS.), 297–322.
16. Sebillotte, S., & Scapin, D. L. (1994). From users' task knowledge to high-level interface specification. *International Journal of Human-Computer Interaction*, 6(1), 1–15.