Génération dynamique de situations critiques en environnements virtuels : dilemme et ambiguïté

Azzeddine Benabbou

▶ **To cite this version:**

Azzeddine Benabbou. Génération dynamique de situations critiques en environnements virtuels : dilemme et ambiguïté. Intelligence artificielle [cs.AI]. Université de Technologie de Compiègne, 2018. Français. NNT : 2018COMP2432 . tel-02615405v3

HAL Id: tel-02615405

https://theses.hal.science/tel-02615405v3

Submitted on 22 May 2020

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.

Par **Azzeddine BENABBOU**

Génération dynamique de situations critiques en environnements virtuels : dilemme et ambiguïté

Thèse présentée
pour l'obtention du grade
de Docteur de l'UTC

Soutenue le 13 septembre 2018
Spécialité : Informatique : Unité de recherche Heudyasic (UMR-7253)

D2432

Génération dynamique de situations critiques en environnements virtuels : dilemme et ambiguïté

Azzeddine Benabbou

Soutenue le 13 septembre 2018 devant le jury composé de :

Mme V. LUENGO (Présidente)
Mme D. LOURDEAUX (Directrice de thèse)
M. D. LENNE (Directeur de thèse)
M. R. QUERREC (Rapporteur)
M. F. VANDERHAEGEN (Rapporteur)
M. G. BONNET
M. N. SZILAS
Mme I. THOUVENIN

Département Génie Informatique
Université de Technologie de Compiègne

Résumé

Nos travaux traitent de la génération dynamique de situations critiques en environnements virtuels. Parmi les dimensions qui caractérisent les situations critiques, nous nous intéressons particulièrement aux dilemmes et aux situations ambiguës. L'enjeu informatique de nos travaux consiste à générer automatiquement des dilemmes et des situations ambiguës à partir de modèles qui ne sont pas destinés, a priori, à décrire ces deux types de situations.

Un dilemme est souvent défini comme étant une situation à choix difficile, où l'individu doit choisir entre deux options, ou plus, qui présentent toutes des inconvénients. L'ambiguïté quant à elle, se réfère aux situations qui sont sujettes à plusieurs interprétations. Dans le cadre de nos travaux, nous proposons tout d'abord un modèle formel de ces deux notions basé sur des théories issues des sciences humaines et sociales. A partir de cette formalisation, nous proposons un ensemble d'algorithmes et de techniques de génération qui raisonnent sur des modèles de connaissances — manipulés par les experts du domaine —– non destinés, a priori, à la description des dilemmes ou des situations ambiguës. Le raisonnement sur ces modèles sémantiques permet d'inférer de nouvelles connaissances. Elles permettent d'identifier les entités (e.g. actions, événements, objets) susceptibles de créer, lorsqu'elles sont combinées ensemble, des situations qui vérifient les propriétés énoncées dans la formalisation des dilemmes et de l'ambiguïté. Afin de proposer un contenu adapté à chaque apprenant, pour la génération des dilemmes moraux, nous proposons d'opérationnaliser la théorie des valeurs universelles de (Schwartz, 2006), afin de prendre en compte le système de valeurs propre à chaque individu. Par ailleurs, pour la génération de l'ambiguïté, il est nécessaire de prendre en compte le degré de connaissance de l'apprenant des variables du monde. Nous proposons alors une modélisation de la représentation mentale de l'apprenant. Afin de considérer les incertitudes et les conflits dans cette représentation, nous proposons l'utilisation du cadre formel des fonctions de croyances qui est adapté à ce besoin.

Abstract

Our work is related to dynamic generation of critical situations in virtual environments. We focus on two particular types of critical situations : dilemmas and ambiguous situations. The challenge of this work is to generate automatically these situations using knowledge models that are not destined to describe dilemmas and ambigous situations.

A dilemma is commonly defined as a situation that includes a difficult choice. It refers to a situation where individuals have to choose between two, or more, inconvenient options. Ambiguity refers to situations that can be interpreted in different ways. In this work, we propose a formalization, inspired by humanities and social sciences, of these two notions. Using this formalization, we propose a set of algorithms and generation technics that use knowledge models — manipulated by domain experts — that are not destined to decribe dilemmas and ambiguous situations. The use of these models enables the generation engine to infer new knowledge used to extract the entities (e.g. actions, events, objects) that can potentially produce situations that meet the properties defined in the dilemma and ambiguity formalization. Furthermore, in order to propose an adapted content to each learner, it is necessary to take into consideration the values system of each person in the dilemma generation process. Thus, we propose to operationnalize the theory of universal values of (Schwartz, 2006). Concerning the ambiguity, it is necessary to take into account the level of knowledge of the learner regarding the world variables. Thus, we propose to model the mental representation of the learner. Besides, in order to consider the uncertainties in this representation, we propose to use the belief functions theory that is well-suited to this matter.

Table des matières

1 Introduction **11**

1.1 Contexte 12

1.2 Compétences non-techniques et situations critiques 13

1.2.1 La criticité 13

1.2.2 Compétences non-techniques 13

1.3 Les environnements virtuels pour l'apprentissage 15

1.3.1 Simulation et environnements virtuels 15

1.3.2 La scénarisation en environnements virtuels 16

1.4 Problématique de recherche et objectifs 18

1.5 Approche 19

1.6 Contexte technologique et applicatif 20

1.6.1 Architecture du projet 20

1.6.2 Plateforme Humans 22

1.7 Organisation du mémoire 22

2 Etat de l'art **25**

2.1 La criticité 26

2.1.1 Critique au sens crucial 27

2.1.2 La criticité chiffrée 27

2.1.3 La criticité non chiffrée 29

2.1.4 Criticité : approche multidimensionnelle 31

2.1.5 Bilan 32

2.2 Les dilemmes en théorie 33

2.2.1 Définition générale 33

2.2.2 Les dilemmes moraux 34

2.2.3 Les valeurs morales 35

2.2.4 Les dilemmes, instruments de recherche et de formation 36

2.2.5 Bilan 38

2.3 Dilemme dans les simulations 38
2.3.1 Les approches scriptées 39
2.3.2 Les approches génératives 40
2.3.3 Bilan et positionnement 43
2.4 L'ambiguïté en théorie 44
2.4.1 Ambiguïté et probabilités 44
2.4.2 Ambiguïté et incertitudes 46
2.4.3 Ambiguïté, contexte et relations 48
2.4.4 Bilan 49
2.5 L'ambiguïté dans les simulations 50
2.6 Bilan 53

3 Contribution **55**
3.1 Architecture 56
3.1.1 Modules externes 56
3.1.2 Modules internes 58
3.1.3 Modèles sémantiques pour la scénarisation 61
3.1.4 Bilan 69
3.2 KOBA : moteur de génération automatique de dilemmes 70
3.2.1 Les conséquences négatives 72
3.2.2 Le profil moral 75
3.2.3 Formalisation des types de dilemmes 77
3.2.4 Les étapes de génération 81
3.2.5 Bilan 97
3.3 Génération de l'ambiguité 97
3.3.1 Approche 98
3.3.2 Représentation mentale incertaine de l'apprenant 98
3.3.3 Génération de l'ambiguïté de l'information 102
3.3.4 Génération de l'ambiguïté du contexte 106
3.3.5 Bilan 109

4 Implémentation **111**
4.1 Moteur de génération de dilemmes 111
4.1.1 Les modèles de connaissances 111
4.1.2 Le moteur de génération KOBA 113
4.1.3 Le profil moral 114
4.2 Preuve de concept 114

4.2.1 Présentation de l'environnement virtuel 114
4.2.2 Les modèles utilisés 115
4.2.3 Les étapes de génération 115

5 Évaluations 119
5.1 Évaluation préliminaire 119
5.2 Évaluation en environnement virtuel 121
5.2.1 Protocole de l'évaluation 122
5.2.2 Résultats 124
5.2.3 Interprétation des résultats 126

6 Conclusion et limites 129
6.1 Résumé des travaux 129
6.2 Limites et perspectives 130
6.2.1 Génération des dilemmes 130
6.2.2 Génération de l'ambiguïté 132
6.2.3 L'aspect apprentissage : limites et discussions 134

Annexe A Glossaire 139

Annexe B Les méta-modèles 141

Annexe C Capture d'écrans des scénarios de l'évaluation 143

Table des figures 147

Liste des tableaux 149

Publications 151

Bibliographie 153

Chapitre 1

Introduction

Sommaire

1.1 Contexte 12
1.2 Compétences non-techniques et situations critiques 13
1.2.1 La criticité 13
1.2.2 Compétences non-techniques 13
1.3 Les environnements virtuels pour l'apprentissage 15
1.3.1 Simulation et environnements virtuels 15
1.3.2 La scénarisation en environnements virtuels 16
1.4 Problématique de recherche et objectifs 18
1.5 Approche 19
1.6 Contexte technologique et applicatif 20
1.6.1 Architecture du projet 20
1.6.2 Plateforme Humans 22
1.7 Organisation du mémoire 22

Il arrive, lors de certains événements professionnels ou de la vie courante, que les personnes soient confrontées à des situations critiques. Celles-ci sont définies comme étant des situations dynamiques et complexes où des facteurs tant internes (e.g. état de fatigue, manque de concentration) qu'externes (conditions climatiques dégradées, incompétences des collègues) à l'individu font qu'elles s'écartent des situations a priori maîtrisées. Pour gérer ce genre de situations, et éviter par la suite des conséquences parfois désastreuses, il est nécessaire d'entraîner les agents concernés à réagir dans ce type de situations. Ils doivent être confrontés pour cela à une variété de scénarios où ils devront comprendre leur environnement et agir, parfois en urgence, développant ainsi les compétences nécessaires. Dans le cadre de cette thèse, nous nous intéressons à deux types de scénarios critiques en particulier : les

dilemmes et les situations ambiguës. Notre ambition est de générer dynamiquement ces deux types de scénarios en environnement virtuel (EV) sans avoir à les décrire a priori.

1.1 Contexte

Ces travaux de recherche s'intègrent dans le cadre du projet national MACCOY CRITICAL [1]. Ce projet multidisciplinaire vise à étudier et à améliorer les dispositifs tels que la simulation et les environnements virtuels pour l'entraînement aux compétences non-techniques (CNT) dans les situations critiques. Dans le cadre de nos travaux de thèse, notre objectif est de générer automatiquement ces situations critiques, en s'intéressant particulièrement aux situations dilemmatiques et ambiguës.

La production de situations d'entraînement n'est pas toujours évidente. En effet, la dangerosité, le manque de ressources et les problèmes d'accessibilité font que les scénarios 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 pouvant reproduire des situations proches de celles rencontrées en conditions réelles. Elle permet, entre autres, 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, dans le cas d'environnements adaptatifs, 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 (Vygotsky, 1978). Nous avons choisi le courant constructiviste car, d'une part, nous sommes convaincus par le contenu de sa théorie [2], et d'autre part, car il soulève des questions de recherche intéressantes, dont la plus saillante est la suivante : est-il possible de générer automatiquement un panel de situations qui soit à la fois varié et pertinent pour l'apprenant, et qui favorise le développement des compétences ciblées ? Pour supporter ce type d'apprentissage, il est donc nécessaire de 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 (Spierling and Szilas, 2009). Nous souhaitons alors mettre en place un système de scénarisation permettant de créer des environnements adaptables, sans avoir à définir explicitement l'intégralité des

1. https ://maccoy.hds.utc.fr/
2. Nous pensons effectivement que l'apprentissage se **construit** plutôt qu'il ne se transmet, et qu'il découle plutôt d'actions que de perceptions.

scénarios possibles. Pour conserver la liberté d'action de l'utilisateur et assurer l'adaptabilité des scénarios, nous étudions l'hypothèse qu'il est possible de générer dynamiquement et automatiquement des situations à partir de modèles de connaissances qui sous-tendent la simulation. Nous ciblons particulièrement la génération de situations critiques. La gestion de ces situations ne requiert pas seulement des compétences techniques, mais elles mobilisent également des compétences transversales dites non-techniques. Dans la section suivante, nous introduisons ces deux notions qui constituent des éléments phares du projet MACCOY CRITICAL.

1.2 Compétences non-techniques et situations critiques

Dans le cadre du projet, nous traitons donc deux notions essentielles qui sont les **compétences non-techniques** et la **criticité**. Dans cette section, nous présentons tout d'abord brièvement les différentes acceptions de la criticité. Nous reviendrons en détail sur cette notion plus tard dans le mémoire. Ensuite, nous clarifions la notion de compétences non-techniques, et présentons en détail une des taxonomies les plus citées de la littérature.

1.2.1 La criticité

Dans la littérature, il n'existe pas vraiment de consensus sur la définition de la criticité. Elle est souvent confondue avec les notions de risque ou de crise. Dans certains cas, on se réfère à la criticité par une valeur numérique calculée à partir de plusieurs paramètres (e.g. gravité des événements et fréquence d'occurrence). Dans d'autres cas, elle émerge des relations conflictuelles et incohérentes dans les bases de connaissances. C'est le cas par exemple en ingénierie de la dissonance, où la criticité est inférée à partir de différents types de conflits (e.g. conflit d'intention, conflit d'utilisation, conflit d'action). Parfois, comme dans le cadre du projet, elle est caractérisée par différentes dimensions de criticité (e.g. dilemme, ambiguïté, nouveauté). Dans ce type de situations, la maîtrise des compétences techniques s'avère insuffisante. Il est nécessaire en effet de mobiliser des compétences transversales, dites non-techniques. Nous revenons plus en détail sur la notion de situation critique plus loin dans le mémoire, dans la section 2.1 de l'état de l'art.

1.2.2 Compétences non-techniques

Nous avons évoqué précédemment la notion de *compétence* sans pour autant donner une définition claire de ce que cela signifiait. A l'instar de la criticité, cette notion renvoie à

différentes définitions dans la littérature. Nous retenons celle de (Tardif, 2006) qui définit la compétence comme étant :

Compétence

Un savoir-agir complexe qui prend appui sur la mobilisation et la combinaison efficaces d'une variété de ressources internes et externes à l'intérieur d'une famille de situations

En plus de la dimension technico-procédurale de la compétence, un autre aspect a été mis en exergue, d'abord dans le secteur de l'aéronautique, et ensuite dans plusieurs autres domaines. En effet dès les années 1980, suite aux analyses d'accidents et d'incidents majeurs en aviation (Helmreich, 2000), il s'est avéré que les **facteurs humains** et organisationnels jouaient un rôle majeur dans la survenue de ces accidents. De la même façon, en médecine, l'étude d'événements indésirables chez les patients a mis l'accent sur le fait que la plupart de ces événements ne sont pas dus à une défaillance technique (manque de connaissance, non-maîtrise de gestes techniques), mais souvent, à des causes humaines et organisationnelles (Cooper et al., 1982; Newbower et al., 1981). Ces facteurs humains englobent un type particulier de compétences qui sont les compétences non-techniques. Dans la littérature, on s'y réfère souvent avec des termes tels que : *Generic skills*, *Key competences*, *Core skills*, *Soft skills* (Nabil Bin NASIR et al., 2011). Les compétences non-techniques ont fait l'objet de plusieurs travaux de (Flin et al., 2008, 2009). Ce sont d'ailleurs les auteurs de l'une des définitions de CNT les plus citées de la littérature. Ils définissent la CNT comme étant :

Compétence non-technique

une combinaison de savoirs cognitifs, sociaux, et de ressources personnelles complémentaires des savoir-faire procéduraux qui contribuent à une performance efficiente et sûre.

Par ailleurs, les auteurs distinguent sept classes de compétences non-techniques :

1. **Conscience de la situation**. Capacité des agents à percevoir les éléments constituant une situation dans un espace-temps donné, de leur donner du sens et à se projeter.
2. **Prise de décision**. Capacité des agents à formuler un problème, à identifier les alternatives, à en sélectionner une et à l'implémenter.
3. **Communication**. Capacité des agents à échanger de l'information, des commentaires, des réponses, des idées ou des pensées en formulant et envoyant une information claire et explicite, en la recevant et en l'interprétant telle qu'attendue par l'émetteur.
4. **Travail d'équipe**. Capacité à soutenir les autres membres de l'équipe, à régler des conflits, à échanger de l'information et à coordonner les différentes activités.

5. **Leadership**. Capacité à exercer une autorité, à maintenir des standards à planifier et prioriser, à gérer la fatigue et les ressources.
6. **Gestion du stress**. Capacité à identifier les symptômes de stress, reconnaître les effets et mettre en œuvre des stratégies de prévention et de réponses adaptées.
7. **Gestion de la fatigue** : Capacité à identifier les symptômes et les effets de la fatigue et à les prévenir et les gérer efficacement.

En complément de ces compétences, dans certains domaines, notamment en médecine, certains auteurs définissent la notion de **compétence éthique** (Bertholet, 2000). Elle se réfère à la capacité "d'intégrer et combiner des savoirs multiples pour discerner les questions éthiques afin de les finaliser dans l'action morale". Cette compétence permet de "résoudre des problèmes moraux nouveaux et imprévus".

1.3 Les environnements virtuels pour l'apprentissage

Afin d'entraîner les individus à ce type de compétences, nous pensons qu'il est intéressant de les confronter à une variabilité de situations critiques. Cependant, la production de situations critiques en conditions réelles n'est pas toujours possible. Les environnements virtuels peuvent pallier ce problème en permettant la création de scènes virtuelles 3D interactives, proches de celles rencontrées en réel. Par ailleurs, afin d'intégrer l'aspect apprentissage, ces environnements virtuels peuvent être couplés avec des outils qui permettent la scénarisation automatique et/ou manuelle des sessions d'apprentissage.

1.3.1 Simulation et environnements virtuels

Pour (Bobillier et al., 1976), la simulation consiste à construire et exécuter un modèle d'un système réel afin d'étudier le comportement de ce système, sans pour autant perturber l'environnement du système réel. Cette définition rejoint plusieurs autres définitions générales dans la littérature. Elle bénéficie de plus de précision lorsqu'on s'intéresse à un domaine particulier. En médecine par exemple, (Chiniara, 2007) distingue dans sa classification deux types de simulations : *la simulation organique* (e.g. utilisation d'humains ou d'animaux) et la *simulation non-organique*. Cette dernière est subdivisée en deux autres classes : la simulation synthétique, e.g. l'usage de mannequins haute fidélité, et la simulation électronique, e.g. l'usage des interfaces graphiques (cf. figure 1.1). Dans la suite de ce mémoire, nous faisons référence à la simulation non-organique électronique lorsque nous utilisons le mot "*simulation*". Au vu de cette classification, les environnements virtuels sont donc un type spécifique de simulation. Aujourd'hui, les travaux de recherche menés dans les EV ont

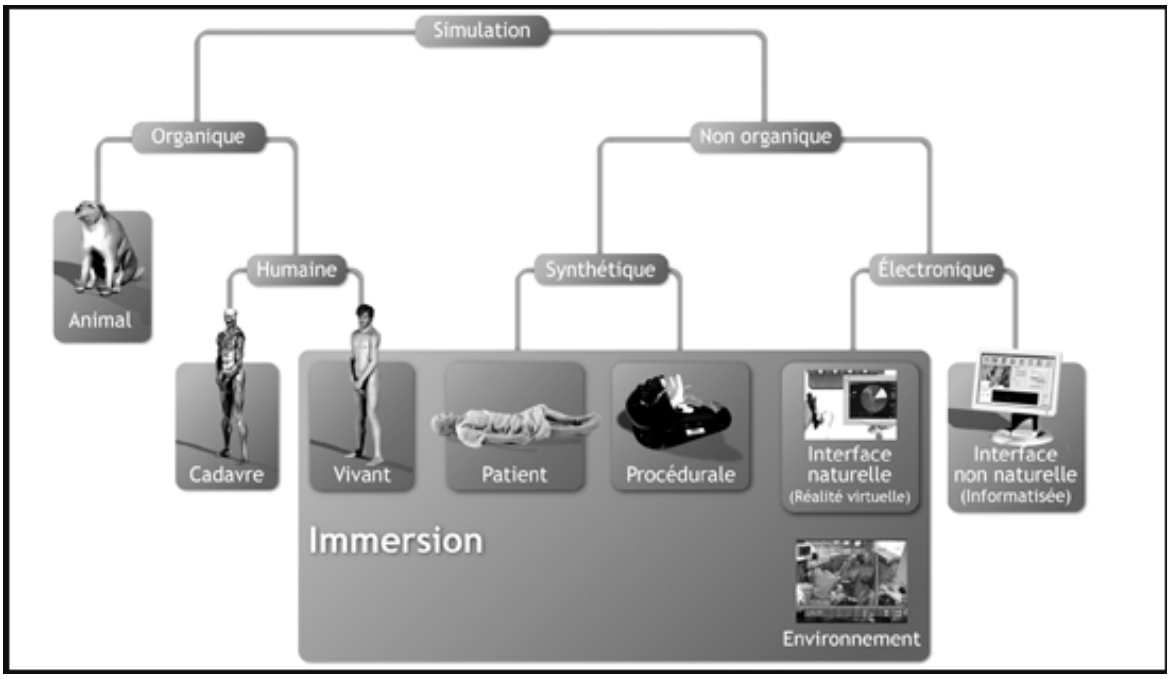

FIGURE 1.1 Classification des simulations proposées par (Chiniara, 2007)

permis leur utilisation efficace pour l'apprentissage. En effet, le potentiel des EV est exploité pour former et/ou entraîner les individus dans plusieurs secteurs, notamment en aéronautique, dans le domaine militaire et en santé (Baker et al., 1993; Huguet et al., 2016; Reznek et al., 2002). Ils sont particulièrement utilisés pour leurs nombreux avantages (Burkhardt et al., 2006) : (1) **mettre en situation** quand les conditions réelles ne le permettent pas (dangerosité, accessibilité, coût), (2) **simuler des scénarios rares** (e.g. incidents techniques, accidents), (3) **contrôler** les variables de la simulation (e.g. conditions climatiques, comportements des autres agents), (4) et **répéter** le même scénario à des fins d'observation, d'entraînement ou d'évaluation. Par ailleurs, l'utilisation de la simulation a montré que les types de pensées et d'émotions qui sont évoqués dans les situations réelles sont susceptibles de survenir durant les expériences en environnements virtuels (Rovira et al., 2009), prouvant la validité écologique de ces environnements. Des études ont également montré que les réponses subjectives, comportementales et physiologiques des personnes en EV correspondent à leur comportement et expérience en environnement réel. Ce fut le cas par exemple lors de la réplique virtuelle de l'expérience de (Milgram, 1963) conduite par (Slater et al., 2006).

1.3.2 La scénarisation en environnements virtuels

Un des éléments cruciaux d'une formation est le *scénario*, car il correspond à la fondation sur laquelle l'apprentissage aura lieu (Salas and Burke, 2002). Ce dernier est conçu et

orchestré par les instructeurs et/ou les experts du domaine. En environnements virtuels, il arrive parfois que l'orchestration et la spécification du scénario soient réalisées par le système de simulation lui-même. Nous parlons alors de *scénarisation informatique*. Dans cette sous-section, nous apportons quelques clarifications sur la notion de scénarisation et les termes qui s'y rapportent.

Tout d'abord, il convient de préciser ce que nous entendons par *situation*. S'agissant d'environnements virtuels, nous retiendrons la définition de (Carpentier, 2015) qui rend compte des considérations informatiques :

Situation

Une situation est une description partielle du monde à un instant donné sous la forme d'un agrégat d'assertions

C'est donc une photographie partielle du monde qui ne contient pas d'informations sur la manière dont elle a été créée (e.g. "la chaise est cassée", "la porte est fermée").

Pour atteindre une situation particulière, il est nécessaire de dérouler un *scénario*. Dans le contexte cinématographique ou théâtral, ce terme renvoie à deux choses : d'une part, aux scripts, dialogues et enchaînements des actions etc. (*what to shoot*), et d'autre part, à la manière de mettre en scène ces éléments (*how to shoot*) (Van Rijsselbergen et al., 2009). Dans un contexte interactif, où les sujets jouissent d'une liberté d'action, il se peut qu'on dévie du scénario initial et que des événements non prévus par les auteurs se produisent. Nous distinguons alors le *scénario prescrit*, tel qu'il a été défini en amont, du *scénario effectif*, tel qu'il a été déroulé (Carpentier, 2015; Pernin et al., 2004). Prenant en compte ces considérations, Carpentier définit un scénario comme :

Scénario

Un ensemble d'événements particuliers, partiellement ordonnés et instanciables dans un envrionnement virtuel

La modélisation d'un scénario consiste en deux phases : d'une part, la définition du contenu scénaristique (e.g. objets, actions possibles, propriétés des agents), et d'autre part, la spécification des objectifs scénaristiques (e.g. sélection d'un sous-ensemble d'événements particuliers). Dans un contexte de scénarisation informatique, la modélisation du scénario est considérée comme une étape initiale seulement, car il est également nécessaire d'opérationnaliser ce scénario afin qu'il soit exploitable et contrôlable dans un environnement virtuel. Nous considérons, comme le propose (Barot, 2014), qu'un système de scénarisation est *un système 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.

1.4 Problématique de recherche et objectifs

Dans le courant constructiviste, l'apprentissage passe par la confrontation à des situations pertinentes diverses et variées. Dans le cadre de nos travaux, nous nous intéressons, principalement, au développement des compétences non-techniques. Par conséquent, pour permettre un apprentissage constructiviste de ces compétences, nous devons confronter l'apprenant à des situations critiques favorisant le développement de ces compétences. L'enjeu informatique consiste à générer dynamiquement ces situations tout en les adaptant à chaque apprenant. Nous distinguons deux niveaux d'adaptation. Le premier concerne l'ajustement des situations en fonction de chaque apprenant pour quelles soient considérées comme critiques. Le deuxième concerne l'adaptation des situations en fonction de niveau de développement courant de l'apprenant pour favoriser l'apprentissage. Dans le cadre de nos travaux, nous nous sommes plutôt intéressés au premier niveau d'adaptation, car il est intrinsèque à la problématique informatique de génération automatique de situations critiques. A terme, l'objectif serait d'intégrer les deux niveaux d'adaptation afin de proposer des situations qui soit à la fois critiques pour l'apprenant, et adaptées pour son apprentissage.

Notre contribution majeure porte sur deux types de situations critiques en particulier que sont les dilemmes et les situations ambiguës. Notre ambition est de produire un moteur de génération, ancré dans un système de scénarisation informatique, capable de générer de telles situations sans avoir à les décrire au préalable. Dans cette section, nous présentons l'ensemble des objectifs auxquels doit répondre notre moteur ainsi que les principaux verrous qui y sont liés.

Pour faciliter le transfert des acquis, une **variabilité** des situations dans lesquelles la tâche est à accomplir est indispensable (Tardif and Meirieu, 1996). De ce fait, un système de scénarisation doit pouvoir proposer un large panel de situations pour de ne pas se restreindre à la simple reproduction non-contextualisée de tâches. Il doit être capable de varier les situations proposées, allant des plus courantes jusqu'aux plus dégradées et improbables.

Face à ces situations, l'apprenant doit jouir d'une **liberté d'action** afin d'expérimenter les conséquences de ses choix. Il ne suffit pas de créer un sentiment de liberté (Thue et al., 2010), il faut produire un impact réel des actions de l'apprenant. Par conséquent, le système de scénarisation doit être capable de générer les conséquences de ces actions à court et à long terme, de manière à ce que l'apprenant puisse les associer à ses actions.

Pour que l'apprenant puisse bénéficier de la meilleure expérience possible, le système doit pouvoir **contrôler** la simulation. Il doit être en mesure de sélectionner les situations d'apprentissage les plus pertinentes et de déterminer le meilleur séquencement possible de celles-ci. Par ailleurs, dans un cadre d'entraînement à la gestion de situations critiques, le système doit pouvoir ajuster le degré de criticité en fonction d'un ensemble d'objectifs pédagogiques.

Un autre point important dans la conception d'un système de scénarisation est sa **généricité**. Plus un système est générique, moins il faudra fournir d'efforts pour le réutiliser dans d'autres domaines.

L'un des premiers verrous associés à ces objectifs naît de la difficulté de les concilier : pour exercer un contrôle sur la simulation tout en offrant une liberté d'action à l'utilisateur, dans une approche classique, il faudrait anticiper à l'avance toutes les actions de l'utilisateur et prévoir pour chacune d'elles les ajustements scénaristiques adéquats. Dans les domaines complexes, cette approche est vouée à l'échec. Le grand nombre d'entités et les relations complexes qui les relient font que l'écriture en amont des scénarios provoque une explosion combinatoire (Spierling and Szilas, 2009). Ceci donne lieu a une première question de recherche : **comment générer une variété de scénarios pertinents sans produire une explosion combinatoire** ? Toutefois, le verrou majeur de nos travaux est lié principalement à la génération dynamique des situations dilemmatiques et ambiguës. Ces deux dimensions ont la particularité d'être non chiffrées. Par conséquent, elles sont difficilement quantifiables et calculables. La génération de ces dimensions est d'autant plus complexe lorsque les situations ne sont pas décrites en amont et lorsque le système de génération doit également répondre aux objectifs cités précédemment. Nous soulevons alors les questions de recherche suivantes : **Comment modéliser ces dimensions ? Comment les qualifier ? Et comment les générer sans avoir à les décrire en amont ?**

1.5 Approche

Afin de préserver la liberté d'action de l'utilisateur, tout en exerçant un contrôle sur la simulation, nous proposons une approche émergente qui fonde le processus de scénarisation sur des modèles de connaissances indépendants du contexte. Cette approche permet de faire émerger des situations à partir de représentations sémantiques expressives, en raisonnant à différents niveaux d'abstractions, permettant ainsi, la génération dynamique d'un large panel de situations. Cette variabilité des situations est contrôlée par un ensemble d'objectifs pédagogiques issus du diagnostic de l'apprenant. Nous proposons un moteur de génération qui prend en compte ces objectifs et sélectionne les situations qui y répondent au mieux.

Nos travaux portent particulièrement sur la génération dynamique des dilemmes et des situations ambiguës. Nous proposons tout d'abord un modèle formel de ces deux notions en nous basant sur des théories issues des sciences humaines et sociales. A partir de cette formalisation, nous proposons un ensemble d'algorithmes et de techniques de génération qui raisonnent sur des modèles de connaissances — manipulés par les experts du domaine — qui ne sont pas destinés, a priori, à la description des dilemmes ou des situations ambiguës. Le raisonnement sur ces modèles sémantiques permet d'inférer de nouvelles connaissances qui permettent d'identifier les entités (e.g. actions, événements, objets) susceptibles de créer, lorsqu'elles sont combinées ensemble, des situations qui vérifient les propriétés énoncées dans la formalisation des dilemmes et de l'ambiguïté.

Afin de proposer un contenu adapté à chaque apprenant, pour la génération des dilemmes moraux, nous proposons d'opérationnaliser la théorie des valeurs universelles de (Schwartz, 2006), afin de prendre en compte le système de valeurs propre à chaque individu. Par ailleurs, pour la génération de l'ambiguïté, il est nécessaire de prendre en compte le degré de connaissance de l'apprenant des variables du monde. Nous proposons alors une modélisation de la représentation mentale de l'apprenant. Afin de considérer les incertitudes et les conflits dans cette représentation, nous proposons d'utiliser le cadre formel des fonctions de croyances qui est adapté à ce besoin.

1.6 Contexte technologique et applicatif

Ces travaux de thèse s'inscrivent dans le cadre général du projet ANR MACCOY CRITICAL dont l'objectif est l'étude et l'amélioration des dispositifs de formation utilisant la simulation et les environnements virtuels pour l'entraînement aux compétences non-techniques. Notre contribution porte sur la génération dynamique de situations critiques grâce à un système de scénarisation. Ce dernier est ancré dans l'architecture de la plateforme HUMANS que nous présentons par la suite.

1.6.1 Architecture du projet

Le projet est appliqué à deux domaines aux forts enjeux en termes de santé publique et de sécurité qui sont la conduite automobile et la médecine. Pour cela, il a été prévu un simulateur pour chaque domaine, permettant d'évaluer ainsi la généricité des systèmes développés (Barre et al., 2017; Burkhardt et al., 2016). A partir de ces simulateurs, sont collectés des indicateurs qui vont permettre de décider du besoin de produire une rétroaction (boucle courte) et/ou d'établir un diagnostic de l'apprenant (boucle longue). Ce diagnostic

émet une décision didactique et pédagogique qui est transmise au système de scénarisation sous la forme d'une consigne pédagogique. Cette dernière est composée de trois éléments essentiels qui sont :

1. une intention pédagogique (e.g. vérifier, renforcer ou déstabiliser une compétence),
2. une ou des compétence(s) (technique et/ou non-technique) cible(s),
3. et un degré de criticité.

Un exemple de consigne correspond alors à : **renforcer** la compétence non-technique **conscience de la situation** avec un degré de criticité **élevé**. Notre système de scénarisation, représenté dans l'architecture par les blocs *Scenario Orchestration* et *Scenario Generation*, se charge ensuite de traduire cette consigne en objectifs scénaristiques, puis en un état du monde concret qui est envoyé au simulateur afin de mettre à jour le rendu visuel. La figure 1.2 présente un aperçu haut-niveau de cette architecture.

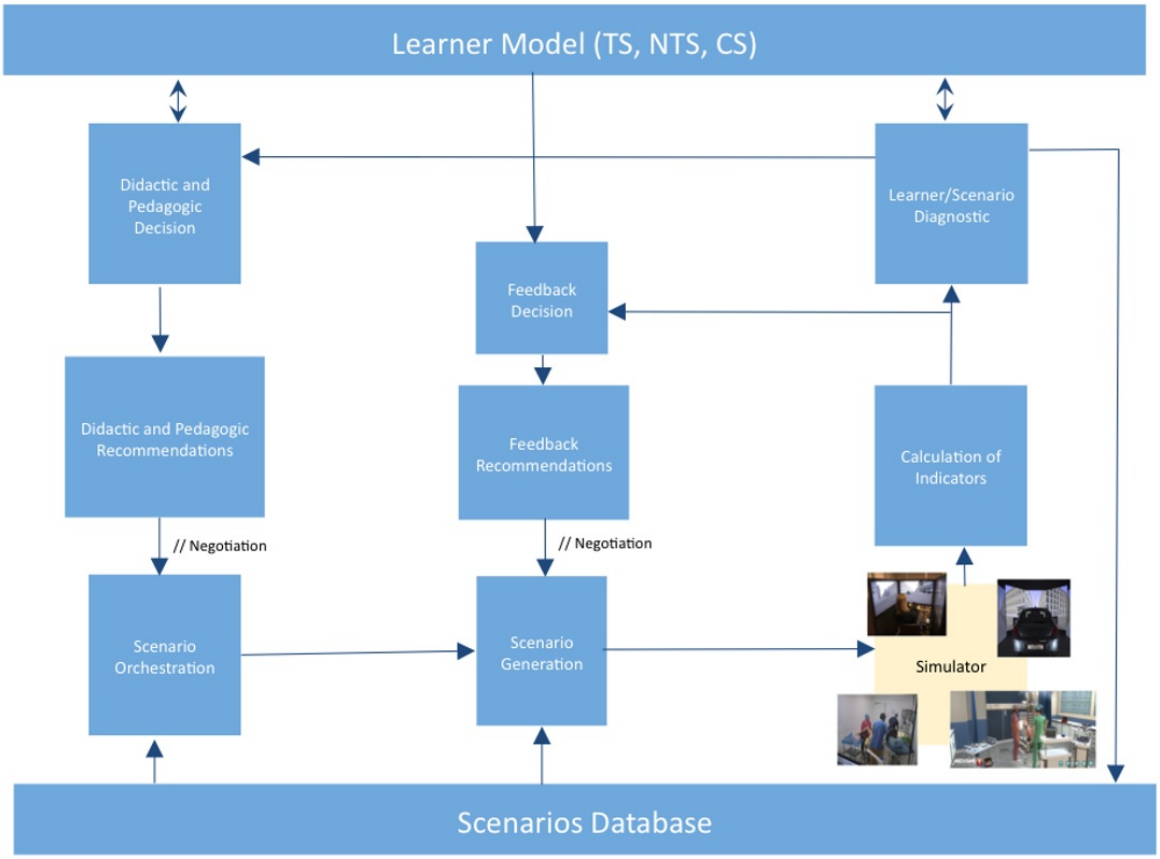

FIGURE 1.2 Architecture du projet

1.6.2 Plateforme Humans

Le système de scénarisation fait partie de la plateforme logicielle HUMANS (HUman Models based Artificial eNvironments Software platform). Celle-ci repose sur une architecture conceptuelle modulaire, qui permet à chaque module de fonctionner indépendamment des autres. La plateforme est constituée d'un ensemble de modules décisionnels raisonnant sur des méta-modèles communs (cf. Figure 1.3). Parmi ces modules, nous retrouvons TAILOR qui est en charge de générer des objectifs scénaristiques en fonction d'un profil incertain de l'apprenant. Ce dernier est construit à partir des traces enregistrées par le module MONITOR qui est responsable du suivi de l'apprenant. Ces objectifs scénaristiques sont envoyés à DIRECTOR qui se charge de diriger le scénario vers des situations qui respectent ces objectifs. Pour cela, il produit des ajustements scénaristiques qui sont envoyés au module qui gère la mise à jour de l'état du monde : le WORLD MANAGER. Finalement, la plateforme comporte le module REPLICANTS qui est en charge de gérer le comportement des personnages virtuels autonomes. Nos contributions dans le cadre de cette thèse, sont opérationnalisées au sein d'un moteur de génération ancré dans le module TAILOR. Il participe à la génération des objectifs scénaristiques par l'intégration de la criticité.

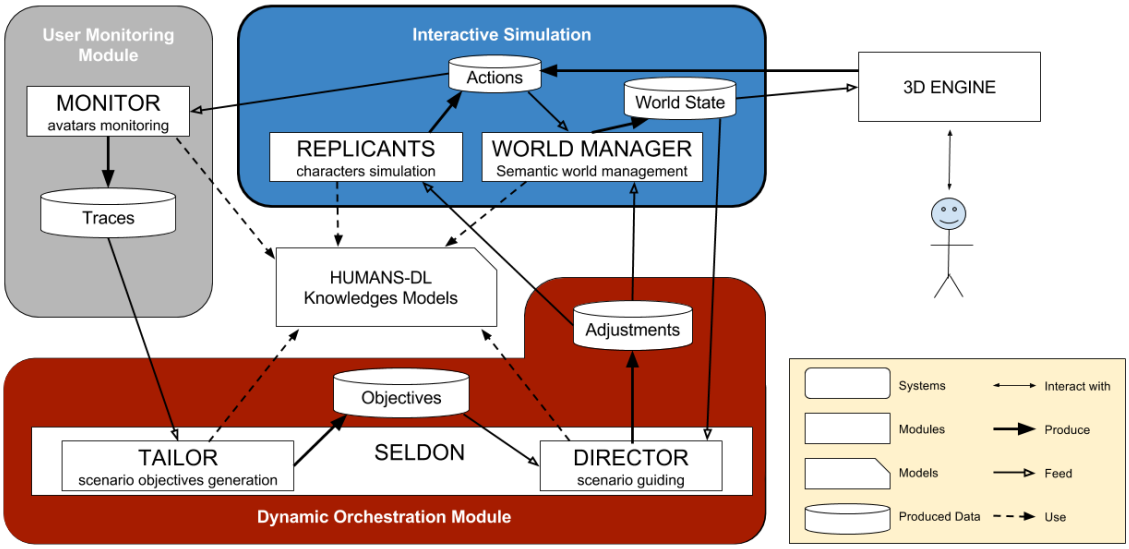

FIGURE 1.3 Architecture de la plateforme HUMANS

1.7 Organisation du mémoire

Le chapitre 2 de ce mémoire est consacré à l'état de l'art sur les notions de criticité, dilemme et ambiguïté. Nous exposons tout d'abord les différentes modélisations de la criticité

dans la littérature et nous présentons la représentation multidimensionnelle de cette notion que nous avons adoptée dans le cadre du projet. Ensuite, nous étudions la notion de dilemme, et détaillons les différentes représentations de cette notion issues des sciences humaines et sociales. Après cela, nous présentons les différents systèmes informatiques de l'état de l'art qui mettent en jeu cette notion. De manière analogue, pour l'ambiguïté, nous détaillons tout d'abord les différentes acceptions théoriques de cette notion. Ensuite, nous présentons les différents systèmes informatiques qui la mettent en jeu.

Dans le chapitre 3, nous décrivons notre contribution. Nous présentons en premier lieu l'architecture globale dans laquelle s'intègre le moteur de génération que nous proposons. Ensuite, nous détaillons le processus de génération automatique des dilemmes que nous avons opérationnalisé au sein d'un moteur que nous avons nommé KOBA. Nous clarifions tout d'abord la notion de conséquence négative et expliquons comment nous la considérons dans le cadre de nos travaux. Ensuite, nous exposons notre modélisation du profil moral de l'apprenant pour la génération des dilemmes moraux. Nous détaillons notre formalisation d'une situation dilémmatique ainsi que les différentes étapes de sa génération. Nous nous intéressons ensuite à la deuxième notion importante de notre contribution : l'ambiguïté. Nous présentons tout d'abord notre modélisation de la représentation mentale incertaine de l'apprenant, puis les différents algorithmes et techniques pour la génération des différents types d'ambiguïté.

Dans le chapitre 4, nous présentons l'implémentation de nos travaux. Ensuite, nous détaillons, dans le chapitre 5, les résultats issus des expériences que nous avons menées dans le cadre de cette thèse. Finalement, dans le chapitre 6, nous dressons un résumé de nos travaux en mettant en exergue les limites qui y sont liées, tout en ouvrant sur des perspectives d'amélioration.

Chapitre 2

Etat de l'art

Sommaire

2.1 La criticité 26
2.1.1 Critique au sens crucial 27
2.1.2 La criticité chiffrée 27
2.1.3 La criticité non chiffrée 29
2.1.4 Criticité : approche multidimensionnelle 31
2.1.5 Bilan 32
2.2 Les dilemmes en théorie 33
2.2.1 Définition générale 33
2.2.2 Les dilemmes moraux 34
2.2.3 Les valeurs morales 35
2.2.4 Les dilemmes, instruments de recherche et de formation 36
2.2.5 Bilan 38
2.3 Dilemme dans les simulations 38
2.3.1 Les approches scriptées 39
2.3.2 Les approches génératives 40
2.3.3 Bilan et positionnement 43
2.4 L'ambiguïté en théorie 44
2.4.1 Ambiguïté et probabilités 44
2.4.2 Ambiguïté et incertitudes 46
2.4.3 Ambiguïté, contexte et relations 48
2.4.4 Bilan 49
2.5 L'ambiguïté dans les simulations 50
2.6 Bilan 53

L'état de l'art traite principalement trois sujets majeurs : la criticité, les dilemmes et l'ambiguïté. Nous abordons tout d'abord dans la section 2.1 la notion de criticité en détaillant ses différentes acceptions dans la littérature. Nous faisons une distinction entre la criticité chiffrée, qui est calculée numériquement à partir de plusieurs paramètres, et la criticité non chiffrée qui émerge des conflits et des incohérences. Nous concluons en proposant une représentation multidimensionnelle de la criticité telle que nous l'adoptons dans le cadre du projet, et en proposant les définitions que nous adoptons pour les situations et les scénarios critiques en environnement virtuel.

Dans la section 2.2, nous réalisons une revue de la littérature sur les dilemmes. Nous présentons tout d'abord quelques définitions générales, en mettant en exergue les distinctions faites par certaines auteurs entre les types de dilemmes. Nous détaillons particulièrement les dilemmes moraux, et présentons un ensemble de travaux sur les valeurs morales qui nous ont guidés dans notre modélisation du profil moral de l'apprenant. Finalement, nous présentons quelques exemples où les dilemmes ont été utilisés à des fins de recherche et de formation. Dans la section 2.3, nous listons les différents travaux en informatique qui se sont intéressés à l'intégration des dilemmes dans les simulations. Nous distinguons les approches scriptées et les approches génératives. Nous concluons cette section en dressant un bilan de ces approches, tout en mettant en évidence les limites des systèmes informatiques recensés, et en prenant soin de clarifier notre positionnement par rapport à ces systèmes.

Dans la section 2.4, nous détaillons les différentes acceptions de l'ambiguïté dans la littérature. Nous distinguons notamment l'ambiguïté de l'information, de l'ambiguïté du contexte et de l'ambiguïté normative. Ensuite, dans la section 2.5, nous recensons l'ensemble des travaux qui se sont intéressés à l'intégration de l'ambiguïté dans les simulations. Enfin nous dressons un bilan de ces travaux et clarifions notre positionnement.

2.1 La criticité

Dans la littérature, il n'existe pas vraiment de consensus sur la définition de la criticité. Elle est souvent confondue avec les notions de crise ou de risque. Dans cette section nous faisons une revue de la littérature autour de la criticité. Tout d'abord, nous abordons un premier sens de cette notion qui se rapporte au caractère crucial d'un événement dans une activité. Ensuite, nous distinguons la criticité chiffrée qui est calculée à partir de différents paramètres, de la criticité non-chiffrée qui émerge plutôt des relations conflictuelles dans les bases de connaissances. Finalement, nous présentons l'approche multidimensionnelle adoptée dans le cadre du projet qui concilie les différents éléments de l'état de l'art.

2.1.1 Critique au sens crucial

Une situation peut être critique au sens de pertinente et représentative pour l'activité étudiée. Cette définition de la criticité trouve ces racines dans la Méthode des Incidents Critiques (MIC) décrite par (Flanagan and C., 1954). La MIC est une technique qualitative d'interview qui consiste à recueillir des incidents pertinents sur des comportements en lien avec la performance positive ou négative dans des situations clairement définies. Cette méthode fut appliquée par son auteur pendant la seconde guerre mondiale dans le cadre d'un programme de recherche (« Aviation Psychology Program ») visant à optimiser les pratiques professionnelles des pilotes de l'armée de l'air américaine. Elle a été reprise plus tard dans une variété de contexte, notamment en marketing (Gremler, 2004), en santé mentale (Pourin et al., 2001) et en agriculture (Irwin and Poots, 2015). Tel que défini par l'auteur, l'incident désigne toute activité humaine observable qui est suffisamment complète en soi pour permettre des inférences et des prédictions sur la personne réalisant l'activité. C'est donc un événement qui n'est pas forcement négatif en soi, bien qu'en pratique, la plupart des incidents reportés le soient. Ainsi, la criticité ici ne se réfère pas au caractère grave ou dangereux d'une situation, mais plutôt à son caractère *crucial* mettant en évidence les caractéristiques du métier.

2.1.2 La criticité chiffrée

Dans un autre contexte, la notion de criticité est souvent assimilée à la dangerosité d'une situation. Les normes EN 292-1, EN 292-2 et EN 1050 définissent notamment des concepts qui s'y rapportent, tels que les situations dangereuses, le danger, le dommage etc. La *situation dangereuse* est définie comme étant une situation dans laquelle une personne est exposée à un ou plusieurs dangers. Un *danger* est considéré comme toute cause capable de provoquer une lésion ou une atteinte à la santé (i.e. un *dommage*). Lorsqu'une personne se trouve dans une situation dangereuse, elle est exposée à un événement susceptible de causer un dommage (i.e événement dangereux ou déclencheur). Selon les cas, un dommage pourra être plus ou moins sévère. On parlera alors de gravité. Par ailleurs, une situation peut être critique parce qu'elle est imprévisible. Dans ce cadre, (Westrum, 2006) distingue trois classes de situations :

— **les menaces courantes**, qui se réfèrent aux événements qui se produisent souvent et auxquels on peut apporter une réponse ;
— **les menaces irrégulières**, qui font référence aux événements pour lesquels il n'existe pas en pratique une réponse standard ;
— **les événements uniques**, qui désignent des situations sans précédents et impossibles à imaginer.

Dans certains contextes, on parlera plutôt de fréquence d'occurrence. Celle-ci correspond à l'incertitude aléatoire due à la variabilité naturelle des événements. En analyse de risque, elle est traditionnellement associée à la gravité pour le calcul de la criticité. Cette dernière est calculée à partir du produit de ces deux paramètres :

Criticité = Fréquence d'occurrence * Gravité des conséquences

		Niveau de Gravité			
		Insignifiant	Marginal	Critique	Catastrophique
Fréquence	Fréquent	Indésirable	Inacceptable	Inacceptable	Inacceptable
	Probable	Acceptable	Indésirable	Inacceptable	Inacceptable
	Occasionnel	Acceptable	Indésirable	Indésirable	Inacceptable
	Rare	Négligeable	Acceptable	Indésirable	Indésirable
	Improbable	Négligeable	Négligeable	Acceptable	Indésirable
	Invraisemblable	Négligeable	Négligeable	Négligeable	Acceptable

FIGURE 2.1 Matrice d'évaluation du risque

Cette description bidimensionnelle de la criticité est souvent représentée grâce à une matrice (cf. figure 2.1). Dans cette représentation, la gravité et la fréquence d'occurrence sont divisées en plusieurs classes. Chaque cellule de la matrice, correspond à un niveau de risque (ou criticité) qui peut aller du négligeable à l'inacceptable. Analogiquement en médecine, on parlera d'*acuité* (Acuity) : la gravité potentielle d'un événement et son impact subséquent sur le patient, et d'*opportunité* (Opportunity) : la fréquence avec laquelle les individus sont impliqués dans la gestion de l'événement, ou alternativement, la probabilité de découverte d'un problème particulier. Ces deux paramètres définissent une matrice qui peut être divisée en quatre quadrants : high-acuity low-opportunity (HALO), high-acuity high-opportunity (HAHO), low-acuity low-opportunity (LALO), and low-acuity high-opportunity (LAHO). Selon cette catégorisation, la gestion d'une hémorragie postpartum est considérée comme une situation de type HAHO. D'une part, c'est un cas récurrent pour le personnel médical (HO), et d'autre part, les retombées pourraient être fatales pour le patient s'il n'est pas correctement pris en charge (HA). Cette matrice, présentée par (Chiniara et al., 2013) , permet également d'identifier la *zone de simulation* (cf. figure 2.2). Celle-ci correspond à la zone où l'utilisation de la simulation s'avère avantageuse par rapport aux autres supports pédagogiques.

Dans d'autres domaines, notamment en sûreté de fonctionnement, et particulièrement en Analyse des Modes de Défaillance, de leurs Effets et de leur Criticité (AMDEC), le calcul de la criticité intègre des paramètres supplémentaires comme la *détéctabilité* de l'événement.

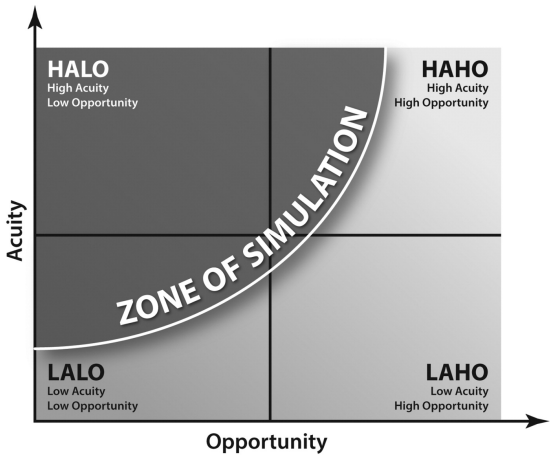

FIGURE 2.2 Zone de simulation de la matrice Acuity-Opportunity

C'est aussi le cas de la norme ISO 13849-1 :2008 qui définit la criticité en fonction des paramètres suivants :

- S : gravité de la blessure
  - S1 : blessure légère (en général réversible),
  - S2 : blessure grave (en général irréversible) ;
- F : fréquence et/ou durée d'exposition au phénomène dangereux
  - F1 : rare à assez fréquent et/ou courte durée d'exposition,
  - F2 : fréquent à continu et/ou longue durée d'exposition.
- P : possibilité d'éviter le phénomène dangereux ou de limiter le dommage
  - P1 : possible sous certaines conditions,
  - P2 : rarement possible.

A partir de ces éléments, la norme définit une évaluation arborescente (cf. figure 2.3) qui permet de déterminer cinq "niveaux de performances" qui peuvent s'apparenter à des niveaux de criticité.

2.1.3 La criticité non chiffrée

Dans des travaux récents en analyse de risque, la criticité n'est pas considérée comme une valeur chiffrée. Elle est plutôt apparentée à des concepts issus de la psychologie sociale. C'est notamment le cas de plusieurs travaux en ingénierie de la dissonance (Vanderhaegen, 2016). Festinger (Festinger, 1957) décrit la dissonance cognitive comme étant un état de tension inconfortable ressenti par une personne en présence de cognitions (connaissances, opinions ou croyances) incompatibles entre elles. Par exemple, "Vitesse limitée à 50km/h" et "Rouler à 100km/h" sont des cognitions incompatibles. Lorsqu'elles sont simultanément mises en

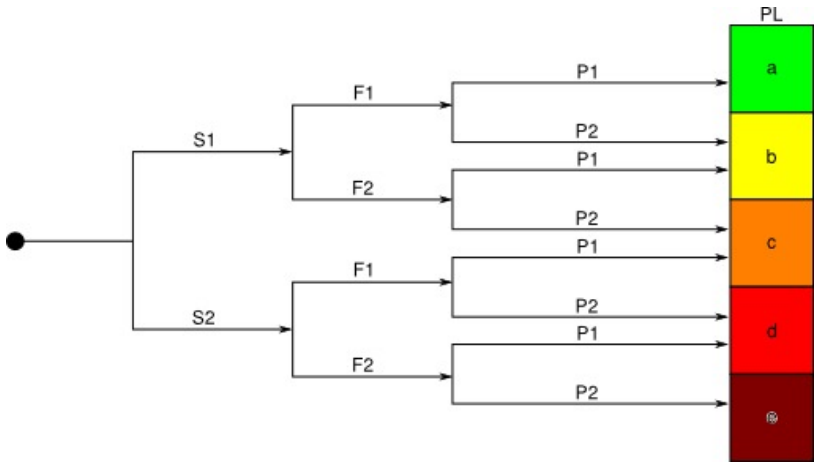

FIGURE 2.3 Niveaux de performance (norme ISO 13849-1 :2008)

jeu, elles produisent un état psychologique inconfortable. Cet état motive l'individu à réduire cet inconfort par différents mécanismes : le changement d'attitude (réduire la vitesse) ou l'ajout de nouvelles cognitions (je suis très en retard). Dans les systèmes techniques, une dissonance survient lorsqu'il existe des lacunes, des incohérences ou des conflits dans une même base de connaissance ou entre plusieurs bases distinctes. Les travaux de la littérature traitent particulièrement les dissonances qui résultent des conflits d'intention, des conflits d'utilisation, des conflit d'alternatives, du manque de connaissance, des conflits entre points de vue, des conflits entre les actions, etc. Dans les travaux de Vanderhaegen (Vanderhaegen, 2016) par exemple, l'auteur propose un système d'aide à la conduite qui exploite les différents types de raisonnements (abductifs, inductifs et déductifs) sur un système à base de règles afin de détecter à la fois les conflits d'intention, les conflits d'action et les conflits d'utilisation. La détection de cette dissonance permet au système d'activer les mécanismes de réduction adéquats. Soit en poussant le conducteur à changer son comportement, en activant des indicateurs audiovisuels (motivateurs passifs) ou en prenant carrément le contrôle du véhicule (motivateur actif), soit en ajoutant de nouvelles cognitions pour expliquer l'attitude du conducteur.

Dans le cadre du projet, nous avons essayé de concilier les différentes descriptions de la criticité présentées jusque-là. Pour cela, nous avons proposé une représentation multidimensionnelle. Dans la section suivante, nous détaillons cette représentation, et présentons brièvement chaque dimension.

2.1.4 Criticité : approche multidimensionnelle

Dans la section précédente, nous avons mis en évidence le fait que la criticité renvoyait à plusieurs acceptions. Nous avons également montré que dans certains cas, elle était considérée comme une valeur numérique, alors que dans d'autres cas, elle émergeait des relations conflictuelles entre les connaissances. Dans le cadre de nos travaux, nous souhaitons former les individus à l'utilisation des compétences non-techniques en situations critiques. Nous avons donc caractérisé la criticité, en collaboration avec l'ensemble des partenaires du projet MacCoy Critical, par un ensemble de dimensions de criticité qui sont à la fois critiques au sens cruciales pour la mobilisation des compétences non-techniques, et critiques au sens qu'elles peuvent engendrer, de manière générale, des conséquences négatives (Burkhardt et al., 2016). Prenant en compte ces considérations, nous avons identifié au total sept *dimensions de criticité* : la gravité, l'imprévisibilité, le dilemme, l'ambiguïté, la charge sociocognitive, la nouveauté et la non-maîtrise. Elles peuvent avoir une origine interne ou externe par rapport au sujet. Nous détaillons ci-après l'ensemble de ces dimensions :

1. **Gravité.** Se réfère au degré de dommages (lésion ou atteinte à la santé) intrinsèque à la situation ou lié à ses conséquences. (e.g "Patient souffrant d'une hémorragie interne" ou "Voiture roulant à toute vitesse et piéton qui traverse soudainement sans faire attention").
2. **Imprévisibilité.** Cette dimension se réfère à l'incertitude aléatoire. Elle correspond à la variabilité naturelle des événements (e.g. tremblement de terre, pluie) désignée aussi par *fréquence d'occurrence*.
3. **Ambiguité.** Cette dimension désigne toute situation qui est sujette à plusieurs interprétations (e.g. consignes vagues et/ou floues). Elle regroupe potentiellement plusieurs types de dissonance à savoir le conflit d'utilisation, le conflit d'intentions, le conflit d'action, le conflit entre points de vue et le conflit d'information.
4. **Dilemme.** Cette dimension caractérise les situations où il n'existe pas de bonne solution. C'est une situation où l'individu est confronté à un choix difficile, sacrifiant, mettant en conflit dans certains cas, ses valeurs morales. Cette dimension peut s'apparenter aux conflits des alternatives en ingénierie des dissonances. (e.g. percuter le piéton ou se sacrifier ainsi que les passagers du véhicule, en cas de situation de collision inévitable).
5. **Charge sociocognitive.** Se réfère aux situations où la charge de travail est affectée par des stresseurs internes (e.g. fatigue, peur) et/ou externes (e.g. klaxons des autres automobilistes, contraintes temporelles, multiplicité des tâches).
6. **Nouveauté.** Situation à laquelle l'apprenant n'a jamais été confronté.

7. **Non-maitrise.** Situation qui mobilise des compétences non maîtrisées par l'apprenant.

2.1.5 Bilan

Dans cette section, nous avons réalisé une revue de la littérature sur la notion de criticité. Nous avons montré qu'elle renvoyait à plusieurs acceptions, et que dans le cadre du projet, nous avons opté pour une représentation multidimensionnelle. En rendant compte de ces considérations, et dans le cadre de la scénarisation informatique, nous définissons alors une *situation critique en environnement virtuel* comme suit :

Situation critique en environnement virtuel

Une situation critique est une description partielle du monde à un instant donné sous la forme d'un agrégat d'assertions, instanciables dans un environnement virtuel, et caractérisant une ou plusieurs dimensions de criticité.

Bien qu'elle soit le résultat du déclenchement d'un ou plusieurs événements, une situation ne contient pas d'information sur la manière dont elle a été créée. S'agissant de situations critiques, il est important de préciser que les événements déclenchés, pris indépendamment, ne sont pas forcement critiques. En effet, dans certains cas, la criticité émerge non pas de la criticité intrinsèque des événements, mais de leur combinaison étant donné certaines conditions particulières. Par exemple, si l'on considère indépendamment les événements *"Piéton qui traverse"* et *"Voiture qui roule"*, ces derniers sont non critiques. Cependant, leur instanciation simultanée, étant donnée une certaine distance séparant la voiture du piéton, entraîne une situation critique. Nous définissons alors un *scénario critique en environnement virtuel* comme étant :

Scénario critique en environnement virtuel

Un ensemble d'événements particuliers, partiellement ordonnés et instanciables dans un environnement virtuel, et dont l'instanciation entraîne une situation critique.

Dans le cadre de nos travaux de thèse, notre objectif consiste à générer ces scénarios critiques pour instancier deux types de situations en particulier : les dilemmes et les situations ambiguës. Dans la suite de l'état de l'art, nous faisons une revue de littérature de ces deux notions, et présentons en particulier les différents systèmes de simulation de l'état de l'art qui mettent en jeu ces deux dimensions.

2.2 Les dilemmes en théorie

La génération dynamique de situations critiques passe par la génération de situations caractérisées par les dimensions de criticité que nous avons déterminées dans la section précédente. Parmi ces dimensions, nous avons identifié le dilemme. Dans l'usage courant, le terme dilemme est souvent utilisé pour décrire un choix difficile. Dans cette section, nous étudions plus en détails les différentes acceptions de ce terme dans la littérature. En général, c'est une notion qui soulève de nombreux questionnements en sciences humaines et sociales. L'existence des dilemmes moraux, par exemple, est remise en cause par plusieurs auteurs pour différentes raisons. L'objectif ici n'est pas de contribuer à ce débat, mais de simplement exposer une brève revue de la littérature sur le sujet ainsi que sur les principales notions qui s'y rapportent.

2.2.1 Définition générale

Nous avons recensé dans la littérature un ensemble de situations dilemmatiques célèbres. Le dilemme Cornélien [1], le Choix de Sophie [2] et le dilemme du prisonnier [3], font partie des dilemmes les plus connus. La particularité de ces situations réside dans la nature du choix auquel est confronté le protagoniste dans ces histoires. Dans ce choix, toutes les issues possibles amènent à des conséquences négatives. Dans le dilemme Cornélien par exemple, si le protagoniste choisit de venger son père, il perd sa bien-aimée. Tandis que s'il choisit de préserver son amour, il trahit la mémoire de son père. Par conséquent, il est perdant dans les deux cas. S'agissant d'un choix entre deux options, le dictionnaire Larousse définit le dilemme comme étant une situation où le sujet est dans *l'obligation de choisir entre deux partis qui comportent l'un et l'autre des inconvénients*. Dans un cadre plus général, il s'agit alors de situations sans échappatoire, pour lesquelles il n'existe pas de "bonne" solution. Classiquement, les dilemmes impliquent un seul décideur, ils sont alors *intrapersonnels* (e.g. le choix de Sophie). Toutefois, dans certains cas, à l'instar du dilemme du prisonnier, ils peuvent impliquer plus d'un décideur, il s'agit alors de dilemmes *interpersonnels* (Marcus,

1. Ce dilemme est appelé ainsi du nom de son inventeur Pierre Corneille. Ce dernier l'illustre dans plusieurs de ses œuvres. Dans le Cid par exemple, Rodrigue, qui doit épouser Chimène dont il est amoureux, doit venger l'honneur de son père « bafoué » par le père de Chimène. Rodrigue est confronté au dilemme suivant : obéir à son devoir et tuer le père de sa promise mais perdre son amour, ou refuser de venger son père — porter toute sa vie la marque de la lâcheté — et garder l'amour de sa promise.

2. Une mère doit choisir lequel de ses deux enfants sauver. Si aucun choix n'est fait, les deux enfants sont assassinés.

3. Deux prisonniers sont retenus dans deux cellules séparées. (1) Si l'un des deux dénonce l'autre, le prisonnier dénoncé écope de 10 ans tandis que l'autre est libéré. (2) Si les deux se dénoncent entre eux, ils écopent chacun de 5 ans d'emprisonnement. (3) S'ils refusent de dénoncer, ils écopent de la peine minimale qui est de six mois.

1980; McConnell, 1988). Les dilemmes se distinguent également par leur origine. Il arrive qu'ils soient le résultat d'une ou de plusieurs actions du décideur (origine interne). Tandis que dans certains cas, ils sont plutôt le produit de l'environnement dans lequel il évolue (origine externe) (McConnell, 1978).

Face à ces dilemmes, le décideur doit faire un choix. Dans certains cas, ce choix met en conflit son système de valeurs, il s'agit alors d'un dilemme moral ou éthique. Ce type spécifique de dilemme est discuté dans la section suivante.

2.2.2 Les dilemmes moraux

L'existence ou non des dilemmes moraux constitue un sujet de débat parmi les philosophes. Notre objectif ici n'est pas de contribuer à ce débat mais d'exposer les différents éléments qui nous ont guidé dans notre réflexion pour construire un modèle informatique de génération de dilemmes moraux.

(Brink, 1994) définit un dilemme moral comme étant une situation impliquant *un conflit d'obligations*. Il distingue toutefois entre deux types d'obligation : les obligations *"prima facie"* et les obligations *"all-things-considered"*. Une obligation *prima facie* de faire une action signifie qu'il existe une raison morale de la faire. Cependant, cette obligation peut être défaite par d'autres plus fortes. Si une obligation *prima facie* est supérieure à toutes les autres, elle est considérée comme une obligation *all-things-considered*. Pour Brink, un véritable dilemme moral ne survient que lorsque deux obligations *all-things-considered* sont en jeu. C'est-à-dire lorsqu'aucune d'entre elles ne prend le dessus sur l'autre (Sinnott-Armstrong, 1988). Il qualifie également ce genre de situations de "conflits insolubles". Dans le cas où une obligation prend le dessus sur les autres, ces situations sont considérées, toujours selon l'auteur, comme des conflits solubles qui ne sont pas de véritables dilemmes moraux. Toutefois, cet avis n'est pas partagé par (Marcus, 1980; Williams and Atkinson, 1965). Pour eux, les conflits solubles correspondent à des dilemmes moraux si les sujets expriment du remord ou de la culpabilité après leur choix.

Dans un registre plus formel, les personnes qui s'opposent à l'existence des dilemmes moraux, dont (Conee, 1982; Zimmerman, 1995), soutiennent leur propos par le fait qu'il est impossible de représenter formellement les dilemmes en logique déontique [4]. En effet, pour que cela soit possible, il faut abandonner un des deux principes fondamentaux qui sont : le *principe de la consistence déontique* [5] et le *principe de la logique déontique* [6]. (Vallentyne,

4. La logique déontique est une logique modale normale comportant quatre modalités : O, Per, F et Op qui se réfèrent respectivement au caractère obligatoire, permis, interdit et optionnel d'une proposition logique (e.g. OA veut dire que l'action A est obligatoire).

5. $OA \longrightarrow \neg O \neg A$ | une même action ne peut pas être à la fois obligatoire et prohibée.

6. $(A \longrightarrow B) \longrightarrow (OA \longrightarrow OB)$ | si une action A implique B, alors si A est obligatoire, B l'est aussi.

1989) partage en partie ce point de vue. Pour lui, cela ne concerne que les dilemmes qu'il qualifie de **dilemmes d'obligation**. Ces derniers font référence aux situations où toutes les actions sont obligatoires. L'auteur les distinguent des **dilemmes de prohibition** qui se réfèrent aux situations où toutes les actions sont prohibées.

2.2.3 Les valeurs morales

Face à un dilemme moral, chaque individu agit différemment selon son système de valeurs. Pour (Schwartz, 1992; Williams, 1968), les valeurs sont des objectifs désirables, qui différent selon leur importance, et qui servent de principes directeurs dans la vie des personnes. En 1922, Eduard Spranger définit une typologie de valeurs comprenant six dimensions (Chataigne, 2014) : théorique (recherche de la vérité et du savoir), économiques (l'intérêt porte sur l'utilité pratique), esthétiques (évaluation basée sur la beauté), sociales (valorisation de la sympathie, l'intégrité et l'altruisme), politiques (évaluation basée sur le pouvoir personnel) et religieuse (recherche de la compréhension du cosmos). Cette typologie fût opérationnalisée par Vernon et Allport en 1931 (Chataigne, 2014) dans un questionnaire pour déterminer quelles valeurs prédominent dans une personnalité donnée. Plus tard, (Rokeach, 1973) définit dans son inventaire, 36 valeurs qu'il répartit en deux types : les valeurs instrumentales qui se rapportent aux modes de comportement (e.g. courage, politesse) et les valeurs terminales qui sont les buts de l'existence (e.g. bonheur, liberté). Ces travaux ont inspiré ceux de (Schwartz, 1992, 2006) qui ont donné lieu à une des théories des valeurs les plus complètes et les plus citées de la littérature : la théorie des valeurs universelles de Schwartz. Cette théorie repose sur des données provenant de 67 pays. Dans cette théorie, les valeurs se distinguent l'une de l'autre par le type d'objectifs motivationnels qu'elles expriment. A partir de ces objectifs, l'auteur décrit 10 grandes valeurs de base : l'Autonomie, l'Universalisme, la Bienveillance, la Tradition, la Conformité, la Sécurité, le Pouvoir, la Réussite, l'Hédonisme et la Stimulation. Ces grandes valeurs regroupent un ensemble d'items représentées dans une structure circulaire qui illustre la dynamique d'opposition et de compatibilité entre ses valeurs (cf. Figure 2.4). Plus deux valeurs sont proches dans la structure circulaire, plus elles sont compatibles. Au contraire, plus deux valeurs sont distantes, plus elles sont antagonistes. Par exemple, la Stimulation s'oppose à la Conformité, tandis que l'Universalisme se concilie avec la Bienveillance. Ces relations d'opposition et de compatibilité sont structurées selon deux grandes dimensions. La première oppose l'Ouverture au changement à la Continuité, tandis que la deuxième oppose l'Affirmation de soi au Dépassement de soi (cf. Figure 2.5).

Nous tenons à préciser que cette structure rend compte des relations entre les valeurs et non pas de l'importance attribuée à chacune d'elles par les individus. En effet, chaque

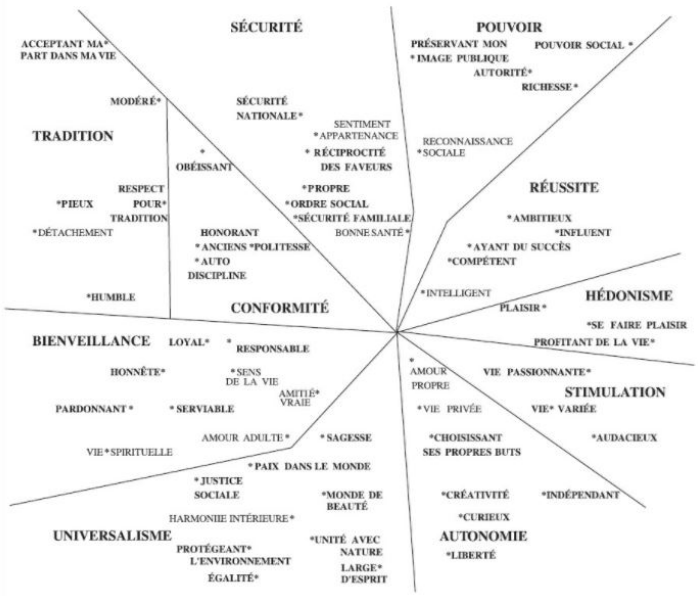

FIGURE 2.4 Les valeurs de bases et items de la théorie des Valeurs Universelles Schwartz (2006)

personne dispose de sa propre hiérarchie de valeurs en fonction de l'importance qui leur est accordée.

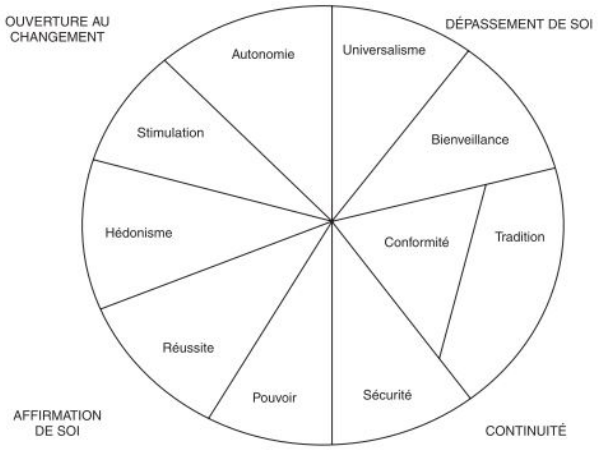

FIGURE 2.5 Valeurs de bases et dimensions

2.2.4 Les dilemmes, instruments de recherche et de formation

Les dilemmes furent un instrument utilisé par de nombreux chercheurs dans leurs travaux. Parmi les travaux les plus célèbres de la littérature, ceux du psychologue (Kohlberg, 1958). Ce dernier fit usage des dilemmes pour définir les six stades de sa fameuse théorie

du développement moral. En effet, il confronta ses sujets à un ensemble d'histoires qui constituent des dilemmes moraux, pour les amener au maximum de leur réflexion éthique. Face à ces dilemmes, les sujets devaient répondre par oui ou par non si le protagoniste de l'histoire devait réaliser l'action demandée. Ce qui importait le plus, ce n'était pas tant la réponse à cette question, mais la justification donné par les sujets. Ces travaux de Kohlberg sur le développement moral ont inspiré plusieurs programmes scolaires, notamment les cours d'éducation civique et morale en France [7].

Dans la vie réelle, ces situations dilemmatiques sont rencontrées quotidiennement dans différents domaines. Nous nous intéressons à la médecine et à la conduite automobile. En médecine, les infirmières sont confrontées quotidiennement à des dilemmes éthiques. Dans son article, (Lecomte, 2006) explique pourquoi les infirmières devraient avoir une réflexion éthique. Elle cite en exemple le fait de pouvoir garder une vue holistique de la personne, d'être capable de négociation et de compromis, mais aussi de prendre de la distance et d'avoir un autre regard pour prendre des décisions. En formation initiale, l'étudiant est sensibilisé à l'éthique mais seul un expert a la capacité de "résoudre" un dilemme éthique. En effet, la *compétence éthique* se construit lentement à partir de la formation initiale et tout au long de sa carrière professionnelle en progressant par l'expérience (Bertholet, 2000). Tout l'intérêt de la simulation consiste à exploiter ses outils et méthodes afin de produire certains dilemmes qui, dans une configuration réelle, seraient éthiquement inacceptables à mettre en place. En conduite automobile, le cas de figure le plus célèbre est celui de la collision inévitable où le conducteur doit choisir entre percuter des piétons ou se sacrifier lui-même ainsi que ses passagers. Ce dilemme fait l'objet de plusieurs études comme celle de (Bonnefon et al., 2016) ou (Noothigattu et al., 2017), dont l'objectif est d'étudier et de déterminer quel comportement un véhicule autonome devrait adopter dans ce cas-là. A l'instar de la médecine, la simulation offre ici un outil permettant de confronter, en particuler, les jeunes conducteurs à des situations dilemmatiques diverses et variées, éthiquement non reproductibles en conditions réelles, leur permettant d'acquérir ainsi suffisamment d'expérience afin de prévenir ce genre de situations.

Par ailleurs, les dilemmes ont été utilisés par différents chercheurs en psychologie sociale (Cushman et al., 2012; Greene et al., 2001). L'expérience de (Francis et al., 2016), par exemple, a mis en évidence la distinction entre le jugement moral d'une personne et son comportement. En effet, les résultats ont montré que le comportement adopté par les sujets en environnement virtuel face à un dilemme particulier était différent de celui adopté en configuration textuelle. Cela confirme que ce qu'on pense devoir faire n'est pas forcement ce que l'on fait en réalité. Cela montre aussi, d'autre part, l'intérêt de l'usage des environnements virtuels pour ce genre de situations.

7. http ://eduscol.education.fr/cid92404/methodes-et-demarches.html

2.2.5 Bilan

Dans cette section, nous avons présenté succinctement les différents éléments de la littérature qui nous ont guidé dans notre réflexion. Notre objectif est de faire usage de ces éléments afin de proposer un modèle générique de génération dynamique des dilemmes, qui soit exploitable par un système de scénarisation informatique. Dans le cadre de nos travaux, nous avons fait le choix de ne considérer que les dilemmes intrapersonnels. Bien qu'ils soient tout aussi intéressants, les dilemmes interpersonnels nécessitent toutefois la prise en compte des problématiques multi-agents, ce qui n'est pas l'objet de nos travaux. Nous avons également choisi, dans un premier temps, de considérer la théorie des Valeurs Universelles de Schwartz comme cadre théorique pour la modélisation des valeurs. La richesse de cette théorie réside dans le fait qu'elle fournit une structure qui rend compte de la dynamique des relations entre les valeurs. Par conséquent, il est possible d'inférer des informations nouvelles à partir de données initiales pauvres. Toutefois, grâce à la généricité du modèle informatique proposé, l'utilisation d'autres théories est tout à fait envisageable.

Avant de décrire notre modèle informatique inspiré par les différents éléments décrits jusqu'ici, il convient tout d'abord de présenter les différents systèmes de l'état de l'art qui se sont intéressés à l'intégration des dilemmes dans les simulations. Dans la section suivante, nous passons en revue ces travaux informatiques et nous clarifions notre positionnement.

2.3 Dilemme dans les simulations

Le dilemme est une notion qui fait l'objet de plusieurs études dans les sciences humaines et sociales. L'intérêt pour cette notion s'est accru avec les développements récents dans le domaine des voitures autonomes. Toutefois, nous remarquons que cette notion n'est pas assez présente dans les simulations destinées à la formation ou l'entraînement. On la retrouve plutôt dans certains systèmes de narration interactive où le dilemme est considéré comme un facteur créateur de tension dramatique. L'analyse des systèmes informatiques de l'état de l'art nous a permis de dégager deux grandes catégories d'approches :

— **les approches scriptées**, qui définissent les dilemmes en amont de l'exécution de la simulation de façon hors-ligne, et
— **les approches génératives**, qui reposent sur un processus automatique de génération de dilemmes.

Dans cette section, nous présentons les différents systèmes qui adoptent ces deux approches, en mettant en exergue les avantages et les inconvénients de chacun.

2.3.1 Les approches scriptées

Les environnements virtuels pour l'entraînement font partie des axes de recherche de l'Institute for Creative Technologies (ICT). Dans certains de leurs travaux, les chercheurs se sont intéressés à la simulation de scénarios militaires complexes où le personnage principal est confronté à des choix difficiles, qui sont parfois sacrifiants. Parmi ces scénarios, celui où l'apprenant joue le rôle d'un lieutenant de l'armée américaine (Rickel et al., 2001). Ce dernier reçoit un ordre radio pour venir en renfort à un peloton en difficulté. Lors du trajet, le lieutenant et sa troupe passent par un village où ils croisent un garçon blessé en situation critique qui a été percuté par un véhicule de la troupe. Cet imprévu confronte le lieutenant à l'alternative suivante : poursuivre son chemin pour venir en renfort à son peloton ou sécuriser une zone d'atterrissage pour un hélicoptère médical pour venir en secours au garçon blessé. Ce dilemme auquel est confronté le lieutenant fait partie du scénario initial de la simulation. Il est scripté et n'est pas sujet à variation.

Dans d'autres travaux de l'ICT, (Gratch and Marsella, 2004) se sont intéressés à la modélisation du comportement émotionnel des agents virtuels. En particulier, ils ont étudié la modélisation du comportement d'un médecin face au cas médical classique de l'accompagnement des mourants par l'administration de fortes doses de morphine. Dans ce scénario, le médecin souhaite prolonger le plus possible la vie de son jeune patient âgé de 11 ans seulement. Cependant, la mère de l'enfant pourrait s'opposer à cette option en souhaitant mettre fin définitivement aux souffrances de son enfant. Cette situation dilemmatique à laquelle le médecin est confronté, fait partie du scénario initial écrit par les auteurs. C'est une situation clé prévue à l'avance et qui n'est pas sujette à variation.

Dans la littérature, nous trouvons aussi un ensemble de travaux sur le célèbre dilemme du tramway. Dans sa version originale, énoncé par (Foot, 1967), un tramway dont les freins sont défaillants roule sur la voie A dans laquelle se trouve cinq ouvriers. Un levier sur le côté permet de dévier le tramway de sa trajectoire initiale et le rediriger vers la voie B où se trouve un seul ouvrier. Que faire dans ce cas ? Sacrifier la personne sur la voie B ou ne rien faire et laisser le tramway suivre son chemin ? Ce dilemme et ses variantes ont fait l'objet de plusieurs études sur papier (Hauser et al., 2007; Valdesolo and Desteno, 2006) qui ont démontré que les sujets adoptaient un comportement utilitariste. C'est-à-dire qu'ils jugeaient que la bonne solution était de sacrifier une personne pour sauver les cinq autres. La première étude qui s'est intéressée à ce dilemme en environnement virtuel était celle de (Navarrete et al., 2012). Cette étude a permis de confirmer le comportement utilitariste observé chez les participants dans les études papier. L'étude concernait ici une situation spécifique, définie en amont. Par conséquent, la simulation ne produisait pas de variations, d'autant plus que cela ne représentait aucun intérêt pour les auteurs. En s'inspirant de cette

étude, (Skulmowski et al., 2014) ont construit leur expérience virtuelle du dilemme du tramway en apportant quelques modifications. Les auteurs se sont intéressés notamment à différentes configuration du dilemme. En effet, en plus du cas classique *un-contre-plusieurs*, ils se sont intéressés au cas du *un-contre-un* où le choix utilitariste n'est plus possible, en faisant varier différents paramètres tels que le sexe, l'ethnie ou l'orientation des victimes par rapport au personnage joueur (qui est le conducteur du tramway). Cette simulation propose une variabilité de situations dilemmatiques qu'on ne retrouve pas dans les systèmes présentés jusque-là. Cependant, elles sont toutes prévues à l'avance et font partie du scénario initial.

Dans des travaux plus récents, (Sütfeld et al., 2017) se sont intéressés aux dilemmes en conduite automobile. Plus particulièrement, aux situations de collision inévitable où le conducteur doit choisir lequel des deux obstacles "sacrifier". L'objectif des auteurs était d'investiguer quel était le modèle de prise de décision le plus pertinent pour ce genre de situations, dans une optique d'implémentation de modèle dans les algorithmes de décisions des véhicules autonomes. Ils ont donc comparé trois modèles et examiné lequel d'entre-eux permettait de prédire le mieux les comportements humains face à ces situations de collision inévitable. Un total de 17 obstacles, de différentes natures, a été prévu pour cette expérience : des obstacles humains (e.g. petit garçon, jeune femme), des animaux (e.g. chien, sanglier) et des objets (e.g. roue, poubelle). A chaque essai, une paire d'obstacles était générée pseudo-aléatoirement en faisant en sorte que chaque obstacle apparaisse au moins une fois durant l'expérience. Semblablement au système précédent, la simulation présente une variété de situations. Cependant, elles sont définies en amont et font partie du scénario initial.

2.3.2 Les approches génératives

La deuxième approche de l'état de l'art consiste à générer de manière automatique et dynamique les dilemmes. A notre connaissance, il n'existe que deux systèmes dans la littérature qui adoptent une telle approche : GADIN et EGAD. Dans cette section, nous présentons ces deux systèmes, en détaillant comment les dilemmes y sont modélisés, et comment ils sont générés à partir de ces modélisations.

GADIN : Les dilemmes par relations sociales

GADIN (Barber, 2008) est un moteur de narration interactive qui déroule une histoire dans une simulation en fonction des choix de l'utilisateur dans les situations de dilemmes. Dans ces situations, l'utilisateur doit choisir entre faire ou ne pas faire une action. Chaque action a une valeur d'utilité propre à chaque personnage du monde virtuel. Par exemple, l'utilité de l'action "Tomber enceinte" pour le personnage X est positive si X souhaite avoir

des enfants, elle est négative sinon. Par ailleurs, les personnages virtuels sont liés par deux types de relations sociales : des relations positives (amis) ou des relations négatives (ennemis). A partir de ces deux éléments clés (l'utilité et les relations sociales), le système distingue cinq catégories de dilemmes :

1. **Trahison**. Situations où l'utilisateur doit choisir entre faire ou ne pas faire une action dont l'utilité lui est positive mais négative pour ses amis. Par exemple : dénoncer son camarade de classe pour bénéficier d'un traitement favorable.
2. **Sacrifice**. Situations où l'utilisateur doit choisir entre faire ou ne pas faire une action dont l'utilité lui est négative mais positive pour ses amis. Par exemple : avouer avoir commis le crime pour lequel son ami a été accusé à tort.
3. **Bien commun**. Situations où l'utilisateur doit choisir entre faire ou ne pas faire une action dont l'utilité est positive à la fois pour ses amis et ses ennemis. Par exemple : dévoiler une information confidentielle ou sensible aux ennemis pour sauver ses amis.
4. **Mal commun**. Situations où l'utilisateur doit choisir entre faire ou ne pas faire une action dont l'utilité est négative à la fois pour lui et ses ennemis. Par exemple : tuer son ennemi sachant qu'il ira en prison pour son acte.
5. **Faveur**. Situations où le choix de l'utilisateur de faire ou ne pas faire une action dont l'utilité est positive pour l'un de ses amis mais négative pour un autre. Par exemple : choisir quel enfant sauver d'une noyade.

Bien que cette approche repose sur une génération automatique, le modèle proposé par les auteurs reste assez limité en terme d'expressivité. En effet, le modèle se restreint à la prise en compte des relations sociales entre les personnages, ce qui écarte un nombre considérable de dilemmes. Nous pensons particulièrement aux situations où le personnage principal n'entretient aucune relation particulière avec les autres ou lorsqu'il est le seul personnage dans l'environnement. Par ailleurs, le modèle ne permet pas de quantifier l'intensité d'un dilemme et donc limite d'une certaine manière l'adaptation.

EGAD : les dilemmes par type de choix

Dans des travaux plus récents, (Harmon, 2016) étend GADIN en proposant EGAD, un système plus expressif qui ne se restreint pas aux relations sociales pour la modélisation des dilemmes. Dans son modèle, l'auteur définit trois listes propres à chaque personnage :

— L, correspond aux valeurs chéries par le personnage ;
— D, correspond aux valeurs méprisées par le personnage ;
— U, correspond aux valeurs dont l'importance pour le personnage est inconnue.

Les valeurs se réfèrent aux entités décrites dans le modèle du monde (e.g. personnages, objets, lieu) et/ou aux valeurs universelles de Schwartz. Par ailleurs, le modèle définit trois types de choix unitaires : *help* , *harm*, *ignore*. Un choix de type *help* déclenche une action qui *préserve* une ou plusieurs valeurs, tandis qu'un choix de type *harm* déclenche une action qui *rejette* une ou plusieurs valeurs. Le troisième type de choix permet au décideur d'ignorer les options proposées. En utilisant ces choix unitaires, le système peut construire des choix composés. Par exemple, si nous supposons que dans une situation dilemmatique deux choix sont proposés au personnage principal, alors étant données a et b deux valeurs, l'espace de possibilité devient :

— Choisir quelle valeur préserver (help) entre a et b, avec $a, b \in [L, U]$
— Choisir quelle valeur rejeter (harm) entre a et b, avec $a, b \in [D]$
— Choisir comment préserver (help) a, avec $a \in [L, U]$
— Choisir comment rejeter (harm) a, avec $a, b \in [D]$

L'auteur explique qu'il est improbable que le décideur veuille rejeter une valeur appartenant à $[L, U]$ ou préserver une valeur qui appartient à $[D]$. Cependant, si le cas se présente, le décideur est confronté à ce que l'auteur définit comme *scenario forcé*. Par conséquent, s'ajoutent à l'espace des possibilités les dilemmes suivants :

— Choisir quelle valeur rejeter (harm) entre a et b, avec $a, b \in [L, U]$
— Choisir quelle valeur préserver (help) entre a et b, avec $a, b \in [D]$
— Choisir comment rejeter (harm) a, avec $a \in [L, U]$
— Choisir comment préserver (help) a, avec $a, b \in [D]$

Bien que EGAD soit plus expressif que GADIN, il ne permet pas de renseigner les conséquences dues à la non-réalisation d'une action. Par ailleurs, le système ne garantit pas le fait que les actions présentées au décideur soient contradictoires. En effet, si le décideur ne peut choisir qu'une seule action parmi celles qui sont proposées, cela n'est pas dû au modèle, mais à l'interface textuelle qui n'autorise pas les choix multiples. Aussi, le modèle ne tire pas pleinement profit des représentations abstraites. Par exemple, le fait d'inclure le concept *Véhicule* dans la liste des valeurs chéries, ne permet pas de dire systématiquement que le personnage aime les *Mercedes*, même si celles-ci héritent de *Véhicule*. Par conséquent, ceci se répercute sur la génération des dilemmes. Si le modèle du monde compte une centaine de voitures de marques différentes, pour que celles-ci puissent être considérées dans la génération d'un dilemme, il faut qu'elles appartiennent toutes à une des listes suivantes : L, D ou U, ce qui nécessite un effort d'écriture supplémentaire inutile. Notons également que le système EGAD n'a pas été validé et que les seules évaluations réalisées par l'auteur concernent l'expressivité du modèle.

2.3.3 Bilan et positionnement

Au travers de cet état de l'art, nous avons pu découvrir les deux familles d'approches utilisées pour la représentation des dilemmes dans les simulations. D'une part, nous retrouvons les approches scriptées qui consistent à décrire les dilemmes en amont de l'exécution de la simulation, et d'autre part, les approches génératives qui consistent à générer automatiquement les dilemmes sans avoir à les décrire au départ. La première approche est souvent utilisée par les concepteurs des systèmes de simulation lorsque le besoin principal n'est pas tant la variabilité des situations dilemmatiques mais la précision de description de ces situations. Il s'agit souvent de travaux en sciences humaines et sociales qui visent à étudier les comportements humains face à ces situations. Cependant, cette approche n'est pas adaptée pour les domaines complexes, plus particulièrement, lorsqu'il est nécessaire de confronter l'utilisateur du système à un large panel de situations. Pour pallier ce problème, seuls deux systèmes de l'état de l'art proposent une approche générative. Cette approche permet d'alléger considérablement l'effort d'écriture initial. Toutefois, le pouvoir génératif des systèmes dépend fortement de la qualité du modèle sur lequel ils reposent.

Nous avons exposé précédemment les limites respectives de chaque système présenté. Nous avons montré que GADIN était limité, à un certain degré, en termes de variabilité, d'adaptativité mais surtout en termes d'expressivité. Les travaux de (Harmon, 2016) ont permis d'étendre ce modèle en améliorant l'expressivité sans pour autant apporter des solutions en termes de variabilité et d'adaptativité. Nous proposons alors KOBA : un système générique plus expressif et plus adaptatif qui offre une plus grande variabilité de situations. Ce système qui constitue notre contribution majeure est détaillé dans la section 3.2. Pour conclure cette section, nous dressons le tableau 2.1 qui reprend les systèmes génératifs de l'état de l'art, en mettant en exergue les points positifs et négatifs de chacun, tout en les comparant avec le système que nous proposons.

	Expressivité	Généricité	Variabilité	Adaptation	Évaluation
GADIN	-	++	+	+	++
EGAD	+	++	+	+	-
KOBA	++	++	++	++	++

TABLE 2.1 Tableau comparatif de notre système avec les approches génératives de l'état de l'art

2.4 L'ambiguïté en théorie

Selon (Weick, 1995), la capacité des personnes à faire face aux situations d'urgence dépend de leur capacité à donner du sens aux situations ambiguës. Ce sont des situations mal structurées, qui prêtent à confusion et qui sont parfois déroutantes (Cottone et al., 2009). Communément, l'ambiguïté est quelque chose d'indésirable que les personnes cherchent à réduire voire à éliminer. (Klinke and Renn, 2002), par exemple, considèrent l'ambiguïté, avec l'incertitude et la complexité, comme les challenges auxquels il faut faire face en évaluation du risque. Cependant, dans certains cas, l'ambiguïté est volontairement introduite, elle est même appréciée. C'est le cas notamment en création artistique où des productions sont volontairement ambiguës pour laisser place à différentes interprétations (Gaver et al., 2003). Dans la littérature scientifique, la notion d'ambiguïté est souvent rapprochée de l'incomplétude, du manque de clarté et de l'imprécision. Certains la considèrent notamment comme une possible combinaison de ces notions (Blockley, 2013). Parfois l'ambiguïté émerge de l'information elle-même (Camerer and Weber, 1992; Ellsberg, 1961). Parfois, elle est plutôt due à la multiplicité des contextes dans lesquels l'information peut être interprétée (Gaver et al., 2003; Johansen and Rausand, 2015). Dans d'autres cas, elle survient, entre autres, à cause des conflits de relations, de valeurs ou de principes des personnes (Catrinu and Nordgård, 2011; March, 1987).

Dans nos travaux, nous considérons l'ambiguïté comme étant une dimension de criticité parmi celles que nous avons définies dans la section 2.1.4. A l'instar des dilemmes, elle constitue également une dimension non chiffrée. Nous nous sommes intéressés alors à la manière dont nous pouvons la modéliser informatiquement, et générer dynamiquement les situations ambiguës à partir de cette modélisation sans avoir à les décrire explicitement à l'avance. La réponse à ces questionnements constitue la deuxième contribution dans le cadre de ce mémoire. Pour nous guider dans notre proposition, nous nous sommes inspirés des différentes conceptions de l'ambiguïté présentes dans la littérature. Dans cette section, nous réalisons tout d'abord une revue de littérature de ces différentes conceptions et définitions. Ensuite, nous présentons quelques simulations qui ont intégré la notion d'ambiguïté.

2.4.1 Ambiguïté et probabilités

Selon le dictionnaire LAROUSSE, un mot ou un énoncé sont ambiguës, lorsque leur "interprétation, leur sens sont incertains ; équivoques". En d'autres termes, et plus généralement, une situation ambiguë est une situation qui laisse place à plusieurs interprétations. C'est d'ailleurs la définition la plus courante qui vient à l'esprit lorsque l'ambiguïté est évoquée. Parmi les définitions les plus citées et les plus reprises de la littérature, nous retrouvons

celle de (Ellsberg, 1961) qui définit l'ambiguïté comme étant "*the quality depending on the amount, type, reliability, and 'unanimity' of information giving rise to one's 'degree of confidence' in an estimate of relative likelihoods*". Cette définition est mise en exergue dans sa fameuse expérience qui donna lieu au paradoxe d'Ellsberg. Dans cette expérience, une urne opaque contient 90 boules de trois couleurs différentes : 30 boules rouges, et 60 boules jaunes et noires dont la distribution est inconnue. Étant donné ces circonstances, les participants doivent choisir lequel de ces deux paris ils préfèrent :

— P_1 : les boules rouges sont gagnantes. Les boules noires et jaunes sont perdantes.

— P_2 : les boules noires sont gagnantes. Les boules rouges et jaunes sont perdantes.

Dans les mêmes circonstances, ils doivent choisir également lequel de ces deux paris ils préfèrent :

— P_3 : les boules rouges et jaunes sont gagnantes. Les boules noires sont perdantes.

— P_4 : les boules noires et jaunes sont gagnantes. Les boules rouges sont perdantes.

Le comportement observé chez la plupart des participants était qu'ils avaient plus tendance à préférer les paris P_1 et P_4 respectivement aux paris P_2 et P_3. Ellsberg explique ce comportement par le fait que les personnes ont tendance à faire confiance au choix dont elles connaissent la probabilité de distribution. Ne pas connaître une information pertinente est perturbant, voire angoissant. Cela rend les individus hésitants lorsqu'il s'agit de parier (Camerer and Weber, 1992; Heath and Tversky, 1991). En effet, les personnes ayant préféré P_1 à P_2, ont pris leur décision tout en sachant que le nombre de boules noires pouvaient potentiellement être le double — à une boule près — du nombre de boules rouges. De même, les personnes ayant préféré P_4 à P_3, ont pris leur décision tout en sachant qu'il pouvait potentiellement n'exister qu'une seule boule noire perdante. Cette absence d'information sur la distribution de probabilité est ce qui est qualifié d'ambiguïté dans certains cas. Cela amène, par conséquent, à la distinction entre les probabilités connues (risque), et les probabilités inconnues (ambiguïté ou incertitude selon les auteurs) (Garner, 1962; Luce and Raiffa, 1989). Une définition plus complète que celle d'Ellsberg décrit l'ambiguïté comme étant l'incertitude sur les probabilités créée par le manque d'informations pertinentes qui peuvent être connues (Fellner, 1961; Frisch and Baron, 1988). Cette définition met l'accent sur un point très important : l'information qui manque peut être connue. En d'autres termes, l'ambiguïté est quelque chose de réductible. Considérons, par exemple, un cas classique en ambiguïté linguistique. X confirme à Y que leur rendez-vous est fixé pour jeudi. Cette information telle que présentée par X, soulève une certaine ambiguïté pour Y. Il ne sait pas si le jeudi dont il est question correspond au lendemain ou à celui de la semaine prochaine. Cette incertitude par rapport au jour peut être réduite par Y en demandant à X de quel jeudi il s'agit. Considérons maintenant le cas d'une personne qui vient de se renseigner sur les

conditions météorologiques. Les prévisions annoncent qu'il pleuvra probablement demain. Cette situation n'est pas ambiguë dans le sens où cette incertitude n'est pas réductible car elle dépend des variations naturelles des phénomènes.

2.4.2 Ambiguïté et incertitudes

Il convient alors de préciser que l'incertitude dont il est question ici, correspond à l'incertitude épistémique — ou subjective. Celle-ci est engendrée, entre autres, par le manque d'information (Duncan, 1972; Galbraith and R., 1973; Miller and Frick, 1949) auquel on pourrait remédier. Cette incertitude est traditionnellement distinguée de l'incertitude aléatoire — ou objective. Cette dernière est engendrée par la variabilité naturelle des événements (e.g. fréquence d'occurrence des catastrophes naturelles). Par conséquent, elle est non réductible.

L'ambiguïté se résume-t-elle donc à une incertitude épistémique ? Oui et non. Oui, parce que l'ambiguïté est le résultat d'une incertitude subjective et réductible sur l'information. Et non, parce que tout ce qui est incertain n'est pas forcement ambigu. L'incertitude et l'ambiguïté ont fait l'objet des travaux de (Schrader et al., 1993) qui se sont intéressés à la différence entre ces deux notions dans le cadre de la résolution de problèmes techniques. Face à un problème, chaque individu dispose d'un modèle mental qui le guide dans sa résolution (Mintzberg et al., 1976; Simon, 1987). Un modèle mental bien défini, prédétermine l'ensemble des solutions possibles (Clement, 1989). C'est à dire, pour chaque combinaison de variables, l'individu peut identifier, le cas échéant, quelles procédures et algorithmes de résolutions mettre en place. Lorsqu'il s'agit de situations incertaines, la réduction de l'incertitude consiste à collecter plus d'informations. En d'autres termes, cela consiste à s'informer sur les valeurs des variables du problème. Lorsqu'il s'agit de situations ambiguës, le processus de résolution consiste plutôt à repenser et/ou reconstruire les modèles mentaux. C'est à dire, revoir les relations entre les variables du problème et les algorithmes de résolution. A partir de ces éléments, selon le degré de connaissance des variables, de leur valeur et des méthodes de résolution, les auteurs définissent l'incertitude et deux niveaux d'ambiguïté :

- **l'incertitude :** elle caractérise les situations où l'ensemble des variables pertinentes du problème sont connues, mais dont les valeurs ne sont pas précisément déterminées ;
- **l'ambiguïté de niveau 1 :** elle caractérise les situations où les variables pertinentes du problème sont connues. Cependant, la relation entre ces variables et l'algorithme de résolution manque de clarté ;
- **l'ambiguïté de niveau 2 :** elle caractérise les situations où l'ensemble des variables pertinentes du problème ainsi que leurs relations avec l'algorithme de résolution ne sont pas connues.

Ces éléments permettent de dresser une matrice incertitude-ambiguïté (cf. figure 2.6), qui fait ressortir cinq cas de figure selon le degré d'incertitude et d'ambiguïté.

Réduction d'incertitude ←

	Incertitude faible	Incertitude forte
Ambiguïté faible	**Cas 1** • Variables connues • Valeurs connues • Relations connues	**Cas 2** • Variables connues • Valeurs inconnues • Relations connues
Ambiguïté forte Ambiguïté (niveau 1)	**Cas 3** • Variables connues • Valeurs connues • Relations inconnues	**Cas 4** • Variables connues • Valeurs inconnues • Relations inconnues
Ambiguïté (niveau 2)		**Cas 5** • Variables inconnues • Relations inconnues

↑ Réduction d'ambiguïté

FIGURE 2.6 Matrice incertitude-ambiguïté (Schrader et al., 1993)

Afin d'illustrer ces trois définitions, ainsi que les cinq cas de figure de la matrice, prenons l'exemple de la gestion d'un feu rouge (bien que cela ne se rapporte pas exactement à une situation de résolution de problème technique). La variable pertinente dans ce cas est la couleur du feu. Dans le modèle mental d'un individu lambda, on peut supposer qu'à chaque couleur est associée une action : s'arrêter lorsque le feu est rouge, rouler lorsque le feu est vert et s'apprêter à s'arrêter lorsque le feu est orange. Le cas 1 se réfère au cas où le feu est rouge — la variable et sa valeur sont connues — et où l'individu sait que dans ce cas-là il faut qu'il s'arrête —-relations connues. Lorsque le feu est occulté par un élément (e.g. camion, arbre) qui rend impossible l'identification de la couleur allumée, cela se réfère au cas 2. L'individu peut réduire cette incertitude en analysant le comportement des piétons et/ou le comportement des autres véhicules qui est conditionné par les autres feux de circulation présents dans le carrefour. Le cas 3, peut se référer à une situation où la couleur du feu ne fait pas partie des trois couleurs traditionnelles pour lesquels les procédures sont connues — relations inconnues. Dans ce cas-là, pour réduire cette ambiguïté, la recherche d'information n'a aucun effet, car les variables et leurs valeurs sont déjà connues. Il faut plutôt repenser son modèle mental. L'individu peut notamment envisager dans cette situation de raisonner en fonction de la position de la couleur allumée. Maintenant, considérons que le feu est défaillant (aucune couleur n'est allumée) ou occulté comme dans le cas 2, mais qu'en plus, l'individu n'associe aucune action à la variable couleur du feu. Cela peut se rapporter, bien que très

improbable, au cas où la personne est débutante et ne maîtrise pas encore suffisamment le code de la route. Ici, la variable est connue, mais sa valeur ainsi que sa relation avec les actions à entreprendre sont inconnues. Cela correspond au cas 4. Finalement, le cas 5 est un cas extrême où les valeurs et leurs relations sont inconnues. Cela peut se référer à une situation où il devrait y avoir un feu rouge en théorie. Les variables pour gérer la situation sont alors inconnues. Il faudrait donc tout d'abord les déterminer. L'individu peut alors présumer que dans cette situation là, c'est la règle de la priorité à droite qui doit être considérée.

Cette conception de l'ambiguïté s'approche de celle de (Blockley, 2013) qui considère l'ambiguïté comme une caractéristique de l'incertitude. L'auteur définit cette dernière par trois attributs distincts, qui constituent les axes d'un espace orthogonal tridimensionnel (cf. figure 2.7) : le flou (Fuzziness), l'incomplétude (Incompleteness) et l'aléatoire (Randomness). Les deux premiers axes se réfèrent à l'incertitude épistémique, tandis que le dernier désigne l'incertitude aléatoire. L'interaction entre les différents axes de l'espace, engendre des points clés qui, selon les contextes, peuvent avoir une signification sémantique particulière. Dans cet espace, l'ambiguïté est le résultat du mixte entre le flou et l'incomplétude. Cela se rapporte aux cas 4 et 5 de la matrice incertitude-ambiguïté.

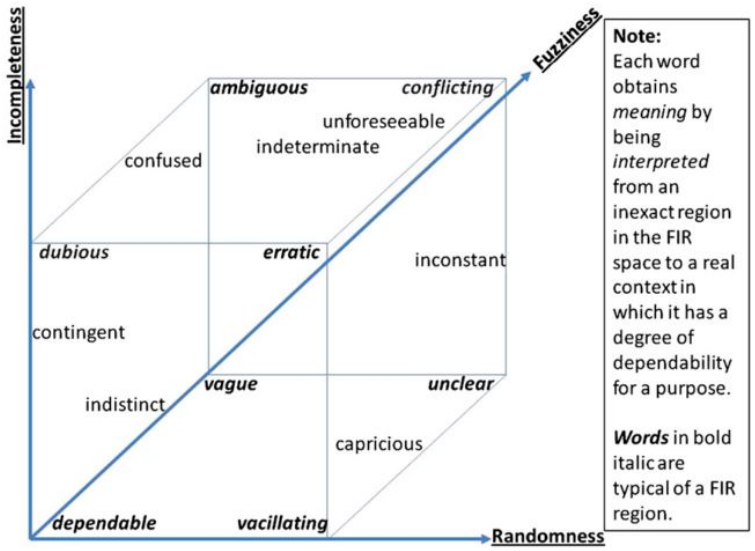

FIGURE 2.7 Espace Fuzziness-Incompleteness-Randomness (FIR) de (Blockley, 2013)

2.4.3 Ambiguïté, contexte et relations

A ce stade, toutes les représentations discutées jusque-là considèrent l'information comme étant la principale source d'ambiguïté. Cependant, dans certains cas, il arrive que l'ambiguïté émerge non pas parce que l'information en soi est floue, incomplète ou incertaine, mais

parce qu'elle est interprétée différemment selon le contexte dans lequel elle se trouve ou dans lequel elle est considérée. En analyse de risque par exemple, l'IRGC [8] (Renn, 2008) définit l'ambiguïté interprétative. Celle-ci se réfère à l'existence de différentes interprétations d'un même résultat issue d'une évaluation. Cela est dû au fait que les décideurs ont des conceptions du risque qui sont différentes. Par exemple, un même résultat peut avoir deux interprétations différentes et légitimes selon que le décideur est un expert ou un débutant. Il est important de souligner qu'ici l'ambiguïté est interpersonnelle. C'est-à-dire qu'il existe une interprétation unique pour chaque individu qui, toutefois, est différente de l'un à l'autre. En effet, pour le débutant comme pour l'expert, le résultat ne présente aucune ambiguïté. C'est plutôt le fait que le résultat soit interprété différemment par ces deux décideurs qui le rend ambigu. Cela n'écarte pas pour autant le fait que, dans certains cas, l'ambiguïté du contexte peut être intrapersonnelle. C'est-à-dire que chaque individu peut avoir lui-même différentes interprétations. C'est le cas notamment lorsque deux contextes incompatibles surviennent. Reprenons l'exemple du feu rouge décrit plus haut. Considérons que cette fois-ci, tous les feux sont allumés. Que doit faire l'individu dans ce cas-là : considérer que le feu est rouge ou orange, et par conséquent, s'arrêter ? ou considérer qu'il est vert, et rouler ? Cela dépend du contexte que l'individu décide de considérer.

Très intimement liée à cette ambiguïté contextuelle, nous retrouvons celle qui émerge des relations (spirituelles, personnelles, professionnelles ...), des valeurs (éthiques, morales, humaines...) et des normes (sociales, industrielles, scientifiques ...) (Johansen and Rausand, 2015) propres à chaque personne. En analyse de risque par exemple, un même résultat peut être interprété différemment selon que le décideur est, par exemple, un partisan des verts ou propriétaire d'une compagnie pétrolière. L'IRGC qualifie ce type d'ambiguïté de normative.

2.4.4 Bilan

Cette revue de la littérature, nous a permis de mettre en exergue que l'ambiguïté pouvait provenir de l'information elle-même, du contexte dans lequel elle apparaît ou des relations qu'entretient l'individu avec elle. Cela nous amène donc à considérer l'ambiguïté telle que décrite par (Gaver et al., 2003). Dans leur conception de l'ambiguïté, les auteurs la classent en trois catégories : l'ambiguïté de l'information, l'ambiguïté du contexte et l'ambiguïté des relations. Nous préférons toutefois l'utilisation de l'appellation ambiguïté normative, qui est plus large que l'ambiguïté des relations. Nous définissons alors les trois types d'ambiguïté comme suit :

8. International Risk Governance Council

Ambiguïté de l'information

L'ambiguïté de l'information survient lorsque de multiples interprétations sont engendrées par le manque remédiable d'informations pertinentes et/ou la qualité des informations à disposition.

La réduction de ce premier type d'ambiguïté passe par la recherche d'informations pertinentes qui pourraient compléter la base d'informations disponibles et/ou en cherchant à apporter plus de clarté et de précision à celles déjà disponibles.

Ambiguïté du contexte

L'ambiguïté du contexte survient lorsqu'une même information, claire et précise, est interprétée différemment selon le contexte dans lequel elle apparaît.

La réduction de ce type d'ambiguïté passe par la reconstruction ou la mise à jour des modèles mentaux en essayant, par exemple, de concilier les contextes ou privilégier les uns aux autres.

Ambiguïté normative

L'ambiguïté normative survient lorsqu'une même information, claire et précise, est interprétée différemment selon les normes, les relations et les valeurs propres à chaque individu.

La réduction de ce type d'ambiguïté passe par la reconsidération et/ou la conciliation des croyances, des valeurs et des principes des individus.

Cette catégorisation nous parait pertinente pour une génération automatique, dans le sens où cela permet de distinguer précisément les potentielles sources d'ambiguïté. Par conséquent, cela permet de bien définir les algorithmes de génération qui cibleront les éléments pertinents pour chaque type.

2.5 L'ambiguïté dans les simulations

Après avoir réalisé une revue des différentes conceptions de l'ambiguïté dans la littérature, nous nous sommes intéressés particulièrement aux simulations qui mettent en jeu cette notion. Notre première remarque est que le nombre de ces simulations est très faible. En effet, peu de travaux se sont intéressés à sa mise en œuvre dans les environnements virtuels pour l'entraînement ou la formation. Dans cette section, nous présentons quelques systèmes que nous avons recensés, ensuite nous clarifions notre positionnement par rapport à ce qu'ils proposent.

(Mantovani et al., 2001) ont mené une étude pour explorer la pertinence de l'utilisation des environnements virtuels pour la formation à la sécurité dans les grands lieux publics. Leur étude consistait, d'une part, à tester l'efficacité de différents types de signaux de guidage vers les issues de secours, et d'autre part, à analyser comment les normes sociales incarnées dans les simulations affectent les mouvements en environnement virtuel. Pour leur expérience, les auteurs ont conçu une représentation virtuelle d'une bibliothèque universitaire. L'objectif des 60 personnes qui ont participé à l'expérience était de retrouver rapidement l'issue de secours, en partant d'un point de départ, et en se laissant guider par deux types de signaux : des flèches rouges fixes (indication standard fixée aux murs) ou des flèches vertes mouvantes. Les auteurs ont évalué l'efficacité de ces signaux dans trois conditions différentes : petite, moyenne et grande distance (cf. figure 2.8). Tous les utilisateurs ont réussi à atteindre l'issue. Le groupe "flèche mouvante" était, en moyenne, plus performant mais sans différence significative. Les auteurs ont cependant remarqué que le groupe évoluant dans la condition "moyenne distance" et utilisant les flèches rouges fixes, s'est confronté à une situation ambiguë. Un ruban rouge est interposé entre le joueur et la sortie. En situation normale, ce ruban est utilisé pour contrôler le flux des personnes entrantes et sortantes. Ici, bien qu'il s'agisse d'une situation d'urgence, et bien que le ruban soit physiquement dépassable, cela a poussé certains sujets à rebrousser chemin et chercher une autre sortie. Il s'agit ici d'une ambiguïté contextuelle qui met en jeu deux contextes que les sujets hiérarchisent chacun à sa manière. Dans cette simulation, la situation ambiguë fait partie du scénario initial. Elle est donc décrite à l'avance.

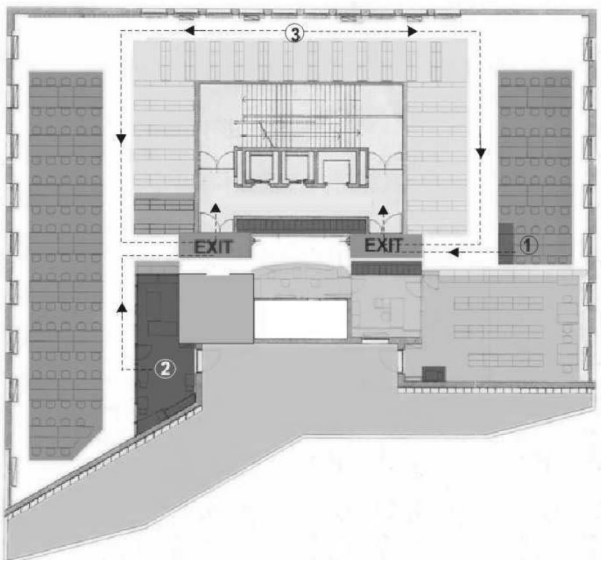

FIGURE 2.8 Plan de la bibliothèque. (1) petite distante, (2) moyenne distance et (3) grande distance.

De leur côté, (Cottone et al., 2009) se sont intéressés à la coopération entre des couples de participants dans des situations ambiguës en environnement virtuel. Les sujets évoluent dans une ville virtuelle qu'ils découvrent pour la première fois. L'objectif de chaque couple est de se retrouver à un endroit de leur choix. Ils ont été répartis en trois groupes. Chaque groupe dispose d'un moyen de communication spécifique : face-à-face, téléphone ou chat. La ville a été volontairement peuplée d'endroits plus ou moins semblables afin de créer une ambiguïté chez les participants. Par exemple, la figure 2.9 montre une boutique de vêtements "Tomyno" qui est présente à deux endroits différents mais dans lesquels elle est visuellement presque identique. Ce genre de disposition est typiquement une potentielle source d'ambiguïté. Il s'avère, qu'en effet, dans certains cas, les participants pensaient être au même endroit alors qu'ils ne l'étaient pas en réalité. Cela produit une ambiguïté du fait que les participants s'attendent à voir leur camarade à l'emplacement convenu. Cette incohérence qui amène également de la confusion, pousse les participants à rechercher des solutions pour l'éliminer, en essayant de trouver, entre autres, des indices environnementaux plus spécifiques. A l'instar de la première simulation, l'ambiguïté ici est prévue à l'avance. Les lieux quasi-identiques ont volontairement été placés à différents endroits de la ville, et cela avant l'exécution de la simulation.

FIGURE 2.9 Lieux visuellement quasi-identiques

Dans le domaine militaire, la gestion de situations critiques nécessite des compétences particulières. Dans le cadre d'un programme d'entraînement, (Raybourn et al., 2005) ont développé une simulation dont l'objectif est d'entraîner les leaders des forces spéciales à gérer des situations critiques en développant des compétences non-techniques telles que la négociation, la communication effective, mais aussi l'analyse de situations ambiguës. La simulation consiste en des scénarios militaires scriptés pilotables par un instructeur qui peut à tout moment déclencher des événements particuliers. Cependant, les auteurs ne donnent aucune description de situations ambiguës spécifiques. En tout cas, l'ensemble des scénarios,

ambigus ou non, sont scriptés, et donc prévus à l'avance. L'instructeur peut intervenir au cours de la simulation, mais aucune adaptation ou génération automatique ne sont réalisées par le système informatique.

2.6 Bilan

La criticité est une notion qui renvoie à différentes acceptions. Dans le cadre de nos travaux, nous l'avons caractérisée par un ensemble de dimensions dont le dilemme et l'ambiguïté.

L'état de l'art sur les dilemmes nous a permis de mettre en exergue qu'il y avait une distinction entre les dilemmes d'obligation et les dilemmes de prohibition. Par ailleurs, s'agissant des dilemmes moraux, nous avons réalisé une revue de la littérature sur les différentes théories des valeurs existantes. Cela nous a amené à considérer, dans le cadre de nos travaux, la théorie des valeurs universelles de (Schwartz, 2006) qui est la théorie la plus aboutie et la plus étudiée de la littérature. S'agissant des dilemmes dans les simulations, l'analyse des systèmes informatiques de l'état de l'art nous a permis de distinguer deux approches : l'approche scriptée et l'approche générative. L'approche scriptée permet un contrôle fin sur les situations dilemmatiques à produire. Toutefois, elle reste inadaptée pour les domaines complexes, étant donné l'effort d'écriture considérable nécessaire. Les approches scriptées pallient ce problème. Cependant, les systèmes de la littérature qui adoptent une telle approche ont du mal à concilier les critères d'expressivité, de généricité, de variabilité et d'adaptation.

L'ambiguïté est une notion qui a beaucoup été étudiée dans la littérature. Les travaux précédents ont donné lieu à différentes conceptions qui la rapprochent, tout en la distinguant, des concepts comme l'incertitude, l'imprécision ou l'incomplétude. Par ailleurs, il a été montré que l'ambiguïté émerge de trois différentes sources : l'information, le contexte dans lequel l'information est interprétée, et les relations qui lient les individus à l'information. Cependant, bien que cette notion soit largement discutée en théorie, elle demeure quasi-absente dans les environnements virtuels pour l'apprentissage. L'état de l'art que nous avons réalisé, ne nous a permis d'identifier qu'un nombre faible de systèmes mettant en jeu cette notion. Par ailleurs, aucun de ces systèmes ne propose un modèle de génération dynamique de l'ambiguïté. Les situations ambiguës qu'ils mettent en jeu sont prévues à l'avance, et font partie du scénario initial. Cette approche n'est pas adaptée pour les domaines complexes, surtout, lorsque il est nécessaire de confronter l'apprenant à un large panel de situations. Les situations ambiguës doivent donc émerger au cours de la simulation sans avoir à être décrites au préalable. Pour cela, nous proposons de générer dynamiquement les situations ambiguës à

partir de modèles de connaissances. A notre connaissance, nos travaux constituent la première tentative de mettre en œuvre une telle approche.

Chapitre 3

Contribution

Sommaire

3.1 Architecture 56
3.1.1 Modules externes 56
3.1.2 Modules internes 58
3.1.3 Modèles sémantiques pour la scénarisation 61
3.1.4 Bilan 69
3.2 KOBA : moteur de génération automatique de dilemmes 70
3.2.1 Les conséquences négatives 72
3.2.2 Le profil moral 75
3.2.3 Formalisation des types de dilemmes 77
3.2.4 Les étapes de génération 81
3.2.5 Bilan 97
3.3 Génération de l'ambiguité 97
3.3.1 Approche 98
3.3.2 Représentation mentale incertaine de l'apprenant 98
3.3.3 Génération de l'ambiguïté de l'information 102
3.3.4 Génération de l'ambiguïté du contexte 106
3.3.5 Bilan 109

Notre problématique porte sur la scénarisation d'environnements virtuels, et plus particulièrement sur la génération dynamique des situations critiques. Nous nous intéressons précisément à la génération des situations dilemmatiques et ambiguës. La difficulté de nos travaux consiste à concilier la **généricité** d'un système de génération, son **expressivité**, sa capacité à proposer une **variabilité** de situations ainsi que son pouvoir **adaptatif**. L'analyse des systèmes informatiques de l'état de l'art nous a permis de constater que les systèmes

existants ont du mal à concilier ces différents aspects. Nous proposons alors, en nous basant sur les éléments de la littérature, une formalisation des dilemmes et de l'ambiguïté. A partir de cette formalisation, nous définissons différents algorithmes génériques qui raisonnent sur des modèles de connaissances afin d'extraire des objectifs scénaristiques permettant l'émergence de situations dilemmatiques et ambiguës. Le processus de génération repose sur des modèles experts expressifs qui permettent de représenter des relations complexes et rendent possible le raisonnement à différents niveaux d'abstraction. Cela permet également de produire une plus grande variabilité de situations. Par ailleurs, le contexte d'entraînement dans lequel nous nous plaçons, implique la prise en compte de contraintes pédagogiques issues du diagnostic de l'apprenant. Aussi, s'agissant de la génération de dilemmes moraux, il est nécessaire de prendre en compte les spécificités de chaque apprenant, à savoir son système de valeurs. Nous proposons alors un moteur de génération adaptatif qui prend en compte à la fois ces contraintes pédagogiques, ainsi que le profil moral de l'apprenant.

Dans ce chapitre, nous décrivons tout d'abord dans la section 3.1 l'architecture dans laquelle s'intègre le moteur de génération. Ensuite, dans la section 3.2, nous présentons l'apport de notre moteur de génération de dilemmes KOBA par rapport aux systèmes de l'état de l'art, en détaillant notre formalisation ainsi que les différentes étapes de génération. Finalement, dans la section 3.3, nous formalisons la notion d'ambiguïté et détaillons les différentes étapes de génération.

3.1 Architecture

Avant de décrire notre moteur de génération, il convient tout d'abord de présenter l'architecture globale dans laquelle il s'intègre. Dans l'introduction de ce mémoire, nous avons présenté brièvement le contexte technologique de nos travaux. Nous avons précisé que notre système s'intègre dans la plateforme HUMANS (Lourdeaux et al., 2017), et qu'il interagit avec différents modules externes gérés par nos partenaires dans le projet. Dans cette section, nous présentons un aperçu globale de l'architecture (cf. figure 3.1), et nous décrivons les interactions entre les modules.

3.1.1 Modules externes

Dans une session d'entraînement, l'apprenant évolue dans un environnement virtuel. Ce dernier consiste en un outil informatique qui immerge l'apprenant dans un monde virtuel où il peut se déplacer et interagir en utilisant les dispositifs standards (clavier-souris) ou les dispositifs de réalité virtuelle (visio-casque et manettes). Dans notre architecture conceptuelle,

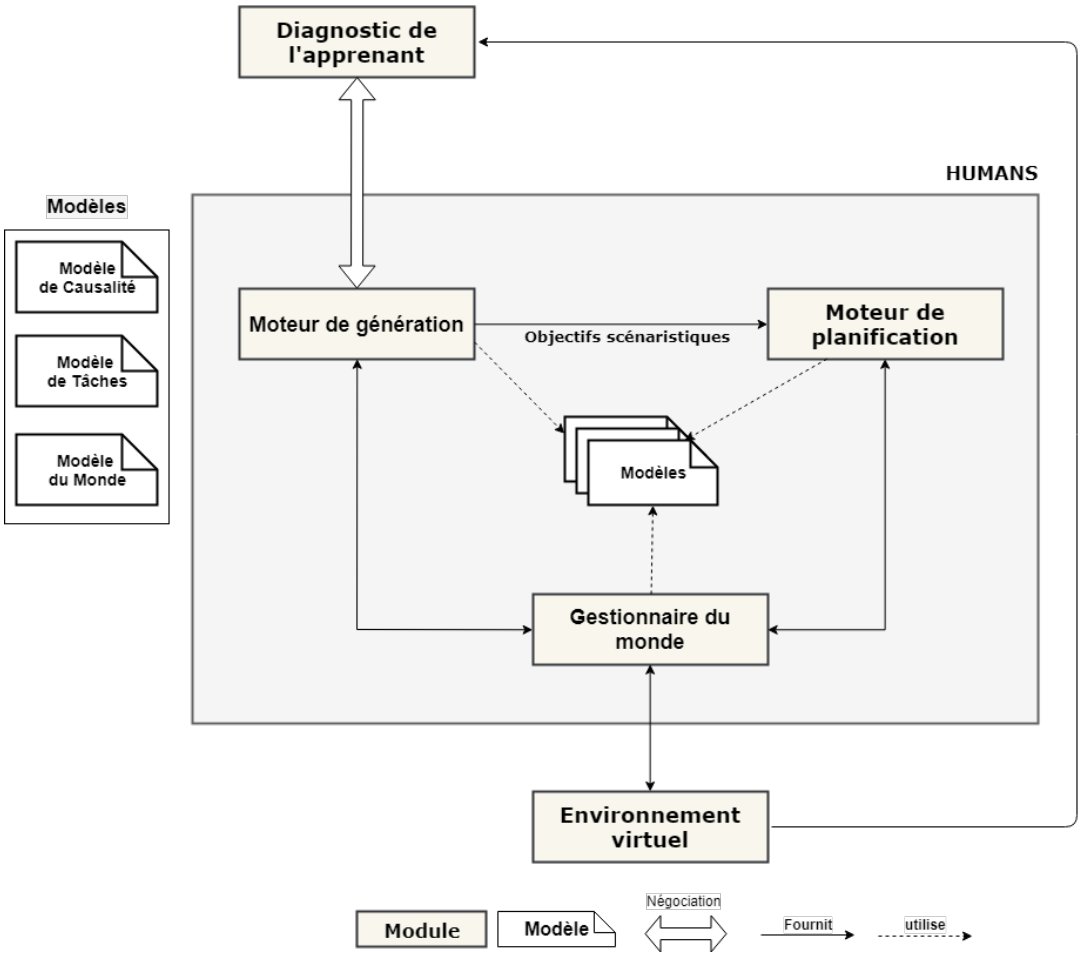

FIGURE 3.1 Architecture globale

les environnements virtuels sont considérés comme des modules [1] externes indépendants des modules de scénarisation. Cette modularité permet de séparer le raisonnement scénaristique lié à la sélection et la génération des situations, du raisonnement purement graphique et visuel lié aux animations et comportements des entités 3D. Dans cet environnement, l'apprenant jouit d'une liberté d'action qui lui offre une certaine autonomie et lui permet de réagir face aux situations auxquelles il est confronté. Ces actions et réactions sont observées par le module de `diagnostic de l'apprenant` qui prend en compte ses comportements pour établir un diagnostic. Ce dernier permet d'identifier les compétences techniques et/ou non-techniques défaillantes chez l'apprenant, et par conséquent, de construire une consigne pédagogique, à envoyer au système de scénarisation, pour générer la prochaine situation désirée. Cette consigne est constituée de trois éléments :

1. Nous définissons un module comme étant un ensemble de composants (e.g. moteurs, modèles, base de données) formant une entité indépendante, dont l'objectif est la réalisation d'une tâche spécifique bien définie.

1. une intention pédagogique (e.g. vérifier, renforcer ou déstabiliser une compétence),
2. une ou des compétence(s) technique(s) et/ou non-technique(s) cible(s),
3. et un degré (ou dimension) de criticité.

Les situations à proposer sont guidées par les intentions pédagogiques. En effet, une situation sélectionnée pour vérifier l'état de connaissance de l'apprenant est différente de celle qui est sélectionnée pour déstabiliser l'apprenant (Luengo, 2009). Dans le premier cas, nous cherchons plutôt à conforter les connaissances de l'apprenant pour informer le système sur son état de connaissance courant. Tandis que dans le deuxième cas, nous cherchons plutôt à déséquilibrer l'apprenant pour lui signaler une utilisation invalide des connaissances. Les deux éléments indispensables de la consigne sont l'intention pédagogique et les compétences à cibler. Le degré de criticité quant à lui est optionnel, d'autant plus qu'il pourrait être déduit de l'intention pédagogique. En effet, si l'objectif est de *renforcer* une compétence cible, la complexité de la situation à produire ainsi que la criticité des enjeux devraient être plus importantes que lorsqu'il s'agit simplement de *vérifier* une compétence. De même, si l'objectif est de *déstabiliser* une compétence, on peut penser que des dimensions critiques particulières seraient plus pertinentes que d'autres pour réaliser cette intention. Les situations ambiguës, par exemple, seraient un bon moyen de déstabilisation.

3.1.2 Modules internes

Cette thèse s'inscrit dans la continuité des travaux liés à la plateforme HUMANS. Cette plateforme est constituée d'un ensemble de modules décisionnels. Parmi ces modules, nous retrouvons TAILOR. Son objectif est d'établir le diagnostic de l'apprenant, d'une part, et de spécifier les objectifs scénaristiques d'autre part. Le coeur de nos travaux se situe au niveau de ce module, plus précisément, au niveau du `moteur de génération` qui est responsable de la génération des objectifs scénaristiques. Nous avons donc écarté la partie responsable du diagnostic, qui est redondante avec le module de diagnostic géré par nos partenaires (Bourrier et al., 2018).

Le moteur de génération reçoit la consigne pédagogique établie par le module de diagnostic telle que nous l'avons décrite plus haut. En fonction de cette consigne, le moteur recherche les situations qui répondent au mieux aux exigences du diagnostic, à savoir les compétences cibles, et les dimensions critiques à mettre en place (e.g "vérifier" la compétence "prise de décision" dans une "situation dilemmatique"). C'est ce moteur en particulier qui est responsable de la génération des dilemmes (KOBA), des situations ambiguës et des situations caractérisées par les autres dimensions de criticité. Pour cela, il utilise des modèles de connaissances expertes qui sous-tendent la simulation. Ces modèles permettent au moteur

d'extraire les propriétés nécessaires à l'émergence des situations critiques, en particulier les dilemmes et les situations ambiguës. A partir de cela, le moteur construit des objectifs scénaristiques qui sont transmis au `moteur de planification`.

Objectif scénaristique

Un objectif scénaristique est une unité de composition d'un scénario prescrit, il décrit un ensemble de contraintes que le scénario effectif doit respecter. (Carpentier, 2015)

Nous distinguons deux types d'objectifs scénaristiques : les objectifs implicites et les objectifs explicites. Les premiers spécifient comment le scénario doit se dérouler sans pour autant expliciter les comportements attendus. Il peut s'agir notamment d'un ensemble de propriétés que le scénario doit respecter (e.g. la gravité ou la complexité). Les objectifs explicites, quant à eux, décrivent ce qu'il doit exactement se passer, permettant ainsi un contrôle plus fin du scénario. Ils peuvent être concrets et reposer sur des instances (e.g. faire traverser le petit garçon qui joue sur le trottoir), comme ils peuvent être abstraits et reposer sur des concepts (e.g. faire traverser un piéton).

Précisons que dans le cadre de nos travaux, nous nous intéressons à la génération d'objectifs scénaristiques qui opèrent au niveau intradiégétique. Ce niveau se réfère à ce qui est intrinsèque au scénario (e.g. le passager qui met en garde le conducteur qu'il est en excès de vitesse). Le niveau extradiégétique, quant à lui, se réfère à ce qui peut être représenté grâce aux environnements virtuels mais qui ne fait pas partie du scénario (e.g. mettre en surbrillance un panneau de limite de vitesse pour mettre en garde le conducteur).

Ces objectifs scénaristiques sont transmis ensuite à un autre module de la plateforme : DIRECTOR. Ce dernier comporte en particulier un `moteur de planification` dont le but est de diriger la simulation vers une situation qui respecte ces objectifs tout en préservant la cohérence du scénario. Ce moteur à fait l'objet des travaux de thèse de (Barot, 2014), et fait actuellement l'objet des travaux de (Lacaze-Labadie et al., 2017). Les deux moteurs que nous venons de décrire, ne communiquent pas directement avec l'environnement virtuel. En effet, les décisions prises au niveau de ces deux modules se répercutent sur un modèle du monde qui est géré par le `gestionnaire du monde`. Ce gestionnaire constitue une interface entre le modèle du monde et l'ensemble des autres modules. Il permet de l'interroger (e.g. quel est le nombre de voitures présentes dans un rayon de 100m) et de le mettre à jour (e.g. changer la couleur du feu de circulation). Ce gestionnaire se charge également de notifier tous les modules concernés, dont l'environnement virtuel, de chaque changement d'état. Par conséquent, lorsque des changements sont opérés dans le modèle du modèle suite à des décisions du moteur de génération et/ou du moteur de planification, elles se répercutent systématiquement sur l'environnement virtuel qui est notifié par le gestionnaire du monde.

La plateforme HUMANS est composée également de deux autres modules : MONITOR et REPLICANTS. Ces deux modules sont respectivement responsables du suivi de l'apprenant et du comportement des personnages virtuels autonomes. Le premier étant redondant avec le module du diagnostic géré par nos partenaires, et les comportements des personnages virtuels autonomes n'étant pas pris en compte dans le cadre de nos travaux, ces deux modules ont été écartés de l'architecture. L'absence de ces deux modules n'affecte aucunement le fonctionnement des autres modules. En effet, la modularité de la plateforme permet à chaque module de fonctionner indépendamment des autres. Cela permet, par ailleurs, d'intégrer des modules supplémentaires en toute facilité.

Afin de mieux comprendre la tâche de chaque composant de cette architecture, nous détaillons ci-après un exemple simplifié en conduite automobile. L'apprenant est au volant d'une voiture dans une ville virtuelle. Après quelques scénarios initiaux, le `module de diagnostic` s'aperçoit qu'il ne dispose d'aucune information sur la capacité de l'apprenant à anticiper la survenue d'événements imprévisibles. Il décide alors de tester l'apprenant sur la compétence non-technique "conscience de la situation". Il établit, par conséquent, la consigne pédagogique suivante :

— intention pédagogique : vérifier

— compétence cible : conscience de la situation

Cette consigne pédagogique est reçue par le `moteur de génération` qui utilise les modèles à disposition afin d'identifier les tâches qui mettent en jeu cette compétence, ainsi que les conditions dans lesquelles elles sont réalisables. Il identifie la tâche "Gérer obstacle imprévu sur la route" comme répondant à cette consigne. A partir de cela, il établit un objectif scénaristique. Ici, la formulation d'un objectif implicite ne permet pas de cibler précisément la compétence désirée. Le moteur spécifie alors un objectif explicite abstrait (e.g. "traversée soudaine d'un piéton occulté"). Bien évidemment cette objectif aurait pu être concret (e.g. "traversée soudaine de la jeune fille qui joue sur le trottoir et qui est occultée par le camion garé sur le côté"). Toutefois, cela aurait contraint très tôt la génération et aurait affecté négativement la variabilité des situations. Cet objectif scénaristique est ensuite envoyé au `moteur de planification` qui utilise les modèles à disposition afin de diriger la simulation vers un état qui répond à cet objectif. Cet état est envoyé au `gestionnaire du monde` qui se charge de mettre à jour le modèle du monde et de notifier l'environnement virtuel qu'il y a eu un changement d'état. A la réception de cette notification, l'environnement virtuel identifie le changement à opérer (traversée d'un piéton occulté) et l'instancie visuellement.

Après avoir situé le moteur responsable de la génération des situations critiques dans l'architecture globale, nous décrivons dans la section suivante, les modèles sémantiques qu'il utilise afin de permettre l'émergence des dilemmes et des situations ambiguës.

3.1.3 Modèles sémantiques pour la scénarisation

La conception d'environnements virtuels pour l'entraînement ou la formation n'est pas du ressort des informaticiens seulement. C'est une tâche collective qui mobilise différentes personnes, à savoir des formateurs, des ergonomes, des experts, etc, qui ne sont pas nécessairement informaticiens. Par conséquent, chaque type de connaissance devrait idéalement être renseigné séparément dans un modèle adapté. Cette modularité facilite en effet la conception collective de la simulation, et permet aux experts de manipuler des outils et des modèles adaptés à leur domaine d'expertise. Par ailleurs, cela facilite la réutilisation de chaque type de connaissances dans des contextes différents. Par exemple, nous pouvons réutiliser le même monde virtuel pour des activités différentes. Dans ce cas, les seules connaissances à modifier seraient celles qui se rapportent à l'activité. Si ces connaissances n'étaient pas dissociées de celles du monde, il aurait fallu modifier, dans ce cas-là, tout le modèle. Que ces modèles experts soient des vues d'un modèle central, ou qu'ils soient des modèles à part entière, ils doivent respecter un certain nombre de critères afin de faciliter la production des environnements virtuels pour l'entraînement ou la formation. D'une part, il est nécessaire de pouvoir représenter des relations complexes entre les entités, et de raisonner à différents niveaux d'abstraction. Par conséquent, ces modèles doivent être dotés d'un grand pouvoir **expressif**. D'autre part, ces modèles sont destinés à être utilisés par des experts non-informaticiens, ils doivent donc être **intelligibles**. Enfin, les systèmes informatiques utilisent ces modèles pour raisonner et inférer de nouvelles connaissances. Par conséquent, ils doivent être **interprétables**. Ci-après nous récapitulons et détaillons l'ensemble de ces critères :

— **l'expressivité** : se réfère à la quantité des choses que nous pouvons décrire à l'aide d'un modèle. Plus un modèle est expressif, plus nous pouvons exprimer de choses. D'une part, un modèle expressif doit être doté d'un grand **pouvoir descriptif** qui permet de renseigner, de manière générale, les entités, leur propriétés et les relations qu'elles entretiennent les unes avec les autres. D'autre part, il doit être doté d'un grand **pouvoir d'abstraction**. C'est-à-dire qu'il ne doit pas se restreindre à la description des instances, mais doit permettre également un raisonnement à différents niveaux d'abstraction. Cela permet en conséquence, en couplant le modèle avec un système informatique adéquat, une plus grande variabilité de situations avec un moindre effort d'écriture. Supposons par exemple que nous disposons des concepts "piétons", "ballons" et "véhicules". Supposons que pour des besoins d'apprentissage nous avons besoin de mettre un obstacle devant le véhicule de l'apprenant. Le raisonnement sur les instances impose de spécifier précisément quelle instance mettre devant le véhicule. Si nous souhaitons être exhaustifs, il faut alors spécifier toutes les instances

de piétons, ballons et véhicules qui peuvent potentiellement servir d'obstacle. Cela représente un effort d'écriture considérable. Si nous décidons de nous restreindre à quelques instances pour alléger l'effort d'écriture, nous écartons, malgré nous, un ensemble d'instances qui auraient pu servir d'obstacle. Supposons maintenant que ces concepts héritent d'un concept abstrait : "Obstacle". Il suffit dans ce cas-là de spécifier que ce concept correspond à l'obstacle. Par conséquent, toutes les instances de ses descendants peuvent servir potentiellement d'obstacle. Par ailleurs, si le modèle évolue, et que de nouveaux concepts peuvent être utilisés à cet escient, il suffit de les faire hériter de "Obstacle", sans avoir à spécifier explicitement toutes les instances.

— **L'interprétabilité** : correspond à la capacité d'un modèle à être interprété par un système informatique afin de permettre des raisonnements dessus. Il doit donc respecter des règles syntaxiques qui permettent d'assurer l'interprétation de son contenu par un système informatique. Un modèle peut reposer sur un formalisme qui lui est propre en définissant ses propres règles, comme il peut reposer sur des formalismes existants, dont les plus répandus sont XML et JSON.

— **L'intelligibilité** : désigne la capacité d'un modèle à être manipulé par les experts sans la participation des informaticiens. Un modèle intelligible doit permettre à un expert de lire et écrire du contenu sans se soucier des spécificités techniques qui sont plus du ressort du programmeur.

Afin de permettre l'émergence des situations dilemmatiques et ambiguës, le moteur de génération a besoin de raisonner sur trois types de connaissances en particulier : les connaissances sur le monde, les connaissances sur l'activité et les connaissances sur la causalité. Les **connaissances sur le monde** se réfèrent aux descriptions des entités (e.g. objets, agents), leurs propriétés ainsi que les relations qu'elles entretiennent les unes avec les autres. Les **connaissances sur l'activité** se réfèrent à la représentation de l'activité telle qu'elle a été observée sur le terrain. Enfin, les **connaissances sur la causalité** renseignent sur les liens de causalité pertinents entre les différents événements du monde. Ces trois types de connaissances peuvent être complétés par des **connaissances pédagogiques** et/ou **didactiques**. Celles-ci sont principalement utilisées par le module de diagnostic de l'apprenant qui est géré par nos partenaires dans le projet.

Le modèle du monde

Lorsque nous nous intéressons à la description du monde, nous trouvons dans la littérature deux modes de représentation : les connaissances internalisées et les connaissances externalisées. La première approche consiste à porter les connaissances directement dans les objets du monde, tandis que la deuxième les centralise dans une représentation globale. S'agissant

de connaissances internalisées, nous retrouvons dans la littérature des systèmes comme les SMART OBJECTS (Kallmann and Thalmann, 1999a,b), STARFISH (Badawi and Donikian, 2004) et STORM (Mollet, 2005). Toutefois, le caractère très opérationnel de ces approches nuit fortement à l'expressivité. En particulier, lorsqu'il s'agit de décrire les relations entre les objets. Pour cela, nous privilégions les approches qui externalisent les connaissances. Nous retrouvons des systèmes tels que VEHA (Chevaillier et al., 2009), SYSML (OMG, 2014), VR-WISE (Bille et al., 2004) et COLOMBO (Edward, 2011; Edward et al., 2010). Toutefois ces systèmes concilient difficilement les critères que nous avons définis plus haut. Dans COLOMBO, par exemple, les possibilités d'abstraction restent très limitées. Ce qui affecte l'expressivité du modèle. VEHA quant à lui, repose sur une représentation UML. L'inaccessibilité à la sémantique de la structure ainsi qu'aux règles d'évolutions constituent un frein en termes d'interprétabilité. Un état de l'art plus détaillé sur ces modèles est présenté dans la thèse de (Carpentier, 2015). Nous privilégions alors la représentation ontologique de (Carpentier, 2015) qui définit le méta-modèle WORLD-DL en conciliant expressivité, intelligibilité et interprétabilité.

WORLD-DL repose sur une représentation ontologique couplée à des règles de fonctionnement. La représentation ontologique permet de décrire les connaissances statiques sur le monde à différents niveaux d'abstraction (e.g. objets, propriétés des objets). Elle permet également de définir les différentes relations entre les entités (e.g relations sémantiques). Les règles de fonctionnement, quant à elles, permettent de gérer l'évolution dynamique du monde provoquée par son fonctionnement interne ou par les actions de l'apprenant. Par ailleurs, l'utilisation d'une ontologie permet d'interroger la base du monde à l'aide de requêtes SPARQL. Par exemple, pour des besoins de génération, il serait pertinent de savoir combien de feux rouges se trouvent dans un rayon de 100m de l'apprenant. Cette information peut être utilisée, par exemple, pour calculer la probabilité d'instanciation d'une situation mettant en jeu un feu rouge. La figure 3.2 présente un fragment d'un modèle du monde. Celui-ci est composé de trois types d'entités : les agents, les objets concrets et les objets abstraits. Parmi les objets concrets, on trouve les voitures et la signalisation routière. Dans cet exemple, le monde est composé d'une seule instance qui est celle d'un feu de circulation.

D'un point de vue plus technique, les ontologies sont généralement représentées en langage RDF. L'utilisation des triplets RDF, sous la forme de (sujet prédicat valeur), permet en particulier d'exprimer facilement des assertions sur le monde. Par exemple, "la porte est fermée" peut être exprimée par le triplet : `?door :is-open false`. Par conséquent, il est possible de décrire un état du monde par un agrégat d'assertions.

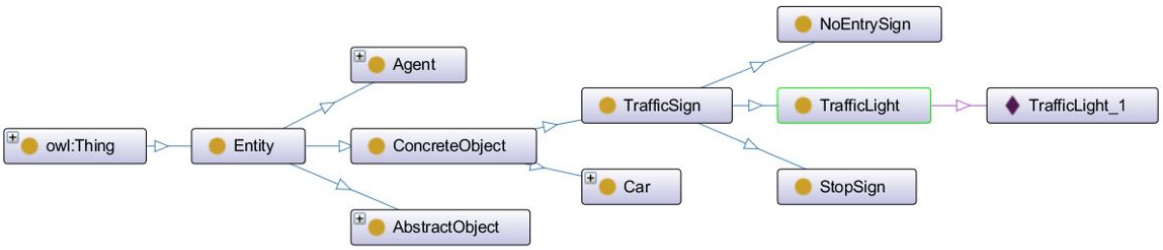

FIGURE 3.2 Fragment d'un modèle du monde

Le modèle de tâches

Le modèle de tâches, ou modèle d'activité, permet de représenter les séquences d'actions attendues de la part de l'apprenant. C'est aussi le modèle à partir duquel sont générés les comportements des personnages virtuels autonomes. Il s'agit alors de pouvoir représenter non seulement l'activité prescrite, mais aussi l'activité réelle telle qu'elle a été observée par les ergonomes chez les opérateurs sur le terrain. Par conséquent, outre les critères d'expressivité, d'interprétabilité et d'intelligibilité, il convient que ce modèle opérationnalise des principes issus de langages utilisés en ergonomie cognitive. Dans la littérature, nous trouvons un ensemble de modèles qui répondent partiellement à ces critères. Nous citons par exemple : LORA (Mollet and Arnaldi, 2006), YALTA (Rouillé et al., 2008) et HAWAI-DL (Amokrane, 2010; Edward, 2011). Le point fort des modèles issus de l'ergonomie cognitive est le fait qu'ils soient expressifs et intelligibles. Cependant, ils sont difficilement interprétables. A l'opposé, nous trouvons des modèles informatiques au caractère très opérationnel qui limite, à un certain degré, leur expressivité et leur intelligibilité. Les travaux de (Barot, 2014) ont permis de proposer le méta-modèle ACTIVITY-DL qui concilie ces différents critères.

Dans ce modèle, les tâches sont organisées sous forme hiérarchique. Cette représentation permet de modéliser la manière dont les opérateurs sur le terrain se représentent la tâche au niveau cognitif. Une tâche (but) peut donc avoir une ou plusieurs sous-tâches (sous-but) sur plusieurs niveaux. Le dernier niveau de la hiérarchie comprend les tâches feuilles. Elles correspondent à des actions élémentaires référencées dans le modèle du monde. Afin de renseigner les relations temporelles entre les tâches, le méta-modèle définit des constructeurs qui permettent de spécifier l'ordonnancement des tâches. ACTIVITY-DL dispose pour cela d'un ensemble de constructeurs prédéfinis, dont les plus courants sont listés dans le tableau 3.1. Par ailleurs, chaque tâche dispose de **préconditions** et de **postconditions**. Ce sont des agrégats d'assertions sur le monde formulés à l'aide de triplets RDF. Cette formulation est d'autant plus pertinente que le modèle du monde repose sur une représentation ontologique. Les préconditions se réfèrent aux conditions à examiner et à réaliser avant qu'une tâche ne puisse être effectuée. Le modèle distingue quatre types de préconditions : nomologiques,

Constructeur	Intitulé	Description
IND	Indépendant	Tâches ne partageant ni ressources, ni cibles, et dont les exécutions n'influent pas les unes sur les autres. Elles n'ont donc aucune contrainte spécifique d'ordonnancement.
SEQ	Séquentiel	Tâches n'ayant pas d'intervalle de temps commun, s'exécutant les unes à la suite des autres dans un ordre déterminé.
PAR	Parallèle	Tâches étant exécutées avec au moins un intervalle de temps commun.

TABLE 3.1 Ensemble des constructeurs disponibles dans ACTIVITY-DL

réglementaires, contextuelles et favorables (cf. tableau 3.2). Cette distinction est intéressante, car elle permet d'isoler les préconditions pertinentes pour chaque raisonnement. Par exemple, le moteur de génération souhaite savoir s'il peut générer une situation mettant en jeu deux tâches sélectionnées. Pour cela, le moteur doit vérifier la compatibilité des préconditions qui rendent les tâches physiquement réalisables. Si ces préconditions sont incompatibles, le moteur sait qu'il ne peut exister une situation qui les met en jeu. Maintenant, si aucune distinction n'était établie entre les préconditions, le moteur aurait pu juger, à tort, qu'aucune situation ne pouvait mettre en jeu ces deux tâches. Or, il se pourrait que l'incompatibilité des préconditions des deux tâches ne corresponde à aucune impossibilité physique. En plus des préconditions, les tâches du modèle disposent de postconditions, appelées aussi conditions de satisfaction. Elle se réfèrent aux conditions que doit vérifier l'état du monde afin que la tâche soit considérée comme réalisée. Elle peuvent également correspondre à des assertions sur l'activité en spécifiant, par exemple, qu'une tâche mère est considérée comme réalisée si toutes ses tâches filles le sont. Par ailleurs, le modèle permet d'étiqueter les tâches par des **marqueurs** qui permettent de renseigner toute information pertinente spécifique à la tâche. Ces marqueurs peuvent être binaires ou valués sous la forme (propriété : valeur). Ils constituent un moyen pratique pour spécifier, entre autres, quelles compétences techniques et/ou non-techniques sont mobilisées durant la réalisation d'une tâche donnée. Le méta-modèle complet de ACTIVITY-DL est présenté dans l'annexe B.

Pour mieux comprendre les éléments que nous venons de décrire, nous présentons un exemple dans le tableau 3.3. Cette exemple décrit une tâche mère "Respecter un feu rouge" composée de deux tâches filles : "Ralentir" et "Freiner". Ces deux tâches filles doivent être réalisées séquentiellement et dans un ordre bien spécifique (`SEQ-ORD`). Pour que la réalisation de la tâche soit pertinente, il faut que le feu de circulation soit au rouge (précondition contextuelle). Par ailleurs, pour que la tâche soit considérée comme réalisée, il faut que

le véhicule soit arrêté (condition de satisfaction). Finalement, la réalisation de cette tâche permet d'évaluer les compétences techniques de l'apprenant d'une part (marqueur booléen), et permet d'attribuer un score de 5 en termes de conformité aux règles d'autre part (marqueur valué).

Catégorie	Précondition	Description
Conditions à réaliser	Nomologique	Les préconditions nomologiques correspondent à l'état dans lequel le monde doit se trouver pour que la tâche soit physiquement réalisable. Elles incluent les préconditions des règles d'exécution de l'action liée à la tâche, définies dans le modèle du monde.
		Exemple : Pour réaliser la tâche "Ouvrir la porte", il faut que la porte soit fermée.
	Réglementaire	Les préconditions réglementaires correspondent aux états du monde nécessaires à la bonne réalisation de la tâche selon la procédure.
		Exemple : Pour réaliser la tâche "Tourner à droite", il faut mettre le clignotant.
Conditions à examiner	Contextuelle	Les préconditions contextuelles correspondent à l'état du monde dans lequel il est pertinent de réaliser la tâche. Lorsque ces conditions ne sont pas vérifiées, la tâche peut être ignorée.
		Exemple : *La réalisation de la tâche "Changer de pneu" n'est pertinente que si le pneu est crevé.*
	Favorable	Les préconditions favorables correspondent aux états du monde dans lesquelles la réalisation de la tâche sera privilégiée par rapport à d'autres tâches. Elles permettent de choisir entre plusieurs tâches lorsqu'il existe plusieurs alternatives permettant de réaliser une même tâche mère.
		Exemple : *S'il y a des bouchons dans le trajet, la réalisation de la tâche "Prendre le train" est à privilégier.*

TABLE 3.2 Préconditions définies dans ACTIVITY-DL (tableau adapté de la source : Carpentier (2015))

Le modèle de causalité

Un événement peut être déclenché par un ou plusieurs autres événements. Par exemple, un "accident" peut être causé par la présence d'un "piéton sur la route" et d'un "véhicule roulant à grande vitesse". Un événement peut éventuellement être inhibé par d'autres événements, ou des actions réalisées par l'apprenant. Si on reprend l'exemple précédent, l'accident peut être

Task		`Respecter un feu rouge`
Subtasks		`Ralentir` `Freiner`
Constructor		`SEQ-ORD`
Conditions	**Contextual**	`(?light :has-color :red)`
	Satisfaction	`(?vehicle :is-stopped true )`
Marqueurs		`Conformité aux règles : 5` `Compétence technique`

TABLE 3.3 Exemple de tâche modélisée avec ACTIVITY-DL

évité si le conducteur effectue un "freinage d'urgence". Par ailleurs, les événements peuvent correspondre à un type particulier. Par exemple, une "infraction du code de la route" peut se référer, entre autres, à une "infraction vitesse" ou une "infraction stationnement/circulation". Afin de produire des comportements représentatifs de ceux observés et qui tiennent compte de ces particularités, il est nécessaire de se munir d'un modèle qui permet de modéliser l'ensemble de ces connaissances. Bien qu'il soit possible de porter ces connaissances dans le modèle du monde, nous pensons que pour assurer la modularité des modèles, il est préférable de disposer, à minima, d'une vue qui permet aux experts concernés de renseigner ce type spécifique de connaissances. Ces experts peuvent être des formateurs ou dans certains cas, des experts de l'analyse de risques, comme c'est le cas pour l'entraînement à la gestion des situations critiques. Il serait alors intéressant dans ce cas-là de disposer d'un modèle qui s'inspire des formalismes utilisés en analyse de risques, qui permettent de représenter les résultats des analyses effectuées sur le terrain. Par ailleurs, à l'instar des modèles précédents, il est également nécessaire que ce modèle réponde aux trois critères définis dans l'objectif de faciliter la production d'environnements virtuels pour l'entraînement ou la formation, à savoir : l'expressivité, l'interprétabilité et l'intelligibilité.

Dans le cadre de nos travaux, nous retenons le méta-modèle CAUSALITY-DL (Barot, 2014), en premier lieu parce que d'un point de vue pratique, il s'intègre parfaitement avec le modèle du monde et le modèle d'activité retenus et définis plus haut, et surtout, parce qu'il répond aux trois critères que nous avons définis. Par ailleurs, CAUSALITY-DL s'inspire des formalismes utilisés en analyse de risques, notamment des nœuds papillons (Debray et al., 2006). Ces derniers permettent de représenter, d'un côté, un arbre de défaillances qui correspond aux causes d'un événement redouté central, et d'un autre côté, un arbre d'événements qui correspond aux conséquences du déclenchement de l'événement redouté central. Cette inspiration rend alors le modèle facilement manipulable par les experts en analyse de risques. Pour ces raisons, nous pensons que l'utilisation de ce méta-modèle, que nous détaillons ci-après, est pertinente dans le cadre de nos travaux.

Les modèles représentés par CAUSALITY-DL sont des graphes orientés acycliques dont les nœuds se réfèrent aux **événements** du monde. Les événements, qui constituent l'élément de base du modèle, sont de deux types : les événements ponctuels (e.g. explosion) et les événements duratifs (e.g. pluie). A l'instar des tâches, ils peuvent être annotés par toute sorte d'information pertinente sous la forme `propriété:valeur`. Nous pouvons, par exemple, spécifier explicitement grâce aux **marqueurs** qu'un événement correspond à une conséquence négative, ou renseigner les propriétés qui permettent de le déduire implicitement (e.g gravité : 2). Les événements sont liés les uns avec les autres par des arcs qui correspondent aux liens de causalité ou de subsomption. Les **liens de causalité** permettent de déterminer quelles combinaisons d'événements peuvent causer un événement "but". Au niveau de ces liens, il est possible d'exprimer des conjonctions et des disjonctions grâce à des **portes logiques** `ET/OU`. Grâce à cela, un moteur de génération peut déterminer, par exemple, si un événement donné conduit à une conséquence négative. Quant aux **liens de subsomption**, ils permettent de renseigner si un événement subsume un autre (cf. exemple d'infraction cité plus haut). Cette capacité d'abstraction permet d'éviter la spécification exhaustive des séquences d'événements de bas niveau. Par ailleurs, chaque événement peut être associé à un ensemble de **barrières**. Ces dernières correspondent aux événements ou actions qui inhibent le déclenchement d'un événement donné. CAUSALITY-DL définit deux types de barrières. Lorsque ces barrières interviennent en amont d'un événement, elles sont dites de prévention, car elles empêchent son déclenchement. En revanche, lorsqu'elles sont présentes après le déclenchement d'un événement, elles sont dites de protection, car elles empêchent la propagation de ses conséquences. Cette notion de barrière est très intéressante, car elle permet au moteur de génération, d'une part, d'identifier les actions dont la **non-réalisation** entraîne le déclenchement des événements qui lui sont postérieurs, et d'autre part, d'identifier les potentielles "menaces" (actions pouvant être réalisées par l'apprenant) qui peuvent mettre en péril le scénario en barrant la route du chemin menant vers l'événement désiré. Toutefois, dans notre cas, la distinction entre les barrières de prévention et de protection n'est pas pertinente, car elles sont traitées identiquement par le moteur de génération. Dans nos travaux, nous supposons que le raisonnement sur les barrières est binaire. C'est-à-dire que l'activation (non-activation) d'une barrière empêche (entraîne) forcement le déclenchement des événements postérieurs. En vérité, dans CAUSALITY-DL, le raisonnement est plutôt probabiliste. Autrement dit, le fait qu'une barrière ne soit pas activée, ne déclenche pas systématiquement ses conséquences. Par exemple, le fait de ne pas mettre la ceinture de sécurité (barrière) n'implique pas forcement des blessures en cas d'accident (conséquence). Finalement, les événements sont parfois confondus avec les actions. C'est le cas lorsqu'ils sont déclenchés systématiquement après la réalisation de l'action associée (e.g action : "Démarrer voiture" et

événement : "Démarrage de voiture"). Dans le modèle, nous considérons ces nœuds comme étant des nœuds action. Le méta-modèle complet CAUSALITY-DL est présenté dans l'annexe B.

Afin de faciliter la compréhension de ces éléments, nous présentons dans la figure 3.3, un exemple d'un fragment d'un modèle de causalité. Dans cet exemple, une "Violation du code de la route" est causée soit par l'événement "Griller un feu rouge", soit par l'événement "Griller un stop". Ces deux événements risquent de se produire, respectivement, lorsque l'apprenant "s'approche du feu rouge" ou "s'approche du stop". Pour les inhiber, l'apprenant peut activer les barrières correspondantes, à savoir, "S'arrêter" au feu rouge ou "Marquer un temps d'arrêt" au stop.

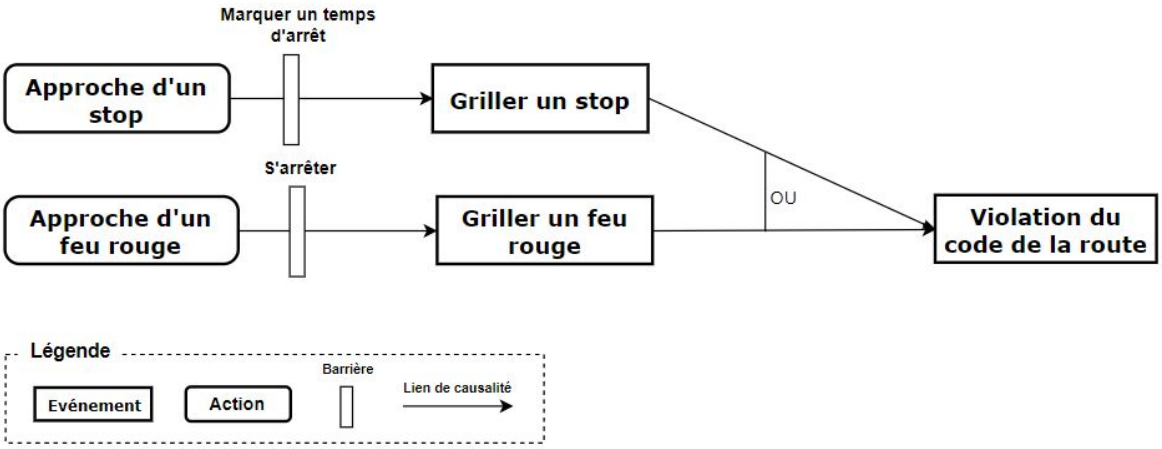

FIGURE 3.3 Fragment d'un modèle de causalité

3.1.4 Bilan

Afin de permettre l'émergence des situations dilemmatiques et ambiguës, le moteur de génération a besoin de raisonner sur trois types de connaissances : les connaissances sur le monde, les connaissances sur l'activité et les connaissances sur la causalité. Pour représenter ces connaissances nous avons choisi trois modèles qui sont à la fois expressifs, interprétables et intelligibles.

Le modèle du monde permet de représenter les entités du monde et les relations qu'ils entretiennent les unes avec les autres à différents niveau d'abstraction. Cela permet au moteur de génération de raisonner à haut niveau et de spécifier ainsi des objectifs scénaristiques concrets abstraits sans avoir à expliciter tous les états désirés. Par ailleurs, la représentation ontologique du monde facilite son interrogation par le moteur de génération.

Le modèle de tâches permet de décrire l'activité telle que les opérateurs se la représentent au niveau cognitif. Les préconditions et les postonditions de tâches permettent au moteur de génération de déduire les relations sémantiques entre les tâches. Par ailleurs, la distinction entre différents types de préconditions permet au moteur de génération de raisonner sur un type spécifique selon les besoins de génération. Cela permet d'éviter, par exemple, d'écarter à tort des situations qui auraient pu être pertinentes. De plus, les constructeurs temporels permettent au moteur de génération de déterminer si la réalisation d'une tâche dépend d'une autre, auquel cas, une situation les mettant toutes les deux en jeu serait impossible. Enfin, les marqueurs sont un moyen pratique pour identifier les tâches qui répondent à certaines propriétés spécifiques.

Le modèle de causalité permet de représenter les chaînes de causalité pertinentes entre les événements. Cela permet au moteur de génération d'identifier les événements qui conduisent à des conséquences négatives. La notion de barrière, inspirée par les formalismes d'analyse de risques, est intéressante car elle permet de déterminer les actions dont la non-réalisation conduit à des conséquences négatives. Aussi, elle permet au moteur de génération de détecter toutes les actions qui pourraient mettre en échec le scénario désiré.

Enfin, nous tenons à préciser que ces modèles sont beaucoup plus riches. Dans cette section, nous avons fait le choix de ne décrire que les propriétés qui nous intéressent. En d'autres termes, les propriétés qui sont utilisées par le moteur pour la génération des dilemmes et des situations ambiguës. Aussi, il est tout à fait envisageable d'utiliser d'autres modèles de l'état de l'art tant qu'ils disposent de ces propriétés. Dans la section suivante, nous détaillons comment le moteur de génération utilise ces modèles pour générer dynamiquement les dilemmes.

3.2 KOBA : moteur de génération automatique de dilemmes

L'objectif de nos travaux est de proposer un moteur de génération de dilemmes qui soit tout d'abord expressif. C'est-à-dire, un moteur qui permet à la fois de décrire des dilemmes complexes, et de spécifier des objectifs scénaristiques abstraits qui permettent d'éviter d'avoir à spécifier explicitement les instances désirées pour l'émergence des situations dilemmatiques. Le moteur de génération doit également permettre l'émergence d'un large panel de situations. Pour cela il doit être capable de produire une variabilité de situations sans effort d'écriture considérable. Par ailleurs, l'émergence des dilemmes moraux dépend des valeurs morales propres à chaque apprenant. Le moteur de génération doit, par conséquent, prendre en compte le profil de l'apprenant, et proposer des situations adaptées qui vont mettre en conflit son système de valeurs. Finalement, le moteur de génération doit être exploitable dans n'importe

quel domaine. Par conséquent, il doit être générique. L'état de l'art sur les dilemmes, nous a permis d'identifier deux systèmes de génération dynamique : GADIN et EGAD. Ces systèmes concilient difficilement les quatre aspects mentionnés ci-dessus. Nous proposons alors KOBA : un moteur de génération générique, plus expressif, plus adaptatif, et qui offre une plus grande variabilité de situations.

Dans notre modélisation, nous distinguons les dilemmes d'obligation et les dilemmes de prohibition. Cette distinction n'est pas que sémantique, elle est justifiée par le fait que le processus de génération n'est pas le même. Cette distinction diminue également les chances de passer à côté de la génération de certains dilemmes en raison de la manière différente dont chaque auteur remplit les modèles. Pour la génération de ces deux types de dilemmes, le moteur utilise des modèles sémantiques expressifs qui permettent de déterminer non seulement les conséquences dues à la **réalisation** des actions, mais aussi les conséquences dues à leur **non-réalisation**. Par ailleurs, comparé à EGAD, nous proposons un moteur de génération qui permet une meilleure exploitation des représentations abstraites. Cela permet une plus grande variabilité de situations avec un moindre effort d'écriture. Par ailleurs, dans les systèmes de l'état de l'art, l'exclusivité du choix présenté au décideur est assurée au niveau extradiégétique grâce à un menu textuel qui n'autorise pas de choix multiples. Dans un environnement virtuel, cela impliquerait de mettre en pause la simulation. Cela romprait l'immersion de l'utilisateur. Par conséquent, nous proposons un moteur de génération qui assure l'exclusivité du choix au niveau intradiégétique. Pour cela, il utilise des modèles sémantiques afin d'identifier des actions contradictoires, c'est-à-dire des actions incompatibles qui ne peuvent pas être réalisées simultanément. Enfin, nous proposons d'utiliser la théorie des valeurs universelles de (Schwartz, 2006) pour modéliser le profil moral de l'apprenant. Ce profil est mis à jour dynamiquement sur la base des relations de compatibilité et d'antagonisme décrites par la théorie.

Dans la section précédente, nous avons décrit l'ensemble des modèles sémantiques utilisés dans le cadre de nos travaux. Ces modèles répondent à des critères que nous avons définis dans l'objectif de faciliter la production d'environnements virtuels pour l'apprentissage. Dans cette section, nous détaillons comment le moteur de génération utilise ces modèles pour spécifier des objectifs scénaristiques qui vont permettre l'émergence des dilemmes. Nous clarifions tout d'abord la notion de conséquence négative telle que nous l'adoptons dans le cadre de nos travaux. Ensuite, dans la section 3.2.2 nous présentons notre modélisation du profil moral de l'apprenant. Dans la section 3.2.3, nous exposons la formalisation des dilemmes de prohibition et des dilemmes d'obligation. Finalement, dans la section 3.2.4, nous détaillons les différentes étapes de leur génération.

3.2.1 Les conséquences négatives

Dans le cadre de nos travaux, les conséquences négatives ne se résument pas seulement aux dommages causés suite à la réalisation d'une action. Un dommage, rappelons le, désigne toute lésion ou atteinte à la santé de la personne. Nous étendons cette notion pour inclure également toute atteinte aux valeurs morales chéries par la personne (section 3.2.2) et, de manière générale, toute violation normative (règles, normes, ordre, etc). Par ailleurs, dans un cadre d'entraînement ou de formation, l'utilisateur d'une simulation peut avoir un score qui, selon les contextes, accepte différentes significations. Il peut s'agir, par exemple, d'un score de performance, du nombre de points de vie (de permis) restants ou du nombre de ressources disponibles. Dans ce cas, nous considérons qu'une perte de points, i.e. une diminution du score, correspond également à une conséquence négative. Nous distinguons alors des conséquences négatives en termes de **gravité**, **de perte de points** et de **violation normatives**.

Pour chaque type de conséquences négatives, nous définissons une fonction booléenne f_i d'évaluation de conséquences, qui, étant donné un événement, évalue si c'est une conséquence négative ou non. Cette fonction prend en entrée un événement. Elle renvoie 1 si c'est une conséquence négative et 0 sinon.

La gravité

La gravité correspond au degré du dommage causé suite à la réalisation d'une action (e.g. percuter un piéton). C'est une propriété numérique des événements qui varie entre 0 (absente) et 10 (catastrophique). Un événement est considéré comme étant une conséquence négative en termes de gravité, si la valeur de gravité associée est supérieure à 0. Nous définissons alors la fonction d'évaluation f_g de la manière suivante :

Fonction d'évaluation de conséquences (gravité)

$$f_g : E \to \{0,1\}$$
$$e \mapsto f_g(e)$$

$$f_g(e) = \begin{cases} 1 & \text{si } g_e \in [1,10] \\ 0 & \text{sinon} \end{cases}$$

avec E, l'ensemble des événements de l'environnement,
et $g_e (\in [0,10])$ la gravité associée à l'événement e.

Perte de points

Lors d'une simulation, le personnage joueur peut se voir attribuer un score. Cela peut correspondre, de manière très générale, à un score de performance qui augmente lorsque les tâches sont correctement réalisées ou, au contraire, qui diminue lorsqu'elles sont réalisées de manière incorrecte. Il peut également s'agir d'un score spécifique à un contexte particulier. En conduite automobile, par exemple, il peut correspondre au total des points de permis restants. Dans un contexte plus ludique, ce score peut correspondre aux points de vie restants au joueur ou à la quantité de ressources disponibles.

Les points sont des propriétés numériques des événements, ils peuvent avoir une valeur positive (gain de points), négative (perte de points) ou nulle (sans effet). Un événement est considéré comme étant une conséquence négative en termes de points si la valeur de cette propriété est négative. Nous définissons alors la fonction d'évaluation f_p de la manière suivante :

Fonction d'évaluation de conséquences (perte de point)

$$f_p : E \to \{0,1\}$$
$$e \mapsto f_p(e)$$

$$f_p(e) = \begin{cases} 1 & \text{si } p_e < 0 \\ 0 & \text{sinon} \end{cases}$$

avec E l'ensemble des événements de l'environnement,
et $p_e (\in \mathbb{Z})$ le nombre de points perdus ou gagnés associé à l'événement e.

Violations normatives

Chaque individu dispose d'une hiérarchie de valeurs qui lui est propre. Il en chérit certaines et en méprise d'autres. Dans certains cas, le déclenchement d'un événement entraîne la violation d'une ou de plusieurs valeurs morales. Par conséquent, un événement est considéré comme étant une conséquence négative en termes de valeurs si son déclenchement entraîne la violation d'une ou de plusieurs valeurs morales chéries par l'agent décideur. Nous définissons alors la fonction d'évaluation de conséquences f_v de la manière suivante :

Fonction d'évaluation de conséquences (violation normative)

$$f_n : E \to \{0,1\}$$
$$e \mapsto f_n(e)$$

$$f_n(e) = \begin{cases} 1 & \text{si } (V_e \cap V_{AD}) \neq \emptyset \\ 0 & \text{sinon} \end{cases}$$

avec E, l'ensemble des événements de l'environnement,
V_e, l'ensemble des valeurs morales violées par le déclenchement de l'événement e,
V_{AD}, l'ensemble des valeurs morales chéries par l'agent décideur.

Conclusion

Dans nos travaux, nous avons étendu la notion de conséquences négatives pour inclure au total trois types de conséquence. Pour chacun de ces types, nous avons défini une fonction d'évaluation qui permet de déterminer, à partir d'un événement donné en entrée, si celui-ci est considéré comme une conséquence négative de ce type. Ainsi, à partir de ces fonctions, nous définissons une fonction d'évaluation plus globale f_{cn}, qui permet de déterminer si un événement est considéré comme une conséquence négative.

Fonction d'évaluation de conséquences

$$f_{cn} : E \to \{0,1\}$$
$$e \mapsto f_{cn}(e)$$

$$f_{cn}(e) = f_g(e) \vee f_p(e) \vee f_n(e)$$

Nous intégrons également dans le modèle de causalité trois nœuds terminaux fictifs (GRAVITÉ, PERTE DE POINTS, VIOLATIONS NORMATIVES) qui correspondent chacun à un type de conséquences négatives (cf. figure 3.4). Ces nœuds fictifs ne sont ni des actions ni des événements. Ce ne sont d'ailleurs pas non plus des nœuds qui sont renseignés par les auteurs. Les liens entre ces nœuds et les événements sont générés automatiquement par le moteur de génération en fonction des propriétés des événements. Un lien est créé entre un événement e et un nœud terminal fictif si la fonction d'évaluation de conséquences négatives associée au nœud retourne 1 pour l'événement en question.

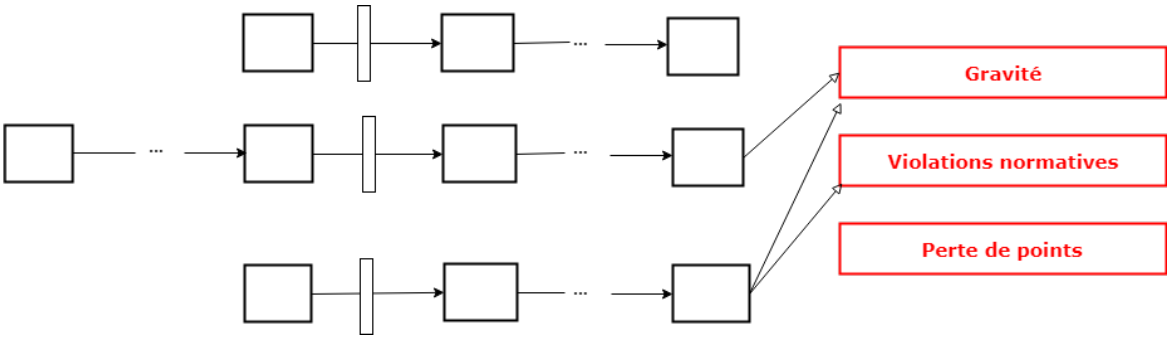

FIGURE 3.4 Noeuds fictifs de conséquences négatives

3.2.2 Le profil moral

Face à un choix, il arrive que dans certains cas, l'individu soit confronté à un choix qui met en jeu son système de valeurs. Il s'agit dans ce cas-là d'un dilemme moral. Dans ces situations, chaque individu agit selon l'importance qu'il accorde à chacune de ces valeurs. Par conséquent, il se peut que ce qui est dilemmatique pour une personne ne le soit pas forcément pour les autres. Afin de pouvoir générer des dilemmes moraux, il est alors nécessaire de prendre en compte le profil moral de l'apprenant. Pour cela, nous nous sommes appuyés sur des théories issues des sciences humaines afin de proposer une modélisation de ce profil. L'état de l'art sur les valeurs humaines, nous a permis de mettre en exergue quelques théories intéressantes. Dans le cadre de nos travaux, nous avons choisi d'adopter la théorie des valeurs universelles de (Schwartz, 2006) qui constitue l'une des théories les plus accomplies et les plus reprises dans la littérature. Dans cette théorie, l'auteur définit dix grandes catégories de valeurs morales universelles représentées dans une structure circulaire. Par conséquent, la théorie ne se contente pas d'établir une taxonomie de valeurs, elle renseigne également les relations de compatibilité et d'antagonisme qui les relient. Cet élément est très intéressant, car cela permet, entre autres, de propager des croyances sur l'importance accordée aux valeurs à partir de données pauvres. Dans la suite, nous détaillons notre conception du profil moral, et nous expliquons comment le système met à jour dynamiquement ce profil.

Les valeurs morales

Nous définissons le profil moral de l'apprenant comme étant constitué d'une liste d'`item-valeurs`. Chaque `item-valeur` est décrit par trois éléments :

— `valeur` : une chaîne de caractère correspondant au nom de la valeur de l'item (e.g. Hédonisme).

— `dimension` : une chaîne de caractère correspondant à la dimension de la valeur (e.g Affirmation de soi).

— `importance` : une valeur numérique comprise entre -2 (très méprisée) et +2 (très importante), qui correspond à l'importance accordée à cette valeur.

Dans la section précédente, nous avons vu que certains événements pouvaient conduire à des violations de valeurs morales. Toutefois, pour qu'ils soient considérés comme des conséquences négatives, il est nécessaire que les valeurs violées soient chéries par l'apprenant. Nous définissons alors à partir du profil moral de l'apprenant, la liste des valeurs V_{AD} qu'il chérit :

$$v \in V_{AD} \text{ ssi } \textsc{Importance}(v) \in \{1,2\}$$

Le renseignement de ce profil peut se faire a priori en se reposant sur un questionnaire en début de session d'entraînement. L'utilisation du questionnaire original et complet de (Schwartz et al., 2012) peut tout à fait être envisagé. Cela permet de renseigner l'ensemble des valeurs d'importance pour tous les item-valeurs.

Mise à jour du profil

Afin de combler le manque d'expressivité du système GADIN engendré par le fait de se retreindre aux relations sociales pour la génération des dilemmes, EGAD propose de s'appuyer sur la théorie des valeurs universelles de (Schwartz, 2006). Toutefois, le potentiel de cette théorie n'est pas complètement exploité par ce système. L'originalité de cette théorie réside dans sa structure circulaire qui renseigne sur les relations d'opposition et de compatibilité entre les valeurs. Elle oppose par ailleurs deux dimensions : La Continuité à l'Ouverture au changement, et le Dépassement de soi à l'Affirmation de soi. Nous proposons alors d'exploiter cette structure circulaire, en particulier les oppositions dimensionnelles afin de mettre à jour dynamiquement le profil moral de l'apprenant à partir des comportements observés. Initialement, les valeurs d'importance des item-valeurs sont définies à 0. Lorsqu'une situation met en jeu une valeur morale particulière, si l'apprenant entreprend une action conformément à (à l'encontre de) cette valeur, l'importance de celle-ci est incrémentée (décrémentée). Tenant compte des relations de compatibilité, l'importance des valeurs de la même dimension est également incrémentée (décrémentée). Par ailleurs, en prenant en considération les relations d'opposition, l'importance des valeurs de la dimension opposée est décrémentée (incrémentée).

Prenons, par exemple, le cas où l'apprenant est confronté à une situation où il peut griller un stop pour gagner du temps. Il choisit de ne pas le faire. Par conséquent, le système met à jour son profil moral en incrémentant l'importance relative à la valeur "Conformité". Le système incrémente également l'importance de toutes les valeurs de la dimension CONTINUITÉ, à savoir la "Tradition" et la "Sécurité". En revanche, l'importance des valeurs de la

dimension opposée (OUVERTURE AU CHANGEMENT) est décrémentée, à savoir "l'Autonomie", la "Stimulation" et "l'Hédonisme". Plus tard dans son trajet, l'apprenant s'aperçoit qu'un automobiliste garé au bord de la route a besoin d'aide pour démarrer sa voiture. Il s'arrête alors pour lui donner un coup de main. Le système met à jour le profil moral de l'apprenant en incrémentant l'importance de la "Bienveillance" ainsi que celle des valeurs de la dimension DÉPASSEMENT DE SOI. Par ailleurs, tenant compte des incompatibilités, il décrémente l'importance des valeurs de la dimension opposée à savoir le "Pouvoir", la "Stimulation" et "l'Hédonisme" (cf. figure 3.5).

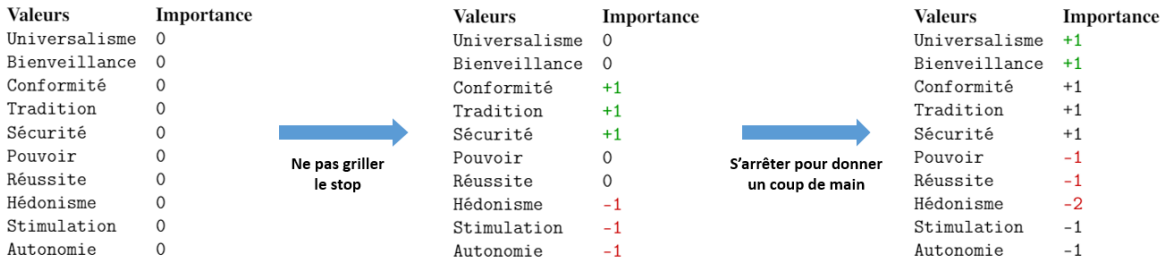

FIGURE 3.5 Exemple de mise à jour du profil moral suite à deux actions observées

3.2.3 Formalisation des types de dilemmes

L'approche que nous proposons consiste à générer automatiquement des dilemmes à partir de modèles de connaissances. Pour réaliser cet objectif, il est nécessaire de disposer a priori d'une formalisation d'une situation dilemmatique. Cela permet au moteur de génération d'identifier précisément quelles sont les propriétés d'une situation dilemmatique, et par conséquent, d'extraire les éléments qui permettent de créer des situations qui respectent ces propriétés. Dans l'état de l'art que nous avons établi, nous avons évoqué une distinction entre les dilemmes de prohibition et les dilemmes d'obligation. Dans cette section, nous proposons une formalisation pour chacun de ces types, en mettant en exergue les conditions dans lesquelles une situation est considérée comme étant un dilemme d'obligation ou un dilemme de prohibition.

Les dilemmes d'obligation

Rappelons qu'un dilemme d'obligation est une situation où toutes les actions possibles sont obligatoires. Pour ce type de dilemme, la condition que doit satisfaire la situation est la suivante : les actions ne peuvent pas toutes être réalisées. Autrement dit, le choix présenté à l'agent décideur doit être exclusif. En effet, si l'agent décideur peut réaliser toutes les actions,

sans retombées négatives, le dilemme est compromis. L'enjeu du moteur de génération, dans ce cas, est de prescrire un état du monde qui garantit cette condition. Il ne suffit donc pas de générer une situation mettant en jeu deux actions, il faut également s'assurer de l'impossibilité physique de réaliser ces deux actions simultanément. Pour assurer cela, les systèmes de l'état de l'art reposent sur l'interface graphique en proposant un menu textuel qui limite le choix à une action. Dans un contexte d'entraînement aux situations critiques en environnement virtuel, cette contrainte doit être assurée au niveau intradiégétique. C'est à dire que l'exclusivité du choix doit être assurée de manière intrinsèque à la situation. Pour ce faire, le moteur de génération raisonne sur la contradiction entre les actions. Deux actions sont contradictoires si elles sont incompatibles lorsqu'elles doivent être réalisées simultanément (e.g. "avancer" et "reculer", "augmenter" et "diminuer"). Lorsque l'apprenant est face à une alternative, si les actions de chaque issue possible sont contradictoires, cela permet de garantir le fait que l'agent décideur ne puisse pas choisir les deux issues. Ainsi, les conséquences de l'une d'entre elles sont forcement déclenchées. Il existe deux types d'incompatibilité entre les actions :

— **nomologique**, qui se réfère aux actions qui sont incompatibles par nature (e.g. "avancer" vs "reculer", "augmenter" vs "diminuer")
— **scénaristique/réglementaire**, qui se réfère aux actions qui sont nomologiquement compatibles mais qui, dans un contexte particulier, ne sont plus compatibles suite à des contraintes scénaristiques et/ou réglementaires (e.g. lois, normes, ordre, consigne, etc). Par exemple, les actions "parler au téléphone" et "conduire" ne sont pas opposées par nature. Elles sont nomologiquement compatibles. Cependant, dans un contexte particulier (le code de la route en France par exemple), les deux actions deviennent incompatibles.

S'agissant de la génération de situations à choix exclusif, seul le premier type d'incompatibilité peut être considéré. En effet, lorsque deux actions sont incompatibles du point de vue scénaristique ou réglementaire, rien n'empêche pour autant l'individu de les réaliser. Toutefois, si le choix n'est pas exclusif, il pourrait y avoir un dilemme si la réalisation des deux actions conduit également à des conséquences négatives. Par conséquent, l'agent décideur serait confronté ici, non pas à deux, mais à trois issues dont les conséquences sont négatives. Pour illustrer ce cas, prenons l'exemple suivant : un individu conduisant sa voiture, se dirige vers l'aéroport. Il est très en retard. Par conséquent, il ne peut absolument pas faire d'arrêt, sinon il risque de rater son avion. Il attend également un coup de fil de la part de son employeur auquel il doit absolument répondre s'il souhaite garder son poste. Pendant son trajet, il reçoit le coup de fil de la part de l'employeur. S'il ne s'arrête pas pour répondre, il perd son poste. S'il s'arrête pour répondre au téléphone, il rate son avion. Ici, l'individu est

face à deux issues menant chacune à des conséquences négatives. L'issue "échappatoire" serait que l'individu réponde au téléphone tout en conduisant. Cela est possible car les actions sont nomologiquement compatibles. Toutefois, cela représente une violation du code de la route. Cela met donc l'individu face à un choix à trois options, toutes menant à des conséquences négatives. A partir de ces éléments, nous formalisons un dilemme d'obligation comme suit.

Soit a_1 et a_2 deux actions qu'un agent décideur AD peut réaliser. Pour certaines raisons, AD doit réaliser a_1. En d'autres termes, ne pas faire a_1 conduit à des conséquences négatives CN_{a_1}. Pour d'autres raisons, AD doit réaliser a_2. Autrement dit, ne pas faire a_2 conduit à des conséquences négatives CN_{a_2}. La condition nécessaire dans un dilemme d'obligation est que l'agent ne peut pas faire les deux actions, ou sinon, que la réalisation des deux actions conduit à des conséquences négatives. Par conséquent, s'agissant d'une situation mettant en jeu deux actions, nous formalisons un dilemme d'obligation de la manière suivante :

Dilemme d'obligation (deux actions)

$S(A,CN)$ est une situation de dilemme d'obligation si $\exists\, a_1, a_2 \in A$ tel que :

$$\neg a_1 \longrightarrow CN_{\neg a_1}$$
$$\neg a_2 \longrightarrow CN_{\neg a_2}$$
$$((a_1 \vee a_2) \wedge \neg(a_1 \wedge a_2)) \quad \vee \quad (a_1 \wedge a_2 \longrightarrow CN_{a_1 \wedge a_2})$$

avec A, est l'ensemble des actions du monde.

Nous généralisons alors pour n actions comme suit :

Dilemme d'obligation (n actions)

$S(A,CN)$ est un dilemme d'obligation si $\exists\, A_{cn} \subset A \;\; et \;\; A_{cn} = \bigcup_{i=1}^{n} a_i \;,\;\; n \geqslant 2$ tel que :

$$\forall\, a \in A_{cn},\;\; \neg a \longrightarrow CN_{\neg a}$$
$$(\vee_1^n a_i \;\wedge\; \neg \wedge_1^n a_i) \quad \vee \quad (\wedge_1^n a_i \;\longrightarrow\; CN_{\wedge_1^n a_i})$$

A est l'ensemble des actions du monde,
A_{cn} est le sous-ensemble d'actions dont la non-réalisation entraîne des conséquences négatives,
CN_a sont les conséquences négatives dues à la réalisation de l'action a.

Les dilemmes de prohibition

Un dilemme de prohibition est une situation où toutes les actions possibles sont prohibées. Dans ce type de dilemme, la condition nécessaire à son émergence est que l'agent est obligé de choisir au moins une action à réaliser. L'enjeu pour le système de génération est de pousser l'agent à faire un choix malgré tout, alors que chacune des actions a des conséquences négatives. Lorsque toutes les actions sont prohibées, la seule issue positive pour l'agent serait de ne rien faire. Afin de garantir le dilemme, il faut écarter cette possibilité. Il faut donc que le non-choix conduise également à des conséquences négatives. Pour illustrer ce cas, prenons l'exemple d'un conducteur qui est collé par une voiture derrière. La voiture de devant freine brusquement. Le conducteur peut choisir de freiner également. Cela entraîne, par conséquent, une collision avec la voiture de derrière. Il peut également choisir d'éviter la voiture de devant en se plaçant dans la voie de gauche. Cela entraîne, en revanche, une collision avec la voiture qui vient à contre-sens. Le conducteur est confronté alors à deux issues dont les conséquences sont négatives. Il peut toutefois choisir de ne rien faire. Si cette issue n'entraîne pas de conséquences négatives, le dilemme est compromis. Dans cet exemple, ce n'est pas le cas. En effet, si le conducteur choisit de ne rien faire (ne pas freiner et ne pas éviter la voiture), cela entraîne une collision avec la voiture de devant. Étant donné ces éléments, nous formalisons un dilemme de prohibition de la manière décrite ci-après.

Soit a_1 et a_2 deux actions qu'un agent décideur AD peut réaliser. Pour certaines raisons, AD ne doit pas réaliser a_1. En d'autres termes, faire a_1 conduit à des conséquences négatives CN_{a_1}. Pour d'autres raisons, AD ne doit pas réaliser a_2. En d'autres termes, faire a_2 conduit à des conséquences négatives CN_{a_2}. La condition nécessaire dans un dilemme de prohibition est que, malgré cela, l'agent doit réaliser au moins une action. Par conséquent, s'agissant d'une situation mettant en jeu deux actions, nous formalisons un dilemme de prohibition de la manière suivante :

Dilemme de prohibition (deux actions)

$S(A,CN)$ est une situation de dilemme de prohibition si $\exists\, a_1, a_2 \in A$ tel que :

$$a_1 \longrightarrow CN_{a_1}$$
$$a_2 \longrightarrow CN_{a_2}$$
$$\neg a_1 \wedge \neg a_2 \longrightarrow CN_{\neg a_1 \wedge \neg a_2}$$

avec A, l'ensemble des actions du monde.

Nous généralisons alors pour n actions comme suit :

Dilemme de prohibition (n actions)

$S(A,CN)$ est un dilemme de prohibition si $\exists\, A_{cn} \subset A$ *et* $A_{cn} = \bigcup_{i=1}^{n} a_i\ ,\ \ n \geqslant 2$ tel que :

$$\forall\, a\ \in\ A_{cn},\ \ a \longrightarrow CN_a$$
$$\wedge_1^n \neg a_i \longrightarrow CN_{\wedge_1^n \neg a_i}$$

A est l'ensemble des actions du monde,
A_{cn} est le sous-ensemble d'actions dont la réalisation entraîne des conséquences négatives,
CN_a sont les conséquences négatives dues à la réalisation de l'action a.

3.2.4 Les étapes de génération

Maintenant que nous avons défini précisément ce qu'est un dilemme de prohibition et un dilemme d'obligation, et que nous avons décrit les différents types de conséquences négatives, nous détaillons dans cette section l'ensemble des étapes du processus de génération de ces deux types de dilemmes.

Nous avons défini un dilemme comme étant une situation où toutes les issues possibles conduisent à des conséquences négatives. Une issue étant le fait de faire ou ne pas faire une action. Par conséquent, la première étape de génération consiste à rechercher les actions dont la réalisation ou la non-réalisation conduit à des conséquences négatives. Le moteur KOBA sépare ensuite les actions en deux listes. Une liste contenant les actions dont la réalisation conduit à des conséquences négatives, et une deuxième, contenant les actions dont la non-réalisation conduit à des conséquences négatives. La première correspond à la base d'actions qui serviront potentiellement pour la génération des dilemmes de prohibition, tandis que la deuxième correspond à la base d'actions qui serviront potentiellement pour la génération des dilemmes d'obligation. Une fois les deux listes établies, le moteur couple les actions les unes avec les autres afin de trouver des paires d'actions qui répondent aux conditions nécessaires de chaque type de dilemme. Pour les dilemmes d'obligation, le moteur retient les paires dont les actions sont contradictoires (pour assurer l'exclusivité du choix), tandis que pour les dilemmes de prohibition, il retient les paires dont la non-réalisation de leurs deux actions entraînent également des conséquences négatives (pour assurer que le non-choix ne soit pas une option positive). Lorsque les paires d'actions sont construites, le moteur procède à des vérifications d'instanciation. Cela consiste à vérifier que les conditions d'instanciation des actions de chaque paire sont compatibles. Si ce n'est pas le cas, la paire d'actions est écartée. Ensuite, afin de proposer le dilemme le plus pertinent pour l'apprenant, KOBA établit un

classement des paires d'actions en fonction d'un score calculé selon différents paramètres. A partir de ce classement, la paire d'action avec le plus grand score est sélectionnée. Le moteur extrait ensuite un état du monde propice à l'instanciation d'une situation mettant en jeu les deux actions de la paire. Cet état du monde est transmis au moteur de planification qui se charge de diriger la simulation vers cet état but.

Dans la figure 3.6, nous illustrons l'ensemble des étapes de génération que nous venons de présenter. Ces étapes sont décrites plus en détail dans la suite de cette section.

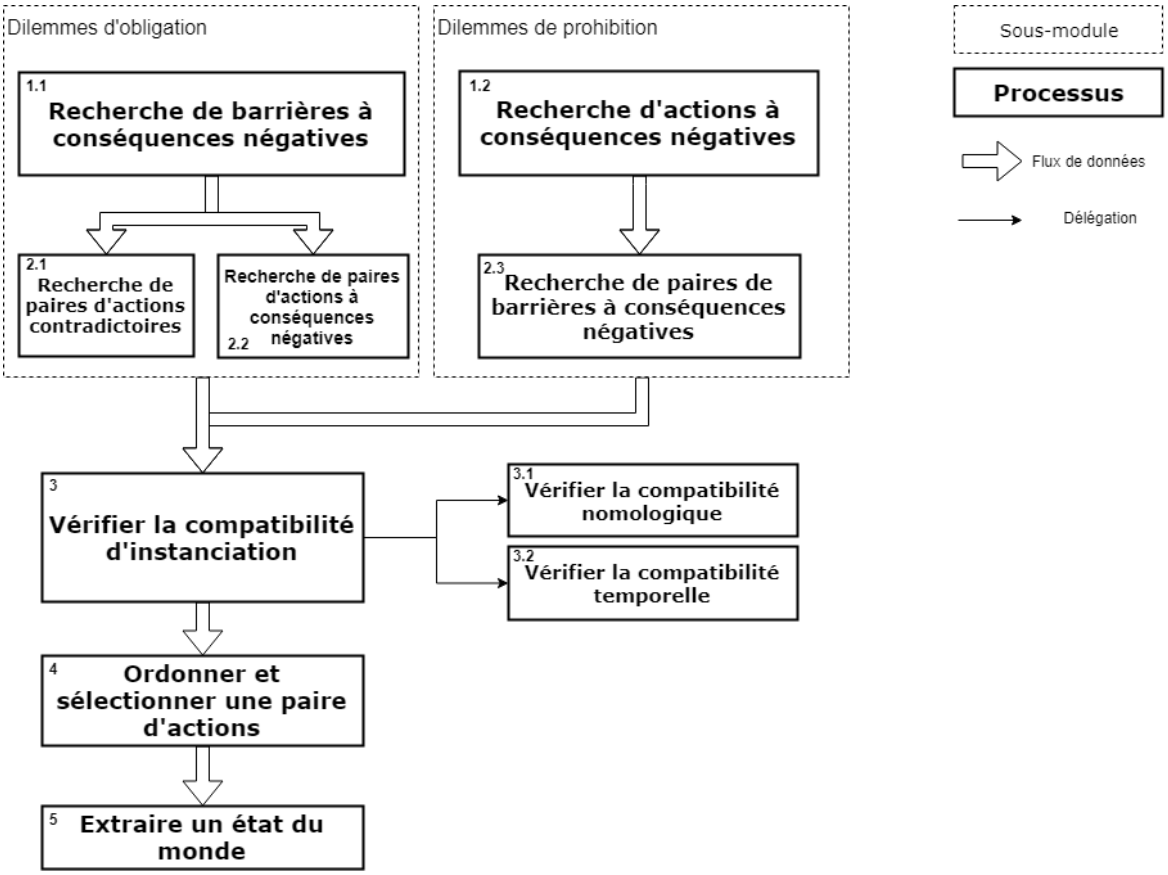

FIGURE 3.6 Étapes de génération des dilemmes

Génération des dilemmes d'obligation

La génération des dilemmes d'obligation passe tout d'abord par la recherche des actions dans la non-réalisation conduit à des conséquences négatives. Pour extraire cette information, KOBA utilise le modèle de causalité décrit dans 3.1.3. Rappelons que ce modèle est un graphe orienté acyclique qui permet de représenter les chaînes de causalité pertinentes entre les événements (nœuds du graphe). Un événement ou une action conduisent à une conséquence

négative, s'il existe un chemin entre eux et un nœud terminal de conséquence négative : GRAVITÉ, VIOLATIONS NORMATIVES, ou PERTE DE POINTS. Ce qui nous intéresse ici, ce ne sont pas les conséquences liées à la réalisation des actions, mais plutôt les conséquences liées à leur non-réalisation. Par conséquent, le moteur raisonne plutôt sur les barrières. Rappelons qu'une barrière (qui peut être une action réalisable par l'apprenant) conditionne le déclenchement des événements qui lui sont postérieurs. C'est-à-dire que les événements ne sont déclenchés que si les barrières ne sont pas réalisées. Cela correspond exactement à ce que nous recherchons. Par conséquent, la recherche d'actions dont la non-réalisation conduit à des conséquences négatives revient à retrouver les barrières dont les conséquences sont négatives (algorithme 1).

Algorithm 1 Recherche de barrières à conséquences négatives

Require: *Barriers* $\neq \emptyset$ {liste des barrières dans le modèle de causalité}
  $B \leftarrow \emptyset$ {Liste des barrières à conséquences négatives}
  for all $b \in$ *Barriers* **do**
    if EXISTSPATH(b, GRAVITÉ) **or**
    EXISTSPATH(b, VIOLATIONSNORMATIVES) **or**
    EXISTSPATH(b, PERTEDEPOINTS) **then**
      $B \leftarrow \mathrm{B} \cup \{b\}$
  return B

Dans l'exemple présenté dans la figure 3.7, la barrière b_2 est retenue car il existe un chemin entre cette barrière et un nœud terminal de conséquence négative (GRAVITÉ). Les barrières b_3 et b_4 sont également retenues car il existe un chemin entre ces barrières et le nœud terminal de conséquence négative : VIOLATIONS NORMATIVES. La barrière b_1, quant à elle, est écartée car le nœud terminal ne correspond pas à une conséquence négative.

Dans la formalisation des dilemmes d'obligation, nous avons expliqué que pour garantir le dilemme, le décideur ne doit pas pouvoir réaliser les deux actions. Par conséquent, pour assurer cette condition, les actions mises en jeu doivent être contradictoires. Pour identifier de telles actions, KOBA utilise le modèle de tâches décrit dans 3.1.3. Rappelons que le modèle de tâches correspond à une représentation hiérarchique de tâches reliées par des constructeurs logiques et temporels. Chaque tâche dispose d'un ensemble de préconditions et de postconditions exprimées par des assertions sous forme de triplets (sujet prédicat objet). Les tâches feuilles se réfèrent aux actions référencée s dans le modèle du monde. Dans la suite de ce mémoire, pour simplifier, nous ferons référence aux tâches par le terme "actions" qu'il s'agisse de tâches de haut niveau ou de tâches feuilles. L'extraction des paires d'actions potentiellement contradictoires se fait suivant l'algorithme 2. Le moteur scanne, par paires, l'ensemble des barrières sélectionnées précédemment (actions qui conduisent à

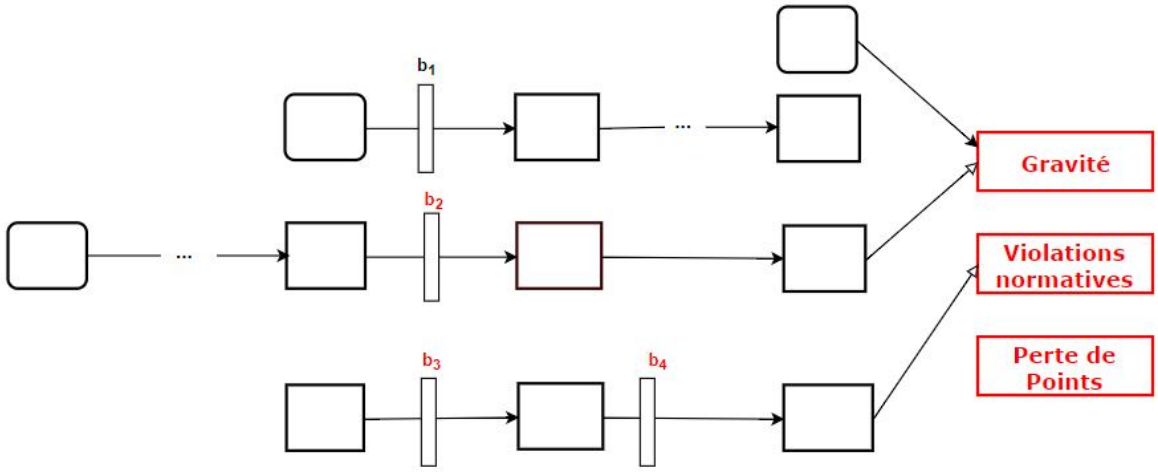

FIGURE 3.7 Recherche de barrières dont les conséquences sont négatives

des conséquences négatives si elles ne sont pas réalisées). Pour chaque paire, il examine les postconditions des actions qui la composent. Si ces préconditions sont incompatibles, alors la paire est nomologiquement incompatible (i.e. opposée par nature). Si elles ne le sont pas, alors la paire d'actions ne peut pas être contradictoire. Elle est donc écartée. Nous considérons que deux conditions sont incompatibles si, pour le même couple (sujet prédicat), l'objet est différent (e.g. `"Vehicle is-stopped true"` et `"Vehicle is-stopped false"`).

Algorithm 2 Recherche d'actions contradictoires

Require: $Actions \neq \emptyset$ {liste d'actions du modèle de tâches}
  $PCA \leftarrow \emptyset$ {Liste des paires d'actions contradictoires}
  for all $a_1 \in Actions$ **do**
    for all $a_2 \in Actions$ **do**
      if a_1.postConditions.subject $=$ a_2.postConditions.subject **and**
      a_1.postConditions.predicate $=$ a_2.postConditions.predicate **and**
      a_1.postConditions.object $\neq$ a_2.postConditions.object **then**
        if $\{a_1, a_2\} \notin$ PCA **then**
          $PCA \leftarrow$ PCA $\cup \{\{a_1, a_2\}\}$
  return PCA

Dans la formalisation des dilemmes d'obligation, nous avons également précisé que dans le cas où les actions ne sont pas contradictoires, il est toujours possible de garantir le dilemme si la réalisation des deux actions conduit également à des conséquences négatives. Pour cela, KOBA déroule l'algorithme 3 sur le modèle de causalité. Il scanne par paires l'ensemble des barrières sélectionnées précédemment, vérifie tout d'abord qu'elles existent aussi en tant qu'actions dans le modèle de causalité, qu'elles ont une porte logique `ET` comme descendant commun, et qu'elles conduisent à un nœud de conséquence négative.

Dans l'exemple présenté dans la figure 3.8, nous supposons que les barrières retournées par l'algorithme 1 correspondent aux actions a_1, a_2 et a_3. La paire d'action $\{a_2, a_3\}$ est retenue car les deux actions entraînent une conséquence négative (GRAVITÉ), et sont connectées par une porte logique ET. Quant aux paires $\{a_1, a_2\}$ et $\{a_1, a_3\}$, bien qu'elles entraînent des conséquences négatives, elles sont écartées parce qu'elles ne sont pas liées par une porte logique ET. Finalement, toutes les paires qui incluent l'action a_4 sont écartées étant donné que a_4 ne conduit pas à des conséquences négatives.

Algorithm 3 Recherche d'actions dont la réalisation simultanée conduit à des conséquences négatives

Require: *Barriers* $\neq \emptyset$ {liste de barrières retournées par l'algorithme 1}
  $A \leftarrow \emptyset$ {liste des paires d'actions dont la réalisation simultanée conduit à des conséquences négatives}
  for all $b_1 \in$ *Barriers* **and** ISACTION(b_1) **do**
    for all $b_2 \in$ *Barriers* **and** ISACTION(b_2) **do**
      AndNode $\leftarrow$ COMMONDESCENDANTAND(b_1, b_2)
      if *AndNode* $\neq$ *nil* **then**
        if EXISTSPATH(*AndNode*, GRAVITÉ) **or**
        EXISTSPATH(*AndNode*, VIOLATIONSNORMATIVES) **or**
        EXISTSPATH(*AndNode*, PERTEDEPOINTS) **then**
          if $\{b_1, b_2\} \notin$ A **then**
            A $\leftarrow$ A $\cup \{\{b_1, b_2\}\}$
  return *A*

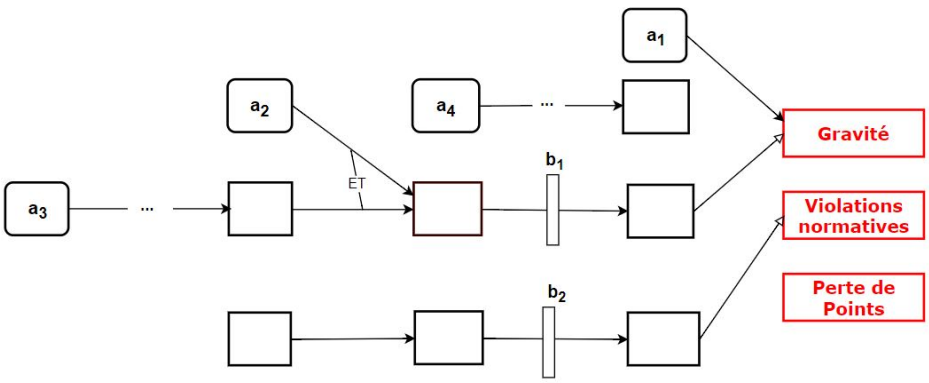

FIGURE 3.8 Recherche d'actions dont la réalisation simultanée conduit à des conséquences négatives

Génération des dilemmes de prohibition

La génération des dilemmes de prohibition passe tout d'abord par la recherche d'actions dont la réalisation conduit à des conséquences négatives. Pour extraire cette information KOBA utilise le modèle de causalité. Il scanne l'ensemble des nœuds action et vérifie s'il existe un chemin entre ces actions et les noeuds terminaux de conséquences négatives. Si c'est le cas, l'action est considérée comme entraînant une conséquence négative. L'algorithme utilisé par le moteur de génération est quasiment identique à l'algorithme 1, à la seule différence qu'au lieu d'effectuer la recherche sur les barrières, il l'effectue sur les actions du modèle. Dans l'exemple présenté dans la figure 3.8, les actions a_1, a_2 et a_3 sont retenues car il existe un chemin entre ces actions et un nœud terminal de conséquences négatives (GRAVITÉ). L'action a_4, quant à elle, est écartée car elle ne conduit pas à un nœud de conséquences négatives.

Dans la formalisation des dilemmes de prohibition, nous avons expliqué que pour garantir le dilemme, le non-choix doit également conduire à des conséquences négatives. Le non-choix correspond ici à la non-réalisation des deux actions. Pour identifier de telles actions, KOBA utilise le modèle de causalité. Pour cela, il scanne, par paires, l'ensemble des actions sélectionnées précédemment, vérifie qu'elles existent en tant que barrières dans le modèle, et garde les barrières qui ont une porte logique ET comme descendant commun. L'algorithme utilisé est quasi-identique à l'algorithme 3, à la seule différence qu'au lieu d'effectuer la recherche sur des paires d'actions, il l'effectue sur des paires de barrières. Dans l'exemple présenté dans la figure 3.9, la paire de barrières $\{b_2, b_3\}$ est retenue car les deux barrières entraînent une conséquence négative (PERTE DE POINTS), et sont connectées par une porte logique ET. Quant aux paires $\{b_1, b_2\}$ et $\{b_1, b_3\}$, elles sont écartées, car a priori, b_1 ne conduit pas à des conséquences négatives.

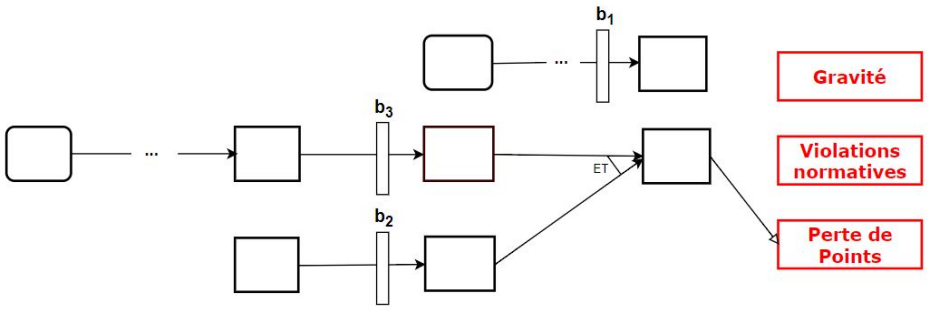

FIGURE 3.9 Recherche de barrières dont la non-réalisation simultanée conduit à des conséquences négatives

Compatibilité d'instanciation

A ce stade, le KOBA dispose de deux listes de paires d'action. Une liste de paires pour les dilemmes de prohibition, et une liste de paires pour les dilemmes d'obligation. Pour que des situations dilemmatiques puissent être instanciées, il faut s'assurer que ces paires d'actions soient instanciables. Par conséquent, le moteur doit vérifier d'une part, la compatibilité nomologique des actions, et d'autre part, leur compatibilité temporelle.

S'agissant de la compatibilité nomologique, le moteur doit s'assurer que les préconditions des actions sont compatibles les unes avec les autres. Rappelons que deux conditions sont incompatibles si pour le même sujet-prédicat, l'objet est différent. Si deux actions d'une paire disposent de préconditions incompatibles, cela contraint l'instanciation d'une situation les mettant en jeu simultanément. Prenons comme exemple les actions `"Ouvrir porte"` et `"Fermer porte"` qui ont respectivement comme préconditions `"?door :has-state :close"` et `"?door :has-state :open"`. L'instanciation d'une situation propice à la réalisation de ces deux actions, nécessite l'agrégation de ces deux préconditions. Par conséquent, dans cet exemple, cela nécessite que la porte soit à la fois ouverte et fermée. Ce qui est impossible. Dans certains modèles de tâches, à l'instar de ACTIVITY-DL, les préconditions sont catégorisées en plusieurs types. Dans ce cas-là, il convient de préciser que l'incompatibilité ne concerne ici que les préconditions qui rendent la tâche physiquement réalisable, à savoir, les préconditions nomologiques.

Algorithm 4 Vérification de la compatibilité nomologique

Require: $ActionsPairs \neq \emptyset$ {liste des paires d'actions}
  for all $\{a_1, a_2\} \in ActionsPairs$ **do**
    if a_1.preconditions.subject $=$ a_2.preconditions.subject **and**
    a_1.preconditions.predicate $=$ a_2.preconditions.predicate **and**
    a_1.preconditions.object $\neq$ a_2.preconditions.object **then**
      $ActionsPairs \leftarrow ActionsPairs - \{a_1, a_2\}$
  return $ActionsPairs$

En plus de la compatibilité nomologique, les actions doivent être temporellement compatibles. C'est-à-dire que la réalisation d'une action doit être indépendante de la réalisation de l'autre. Dans le modèle de tâches, les actions sont liées par des constructeurs logiques et temporels. KOBA utilise donc ce modèle pour identifier toutes les actions dont l'ancêtre commun dispose d'un constructeur temporel séquentiel `(SEQ)`. Lorsque deux actions sont liées par un constructeur temporel séquentiel, cela veut dire que la réalisation de l'une d'entre elles n'est possible que si l'autre a déjà été réalisée. Par conséquent, le moteur de génération ne retient que les actions qui sont liées par des constructeurs temporels indépendants `(IND)` ou parallèles `(PAR)`.

Algorithm 5 Vérification de la compatibilité temporelle

Require: $ActionsPairs \neq \emptyset$ {liste des paires d'actions}
  for all $\{a_1, a_2\} \in ActionsPairs$ **do**
    if CONSTRUCTOR(COMMONANCESTOR(a_1, a_2)) = `SEQ` **then**
      $ActionsPairs \leftarrow ActionsPairs - \{a_1, a_2\}$
  return $ActionsPairs$

Ordonner et sélectionner une paire d'actions

A ce stade, KOBA dispose de deux listes de paires d'actions qui vérifient les compatibilités d'instanciation. Toutefois, afin de choisir le dilemme le plus pertinent à présenter à l'apprenant, le moteur doit établir un classement de ces paires d'actions. Pour cela, il attribue à chaque paire un score qui prend en compte différentes contraintes. Nous distinguons deux types de contraintes : les **contraintes pédagogiques**, qui sont plutôt liées aux propriétés auxquelles doit répondre la situation dilemmatique afin d'atteindre certains objectifs pédagogiques, et les **contraintes scénaristiques**, qui sont plutôt liées aux préférences qui relèvent du scénario.

Les contraintes pédagogiques sont définies par :

— **une gravité maximale** à ne pas dépasser,
— **une gravité minimale** à respecter,
— **une différence de gravité** entre les actions d'une paire,
— **un ou des types de conséquence négatives** à cibler.

La paire d'action qui répond au mieux à ces critères pédagogiques est celle qui aura le plus de chance d'être sélectionnée. Nous pouvons alors formaliser cela comme étant un problème d'optimisation multicritère :

Optimisation multicritère

$$Max\ \{f_{gma}(x), f_{gmi}(x), f_{dg}(x), f_{tc}(x)\ /\ x \in A\}$$

avec

A : l'ensemble des solutions potentielles (l'ensemble des paires d'actions).
$f_{gma}(x)$: critère d'évaluation à optimiser pour la gravité maximale.
$f_{gmi}(x)$: critère d'évaluation à optimiser pour la gravité minimale.
$f_{dg}(x)$: critère d'évaluation à optimiser pour la différence de gravité.
$f_{tc}(x)$: critère d'évaluation à optimiser pour le type de conséquence.

Par ailleurs nous définissons une contrainte pédagogique comme suit :

Contrainte pédagogique

Une contrainte pédagogique C_p est constituée d'un couple $< t_p, R_p >$ tel que $t_p \in D_p$ est une propriété de la paire d'actions et R_p une relation dans D_p.
Avec D_p, le domaine de définition de la propriété.

La contrainte de **gravité maximale** permet de contrôler la sévérité de la situation. C'est à dire le degré de dommage potentiel engendré par la situation. Dans certains cas, nous voulons nous assurer que la gravité ne dépasse pas une certaine valeur afin de ne pas traumatiser l'apprenant, et/ou afin de faciliter, à un certain degré, la réalisation de la tâche qu'il entreprend. Nous définissons alors la contrainte de gravité maximale comme suit :

Contrainte de gravité maximale

Une contrainte de gravité maximale C_{gma} est constituée d'un couple $< t_{gma}, R_{gma} >$ tel que $t_{gma} \in [1,10]$ est la gravité de la paire de tâches, et R_{gma} une relation telle que $t_{gma} \leq t_{max}$.
Avec t_{max}, la gravité maximale désirée.

Pour cette contrainte, la gravité t_{gma} d'une paire d'actions $< a_1, a_2 >$ correspond à la gravité maximale des actions qui la composent :

$$t_{gma} = max(g_{a_1}, g_{a_2})$$

Afin de calculer le degré de satisfaisabilité de cette contrainte par une paire d'actions, nous définissons une fonction booléenne qui retourne 1 si la contrainte est respectée, et 0 sinon.

Satisfaction de la contrainte de gravité maximale

$$f_{gma} : \mathbb{N} \rightarrow \{0,1\}$$
$$x \mapsto f_{gma}(x)$$

$$f_{gma} = \begin{cases} 1 & \text{si } x \leq t_{max} \\ 0 & \text{sinon} \end{cases}$$

avec t_{max} la gravité maximale à ne pas dépasser.

Dans d'autres cas, nous souhaitons augmenter l'intensité de la situation et la criticité des enjeux. Pour cela, il est possible de spécifier un seuil minimal de gravité que la situation doit respecter. Nous définissons alors une contrainte de gravité minimale de la manière suivante :

Contrainte de gravité minimale

Une contrainte de gravité minimale C_{gmi} est constituée d'un couple $< t_{gmi}, R_{gmi} >$ tel que $t_{gmi} \in [1,10]$ est la gravité de la paire de tâche, et R_{gmi} une relation tel que $t_{gmi} \geq t_{min}$.
Avec t_{min}, la gravité minimale désirée.

Pour cette contrainte, la gravité t_{gmi} d'une paire d'actions $< a_1, a_2 >$ correspond à la gravité minimale des actions qui la composent :

$$t_{gmi} = min(g_{a_1}, g_{a_2})$$

Afin de calculer le degré de satisfaisabilité de cette contrainte par une paire d'actions, nous définissons une fonction booléenne qui retourne 1 si la contrainte est respectée, et 0 sinon.

Satisfaction de la contrainte de gravité minimale

$$f_{gmi} : \mathbb{N} \to \{0,1\}$$
$$x \mapsto f_{gmi}(x)$$

$$f_{gmi} = \begin{cases} 1 & \text{si } x \geq t_{min} \\ 0 & \text{sinon} \end{cases}$$

avec t_{min} la gravité minimale désirée.

La troisième contrainte concerne la **différence de gravité** entre les actions d'une paire. Nous pensons que dans certains cas, plus la différence de gravité entre les deux actions est faible, plus le dilemme est intense. Choisir entre sacrifier une personne pour en sauver deux autres est sans doute plus difficile que de choisir entre sacrifier une personne et en sauver mille. Par ailleurs, lorsque la différence de gravité est nulle, cela permet d'écarter les situations où le comportement utilitariste peut être observé chez le décideur. En effet, si la gravité engendrée est la même dans les deux issues (e.g. sacrifice du même nombre de personnes), raisonner purement en terme de conséquences n'est plus possible dans ce cas-là. Nous définissons alors la contrainte de différence de gravité de la manière suivante :

Contrainte de différence de gravité

Une contrainte de différence de gravité C_{dg} est constituée d'un couple $< t_{dg}, R_{dg} >$ tel que $t_{dg} \in [0,9]$ est la différence de gravité entre les actions de la paire, et R_{dg} une relation telle que $t_{dg} = t_{dd}$.
Avec t_{dd}, la différence de gravité désirée.

Pour cette contrainte, la différence de gravité t_{dg} entre les actions d'une paire $< a_1, a_2 >$, correspond à la valeur absolue de la différence entre la gravité de chaque action qui la compose :

$$t_{dg} = |g_{a_1} - g_{a_2}|$$

Afin de calculer le degré de satisfaisabilité de cette contrainte par une paire d'actions, nous définissons une fonction qui retourne l'écart normalisé entre la différence de gravité d'une paire d'actions et la différence de gravité désirée.

Satisfaction de la contrainte de différence de gravité

$$\begin{aligned} f_{dg} &: \mathbb{N} \to [0, 10] \\ x &\mapsto \frac{9 - |x - t_{dd}|}{9} \end{aligned}$$

avec t_{dd} la différence de gravité désirée.

Finalement, la dernière contrainte pédagogique concerne le **type de conséquences négatives**. Elle permet de se restreindre à un ou plusieurs types particuliers de conséquences négatives. Par exemple, nous pouvons cibler en particulier les violations morales dans les conséquences afin de générer des dilemmes moraux. Nous définissons alors la contrainte de types de conséquences négatives de la manière suivante :

Contrainte de types de conséquences négatives

Une contrainte de types de conséquences C_{tc} est constituée d'un couple $< t_{tc}, R_{tc} >$ tel que $t_{tc} \in CN$ est l'ensemble des types de conséquences négatives de la paire d'actions, et R_{tc} une relation tel que $t_{tc} = t_{td}$.
Avec t_{td}, l'ensemble des types de conséquences désirés, et CN l'ensemble des types de conséquences négatives du modèle.

Pour cette contrainte, l'ensemble des types de conséquences négatives t_{tc} d'une paire d'actions $< a_1, a_2 >$, correspond à l'union des ensembles de types de conséquences négatives de chaque action qui la compose :

$$t_{tc} = CN_{a_1} \cup CN_{a_2}$$

Afin de calculer le degré de satisfaisabilité de cette contrainte par une paire d'actions, nous définissons une fonction qui retourne le cardinal normalisé de la différence symétrique entre l'ensemble des conséquences négatives de la paire et l'ensemble des types de conséquences négatives désirés.

Satisfaction de la contrainte de types de conséquences négatives

$$f_{tc} : X \to \mathbb{N}$$
$$x \mapsto \frac{Card(CN) - Card(\, x \,\Delta\, t_{tc})}{Card(CN)}$$

avec
t_{tc}, l'ensemble des types de conséquences négatives souhaité,
CN, l'ensemble des types de conséquences du modèle.

Afin de tenir compte de l'importance particulière que représente chaque contrainte, nous définissons un poids ω_i associé à chacune des contraintes. Cela permet de calculer un score global s_{cp} de satisfaction des contraintes pédagogiques par chaque paire d'actions.

Score de satisfaction des contraintes pédagogiques

$$s_{cp} = f_{gma} * \omega_{gma} + f_{gmi} * \omega_{gmi} + f_{dg} * \omega_{dg} + f_{tc} * \omega_{tc}$$
$$\text{avec } \omega_{gma} + \omega_{gmi} + \omega_{dg} + \omega_{tc} = 1$$

Après avoir détaillé l'ensemble des contraintes pédagogiques, nous décrivons dans la suite, le deuxième type de contraintes qui sont les **contraintes scénaristiques**. Ces dernières correspondent aux critères auxquels doivent répondre les paires d'actions afin d'assurer une bonne continuité du scénario. Nous définissons les contraintes suivantes :

— **la probabilité d'instanciation minimale** : désigne le degré de certitude minimal que doit avoir le moteur de génération par rapport à la possibilité d'instanciation d'une paire d'actions, étant donné l'état du monde courant.

— **le temps maximal nécessaire à l'instanciation** : correspond au temps théorique nécessaire à la mise en place de la situation mettant en jeu la paire d'actions.

Nos travaux s'inscrivent dans le cadre de la scénarisation dans les domaines complexes. Dans ces domaines, le nombre d'interactions entre les entités du monde ainsi que le panel d'évolutions possibles sont très grandes. Aussi, l'apprenant jouit d'une liberté d'action dans un monde qui évolue de manière dynamique. Pour ces raisons, la spécification des objectifs scénaristiques ne garantit pas systématiquement leur instanciation en environnement virtuel. Cela est dû aux incertitudes engendrées par les comportements imprévisibles de l'apprenant et des entités du monde. Par conséquent, en fonction de l'état du monde courant, chaque paire d'actions dispose d'une probabilité d'instanciation. Cette probabilité représente le degré de certitude du système par rapport à l'instanciation d'une situation mettant en jeu les deux actions. Nous définissons alors la contrainte de **probabilité d'instanciation minimale**, qui

correspond au degré de certitude minimal dont doit disposer le moteur de génération sur l'instanciation d'une situation mettant en jeu une paire de tâches.

Contrainte de probabilité d'instanciation minimale

Une contrainte de probabilité d'instanciation minimale C_{pim} est constituée d'un couple $< t_{pim}, R_{pim} >$ tel que $t_{pim} \in [0,1]$ est la probabilité d'instanciation de la paire, et R_{pim} une relation tel que $t_{pim} \geq t_{min}$.
Avec t_{min}, la probabilité d'instanciation minimale à respecter.

Le calcul de la probabilité d'instanciation d'une paire d'actions n'a pas fait l'objet de réflexions approfondies dans le cadre de nos travaux, d'autant plus, que cela relève plus de la planification que de la spécification d'objectifs scénaristiques. Toutefois, nous pensons que les éléments suivants doivent être pris en compte dans le calcul de cette probabilité :

- **le nombre de préconditions à réaliser** : chaque paire d'actions dispose de préconditions nécessaires à son instanciation. Selon l'état du monde, il se peut que certaines de ces préconditions soient déjà vérifiées. Nous pensons que la probabilité d'instanciation d'une paire est inversement proportionnelle au nombre des préconditions qui restent à vérifier.
- **la réalisation des préconditions** : la prise en compte uniquement du nombre de préconditions non vérifiées n'est pas suffisant pour calculer la probabilité d'instanciation de la paire. Il faut également prendre en compte la complexité de leur réalisation. En effet, il se peut que dans certains cas, dix préconditions réunies soient réalisables plus facilement qu'une seule précondition. Cela peut s'expliquer par le fait que le nombre d'événements nécessaires à la réalisation d'une précondition soit différent selon la précondition que l'on cherche à réaliser. Par conséquent, nous pensons que la probabilité d'instanciation d'une paire d'actions est inversement proportionnelle au nombre d'événements nécessaires à la réalisation de ses préconditions non vérifiées.

Afin de calculer le degré de satisfaisabilité de cette contrainte par une paire d'actions, nous définissons une fonction qui retourne la différence entre la probabilité d'instanciation d'une paire et la probabilité d'instanciation minimale exigée, si la première est supérieure à la deuxième. La fonction retourne 0 sinon.

Satisfaction de la contrainte de probabilité d'instanciation minimale

$$f_{pim}\colon [0,1] \to [0,1]$$
$$x \mapsto f_{pim}(x)$$

$$f_{pim} = \begin{cases} x - t_{min} & \text{si } t_{min} \leq x \leq 1 \\ 0 & \text{sinon} \end{cases}$$

avec t_{min} la probabilité d'instanciation minimale à respecter.

La deuxième contrainte scénaristique concerne le **temps maximal nécessaire** à la mise en place d'une situation mettant en jeu la paire d'actions. Dans certains cas, nous souhaitons produire une rétroaction immédiate. Par conséquent, il est important de pouvoir contrôler le temps nécessaire à la génération de la prochaine situation. Nous définissons alors une contrainte de **temps maximal nécessaire à l'instanciation**, qui correspond au temps théorique maximal nécessaire à la génération de la situation mettant en jeu les deux actions de la paire.

Contrainte de temps maximal nécessaire à l'instanciation

Une contrainte de temps maximal nécessaire à l'instanciation C_{tmi} est constituée d'un couple $< t_{tmi}, R_{tmi} >$ tel que $t_{tmi} \in \mathbb{N}$ est le temps théorique nécessaire à l'instanciation de la paire, et R_{tmi} une relation tel que $t_{tmi} \leq t_{max}$.
Avec t_{max}, le temps maximal nécessaire à l'instanciation à ne pas dépasser.

A l'instar de la probabilité d'instanciation, et pour les mêmes raisons que celles qui ont été évoquées plus haut, le calcul du temps théorique nécessaire à l'instanciation d'une paire d'action n'a pas fait l'objet de réflexions approfondies dans le cadre de nos travaux. Toutefois, nous pensons que le temps nécessaire à l'instanciation peut se résumer à la somme des durées des événements duratifs nécessaires à la réalisation des préconditions des actions.

Afin de calculer le degré de satisfaisabilité de cette contrainte par une paire d'actions, nous définissons une fonction booléenne qui retoune 1 si la contrainte est respectée, et 0 sinon.

Satisfaction de la contrainte de temps maximal nécessaire à l'instanciation

$$f_{tmi}: \mathbb{N} \to \{0,1\}$$
$$x \mapsto f_{tmi}(x)$$

$$f_{tmi} = \begin{cases} 1 & \text{si } x \leq t_{max} \\ 0 & \text{sinon} \end{cases}$$

avec t_{max} le temps maximal nécessaire à l'instanciation à ne pas dépasser.

Afin de tenir compte de l'importance particulière que représente chaque contrainte scénaristique, nous définissons un poids ω_i associé à chacune d'elle. Cela permet de calculer un score global s_{cs} de satisfaction des contraintes scénaristiques par chaque paire d'actions.

Score de satisfaction des contraintes scénaristiques

$$s_{cs} = f_{pim} * \omega_{pim} + f_{tmi} * \omega_{tmi}$$
$$\text{avec } \omega_{pim} + \omega_{tmi} = 1$$

La paire d'action idéale est celle qui répond parfaitement aux contraintes pédagogiques et scénaristiques. En pratique, ce n'est pas toujours le cas. Par conséquent, parfois on peut vouloir qu'un dilemme réponde absolument aux contraintes pédagogiques même si les chances qu'il soit effectivement instancié soient infimes. A l'inverse, on peut vouloir qu'un dilemme soit absolument instancié même s'il ne répond pas exactement aux contraintes pédagogiques. Pour permettre de telles préférences, il est possible d'attribuer un poids à chaque type de contraintes. Par conséquent, le moteur de génération calcule un score total pour chaque paire d'actions de la manière suivante :

Score total

$$s_{total} = s_{cp} * \omega_{cp} + s_{cs} * \omega_{cs}$$
$$\text{avec } \omega_{cp} + \omega_{cs} = 1$$

Extraction d'un état du monde

A ce stade, KOBA dispose d'une liste triée de paires d'actions. A partir de cette liste, il sélectionne la paire qui maximise le score total calculé à l'étape précédente. Pour instancier une situation propice à la réalisation de cette paire, le moteur **prescrit** un état du monde. Ce dernier correspond à une description partielle du monde sous forme d'un agrégat d'asser-

tions. Il correspond en particulier à l'agrégation des préconditions des actions de la paire sélectionnée :

$$E_{prescrit} = preconditions_{a_1} \cup preconditions_{a_2}$$

Cet état du monde prescrit, est en réalité un objectif scénaristique. Il est transmis au moteur de planification qui se charge de diriger la simulation vers un état qui remplit cet objectif. L'exemple illustré dans la figure 3.10, décrit l'extraction d'un état du monde à partir de la paire d'actions : <`Respecter_le_feu_rouge`, `Gérer_aquaplaning`>. Une situation propice à la réalisation de ces deux actions, est une situation où le véhicule est en aquaplaning, et où il existe un feu de circulation rouge actif.

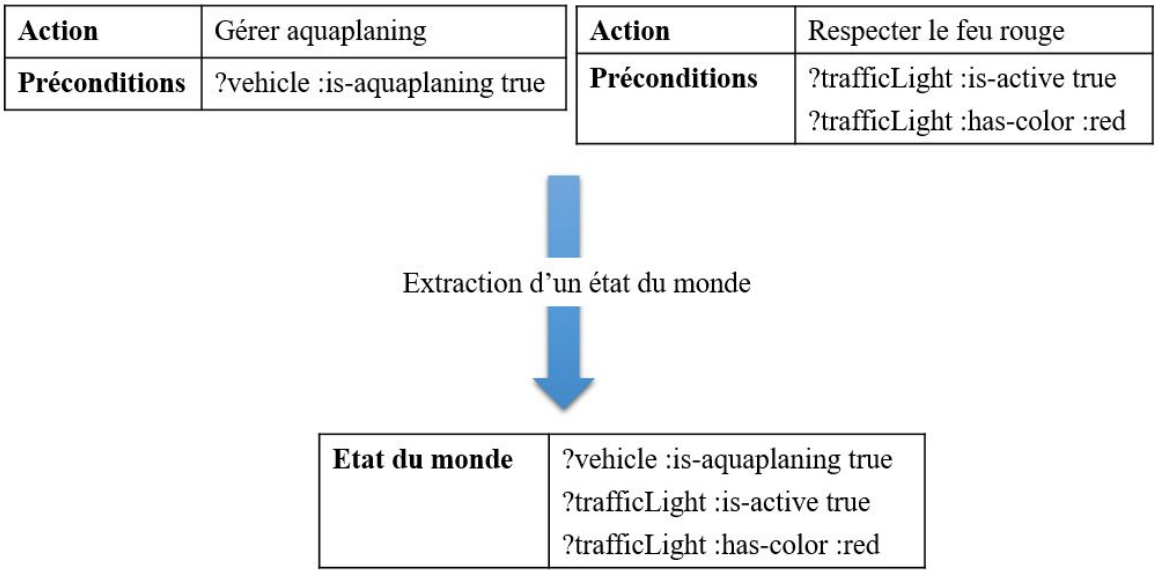

FIGURE 3.10 Extraction d'un état du monde

Toutefois, dans certains cas, la prescription d'un état du monde n'est pas suffisante pour garantir l'émergence d'un dilemme. Parfois, il est également nécessaire de **proscrire** un état du monde. C'est le cas lorsqu'il existe des éléments qui risquent d'inhiber le déclenchement des conséquences négatives attendues, et donc qui risquent de mettre en péril le dilemme. Par conséquent, le moteur de génération doit veiller à ce que ces éléments ne puissent pas être instanciés. En d'autres termes, il doit proscrire l'état du monde propice à leur déclenchement. Les éléments dont il pourrait s'agir ici, sont les barrières de prévention et de protection que peut mettre en place l'apprenant. Rappelons que la première étape du processus de génération des dilemmes consiste à rechercher les actions dont la réalisation ou la non-réalisation entraîne des conséquences négatives. Cela se fait en vérifiant l'existence d'un chemin entre les actions et les nœuds de conséquences négatives. Toutefois, il se peut qu'il existe dans ce

chemin, des barrières qui pourraient empêcher le déclenchement des événements qui leur sont postérieurs. Par conséquent, le moteur de génération doit proscrire les états du monde propices à la réalisation de ces barrières en proscrivant leurs préconditions.

3.2.5 Bilan

Dans cette section, nous avons détaillé notre contribution principale dans le cadre de cette thèse qui consiste en la génération dynamique des dilemmes. Tout d'abord, nous avons déterminé l'ensemble des éléments que nous considérons comme des conséquences négatives. Ensuite, en nous inspirant de la littérature en sciences humaines et sociales, nous avons décrit une formalisation des dilemmes d'obligation et des dilemmes de prohibition. A partir de cette formalisation, pour générer ces dilemmes, nous avons proposé KOBA : un moteur qui raisonne sur des modèles sémantiques afin d'extraire les paires d'actions qui vérifient les propriétés énoncées dans la formalisation. L'utilisation de modèles sémantiques expressifs permet à la fois un raisonnement à différents niveaux d'abstraction, et l'inférence de connaissances nouvelles permettant l'émergence des situations dilemmatiques. Par ailleurs, afin de répondre au mieux aux contraintes pédagogiques et scénaristiques nous avons proposé de classer les paires d'actions en fonction d'un score calculé selon la satisfaisabilité de ces paires à ces contraintes. Finalement, pour la génération des dilemmes moraux, il est nécessaire de prendre en considération les valeurs morales propres à chaque individu. Pour cela, nous avons proposé une opérationnalisation de la théorie des valeurs universelles de (Schwartz, 2006) qui décrit une taxonomie de valeurs humaines représentées dans une structure circulaire qui renseigne sur les relations de compatibilités et d'opposition entre les valeurs.

3.3 Génération de l'ambiguité

Notre deuxième contribution dans le cadre de cette thèse concerne la génération dynamique de l'ambiguïté. Pour cette deuxième dimension de criticité, notre apport a été plus théorique que pratique. Dans cette section, nous proposons des fondations théoriques qui serviront de bases au moteur de génération pour la production des situations ambiguës. L'état de l'art sur l'ambiguïté nous a permis de constater que l'ambiguïté existait sous trois formes :

— **l'ambiguïté de l'information** qui se réfère à l'ambiguïté engendrée par le manque remédiable d'informations et/ou la qualité des informations à disposition.

— **l'ambiguïté du contexte** qui se rapporte à l'ambiguïté qui émerge non pas à cause de l'imperfection des informations à disposition, mais plutôt à cause de la présence de différents contextes, impliquant chacun une interprétation différente.

— **l'ambiguïté normative** qui se réfère à l'ambiguïté qui émerge suite aux valeurs, principes et normes, parfois contradictoires, entraînant différentes interprétations des informations à disposition.

Dans le cadre de nos travaux, nous nous sommes particulièrement intéressés à l'ambiguïté sous ses deux premières formes.

3.3.1 Approche

L'approche que nous proposons pour la génération de l'ambiguïté, repose sur des techniques inspirées de (Gaver et al., 2003) pour la production de l'ambiguïté. Tout d'abord, nous proposons l'utilisation de représentations **incertaines** afin de générer l'ambiguïté de l'information. Cela amène chaque individu à combler le vide à sa manière, et crée par conséquent un champ d'interprétation. Afin de produire ces représentations incertaines, nous proposons de réaliser cela de deux manières. Premièrement, en dissimulant des informations pertinentes afin de créer un état d'**incomplétude**. Et deuxièmement, en mettant du doute sur les sources d'informations afin de créer un état de **flou**. Par ailleurs, pour générer l'ambiguïté du contexte, nous proposons de produire des **inconsistances** entre les représentations. Nous utilisons pour cela des contextes incompatibles qui mobilisent des actions contradictoires, ou des contextes qui accentuent le conflit dans les croyances de l'apprenant. Afin d'identifier quelles sont les informations pertinentes pour l'apprenant et quel est son degré de connaissance de ces informations, il est nécessaire de modéliser comment l'apprenant se représente l'état courant du monde. Nous proposons alors une modélisation de la représentation mentale de l'apprenant. Par ailleurs pour prendre en compte les incertitudes dans cette représentation, nous proposons d'utiliser un cadre formel de représentation d'informations incertaines qui permet de représenter à la fois les états d'ignorance, de flou et de conflit dans les croyances.

3.3.2 Représentation mentale incertaine de l'apprenant

En ergonomie cognitive, un modèle mental est défini comme étant une forme de représentation interne d'une réalité externe. Il contient des connaissances déclaratives qui se rapportent aux connaissances théoriques (e.g code de la route), et des connaissances procédurales qui se rapportent à la manière de réaliser les tâches (e.g. ralentir et freiner pour s'arrêter au feu rouge). Le modèle de tâches que nous avons décrit dans la section 3.1.3, permet de modéliser en partie le modèle mental de l'apprenant. Rappelons-le, le modèle de tâches est utilisé pour modéliser l'activité observée sur le terrain telle que les opérateurs se la représentent. Pour compléter ce modèle, il est nécessaire de se munir d'un modèle pour

décrire comment l'apprenant se représente l'état du monde courant. C'est à dire, quelles sont les valeurs que l'apprenant associe aux propriétés des entités du monde.

Besoins

Afin de pouvoir produire de l'ambiguïté, il est nécessaire de modéliser la façon dont l'apprenant se représente l'état du monde. Plus précisément, il est nécessaire de modéliser le degré de connaissance de l'apprenant des variables pertinentes pour la réalisation de ses tâches. A partir de cette représentation, il doit être possible d'identifier les informations qui sont complètement inconnues de l'apprenant (état d'incomplétude), quelles sont les informations incertaines (état de flou), et enfin, quelles sont les informations qui sont contradictoires (état d'incohérence). Par ailleurs, la représentation que se fait l'apprenant du monde se met à jour au fur et à mesure qu'il avance dans la simulation. En particulier, lorsqu'il effectue des actions ou lorsque des événements du monde se déclenchent. Par exemple, l'apprenant n'est conscient qu'une voiture est derrière lui que lorsqu'il regarde dans les rétroviseurs intérieurs ou extérieurs. Tant qu'il n'a pas regardé dans les rétros, il y aura toujours une différence entre sa représentation mentale et la représentation réelle (état du monde courant). Il est donc nécessaire de disposer de mécanismes qui permettent de mettre à jour les informations. Aussi, lorsque l'apprenant reçoit une information, celle-ci peut provenir de différentes sources qui ont chacune un de degré de fiabilité. Par ailleurs, ces sources peuvent porter des informations qui se contredisent. Il est donc nécessaire de pouvoir fusionner des informations provenant de sources différentes, tout en mettant en exergue le potentiel conflit émergeant de leur fusion.

Afin de représenter tous ces aspects, il est nécessaire alors de se munir d'un cadre formel qui permette de décrire à la fois l'ignorance, le flou et la contradiction, tout en permettant la fusion d'informations à partir de différentes sources.

Cadre formel

Pour répondre à ces besoins, nous avons considéré les cadres formels suivants :

— **la théorie des ensembles** permet de représenter des informations imprécises. Une information se représente grâce à un ensemble ou un intervalle de valeurs (e.g. `vitesseVoiture` $\in [90,100]$, `CouleurFeu` $\in \{Rouge, Orange, Vert\}$) ;

— **la théorie des probabilités** permet de représenter l'incertitude sur l'information. Elle est représentée par une mesure de probabilité (e.g. $P(crevaison) = 30\%$) ;

— **la théorie des ensembles flous** introduite par (Zadeh, 1965) étend la théorie des ensembles et la logique propositionnelle classique. Contrairement à la théorie des

ensembles, l'appartenance d'un élément à un ensemble n'est pas booléenne. Elle est quantifiée par un degré d'appartenance entre 0 et 1 ;

— **la théorie des possibilités** (Dubois and Prade, 1988; Zadeh, 1978) étend la théorie des ensembles flous en apportant la gestion de l'incertitude épistémique. L'information sur une variable est représentée par deux valeurs, une valeur de possibilité $\Pi(x)$ et une valeur de nécessité $N(x)$;

— **la théorie des fonctions de croyance** (Shafer, 1976) généralise toutes les théories précédentes. Elle est basée sur l'utilisation des fonctions de croyance qui "représentent le degré avec lequel l'ensemble des informations disponibles accrédite l'hypothèse selon laquelle la valeur d'un élément est comprise dans un ensemble, et aucune hypothèse plus spécifique". A chaque hypothèse est associée une masse de croyance m, en particulier à l'ignorance $m(\Omega)$ et au conflit $m(\emptyset)$.

Le cadre formel des fonctions de croyances nous parait pertinent pour la modélisation de la représentation mentale de l'apprenant. En effet, cette théorie répond à tous nos besoins, à savoir la représentation du flou, de l'ignorance et du conflit. Nous utilisons alors ce cadre formel pour représenter le degré de connaissance de l'apprenant des variables pertinentes pour la réalisation de ses tâches.

Formalisation

La représentation mentale de l'apprenant correspond aux valeurs qu'associe l'apprenant aux propriétés des entités du monde. Celles-ci peuvent être différentes des valeurs réelles, notamment si la représentation mentale de l'apprenant n'a pas été mise à jour. Il est donc tout à fait normal que la représentation mentale de l'apprenant soit différente de l'état du monde courant. Rappelons que l'état du monde est décrit par un agrégat d'assertions. Ces derniers sont décrits sous forme de triplets (`sujet prédicat objet`). Par conséquent, nous décrivons la représentation mentale de l'apprenant de la même manière qu'un état du monde, en utilisant un agrégat d'assertions. L'apprenant a besoin de connaître l'ensemble des informations pertinentes pour la réalisation des tâches. En d'autres termes, cela correspond à l'ensemble des préconditions contextuelles des tâches du modèle de tâches. En effet, les préconditions contextuelles sont les conditions qui rendent la réalisation des tâches pertinentes. Par conséquent, l'apprenant doit observer la présence ou non de ces préconditions afin d'évaluer la pertinence de la réalisation des tâches. Nous proposons par conséquent, de n'inclure dans la représentation mentale de l'apprenant que les informations qui sont pertinentes. Autrement dit, la représentation mentale de l'apprenant est constituée d'un agrégat d'assertions qui correspond à l'union des préconditions contextuelles de l'ensemble des tâches du modèle de tâches.

Représentation mentale de l'apprenant

$$RM = \{A_1, A_2, ..., A_n\} = \bigcup_{i=1}^{n} A_i$$

A_i : préconditions contextuelles de la tâche T_i sous forme d'assertions
n : le nombre de tâches du modèle de tâches

Afin de prendre en compte les incertitudes, nous proposons d'enrichir cette représentation par l'utilisation des fonctions de croyances. Un cadre de discernement Ω est un ensemble fini de réponses possibles à une certaine question. Dans notre cas, il correspond à l'ensemble des valeurs possibles d'une variable (e.g. CouleurFeu $\in \Omega$ avec Ω ={Rouge, Orange, Vert})

Cadre de discernement

$$\Omega = \{\omega_1, \omega_2, ..., \omega_K\} = \bigcup_{k=1}^{K} \omega_k$$

La fonction de masse m correspond à une mesure de croyance allouée à une hypothèse et à aucune autre plus spécifique.

Fonction de masse

$$\begin{array}{rcccc} m & : & 2^{\Omega} & \rightarrow & [0,1] \\ & & A & \mapsto & m(A) \end{array}$$

La fonction de masse est définie par :

$$\sum_{A \subseteq \Omega} m(A) = 1 \text{ et } m(\emptyset) = 0$$

— $m(A)$, correspond à la part de croyance alloué à A,
— $m(\Omega)$, correspond au degré d'ignorance totale.

Prenons l'exemple d'un feu de circulation où Ω ={Rouge, Orange, Vert}. Le conducteur étant loin du feu, il a du mal à discerner la couleur active. Toutefois, étant donnée la densité de circulation devant lui, il pense qu'il y a 70% de chance que le feu soit rouge. Par conséquent :

— $m(Rouge)$ = 0.7, correspond à la croyance allouée au fait que le feu soit rouge,
— $m(Vert, Rouge, Orange) = 0.3$, correspond au degré d'ignorance.

Le représentation mentale telle que nous l'avons définie correspond à un agrégat d'assertions sur le monde. L'application des fonctions de croyance dans notre cadre consiste à représenter la véracité et la fausseté de ces assertions. Nous définissons alors le cadre de discernement Ω pour chaque assertion A_i de la représentation mentale :

$$\Omega(A_i) = \{Vrai, Faux\}$$

— *Vrai*, l'hypothèse que l'assertion sur le monde soit vraie ;

— *Faux*, l'hypothèse que l'assertion sur le monde soit fausse.

Nous définissons, par conséquent, les termes suivants pour chaque assertion :

— $v(A_i) = m^{A_i}(Vrai)$ est la masse de croyance sur la véracité de l'assertion A_i ;

— $f(A_i) = m^{A_i}(Faux)$ est la masse de croyance sur la fausseté de l'assertion A_i ;

— $i(A_i) = m^{A_i}(Vrai, Faux)$ est la masse de croyance sur l'ignorance qu'a l'apprenant quant à la véracité ou la fausseté de l'assertion A_i ;

— $c(A_i) = m^{A_i}(\emptyset)$ est le conflit suite aux contradictions entre les sources d'informations concernant l'assertion A_i.

La construction de la représentation mentale de l'apprenant se fait de manière automatique au début de la simulation. Le moteur de génération se charge d'extraire les assertions qui correspondent aux préconditions contextuelles des tâches et les agrège au sein de cette représentation. A l'initialisation, nous supposons que l'apprenant est dans un état d'ignorance totale. Par conséquent, la masse de croyance associée à l'ignorance pour chaque assertion est égale à 1.

Algorithm 6 Initialisation de la représentation mentale de l'apprenant

Require: $Tasks \neq \emptyset$ {Liste des tâches du modèle de tâches}
  $Mental \leftarrow \emptyset$ {Représentation mentale à construire}
  for all $T \in Tasks$ **do**
    for all $p \in T.preconditions$ **do**
      if ISCONTEXTUAL(p) **then**
        $Mental \leftarrow Mental \cup \{p\}$
        $i(Mental.p) \leftarrow 1$
        $v(Mental.p) \leftarrow f(Mental.p) \leftarrow c(Mental.p) \leftarrow 0$
  return $Mental$

3.3.3 Génération de l'ambiguïté de l'information

L'ambiguïté de l'information survient lorsque de multiples interprétations émergent de l'imperfection des informations à disposition. Il s'agit particulièrement de l'incertitude épistémique à caractère subjectif à laquelle l'individu peut remédier. Selon (Blockley, 2013), l'incertitude épistémique est caractérisée par deux dimensions : l'incomplétude et le flou. Dans cette section, nous présentons tout d'abord comment nous générons les états d'incomplétude, en dissimulant les informations pertinentes à la réalisation des tâches. Ensuite, nous expliquons comment nous entachons les informations d'incertitude en spécifiant des objectifs scénaristiques intermédiaires qui vont réduire la fiabilité des sources.

Génération de l'incomplétude

Un des facteurs permettant l'émergence de l'ambiguïté est l'incomplétude des informations qui sont à disposition. La génération de l'ambiguïté passe alors par la génération de situations où certaines informations pertinentes sont volontairement dissimulées afin de laisser place à différentes interprétations. Dans notre contexte, les informations pertinentes se réfèrent aux préconditions des tâches. En particulier, celles qui rendent pertinentes la réalisation de la tâche. Dans ACTIVITY-DL, cela correspond aux préconditions contextuelles. Par exemple, allumer les feux de route n'est pertinent que si nous nous trouvons dans une ville ou dans une chaussée non-éclairées. L'information pertinente, dans ce cas-là, est l'éclairage de la ville ou la chaussée. Prenons un exemple en médecine. Supposons que pour prescrire le médicament-1 (tâche 1), la présence de trois symptômes est nécessaire (symptôme-1, symptôme-2 et symptôme-3). Ces derniers correspondent aux préconditions contextuelles de la tâche 1. D'autre part, pour pouvoir prescrire le médicament 2 (tâche 2), la présence de trois symptômes également est nécessaire (symptôme-1, symptôme-2, symptôme-4). Si à un moment donné, les seuls symptômes connus sont les deux symptômes communs (symptôme-1, symptôme-2), il peut y avoir ambiguïté quant au médicament à prescrire — et donc quant à la tâche à réaliser. En appliquant le cadre formel des fonctions de croyances, cette exemple se représente de la manière suivante :

Masses de croyance associées aux assertions sur les symptômes

A_1 : `(?patient :has-symptom :symptom_1)`
A_2 : `(?patient :has-symptom :symptom_2)`
A_3 : `(?patient :has-symptom :symptom_3)`
A_4 : `(?patient :has-symptom :symptom_4)`
avec $v(A_1) = 1$, $v(A_2) = 1$, $i(A_3) = 1$ et $i(A_4) = 1$

De manière générale, étant donné deux tâches, l'incomplétude se réfère à la connaissance des préconditions communes aux deux tâches et à l'ignorance des préconditions propres à chacune d'elles. Nous formalisons alors l'incomplétude étant données deux tâches quelconques de la manière suivante :

Etat d'incomplétude

Soit T_1 et T_2 deux tâches quelconques, et P_1 et P_2 leur préconditions respectives. Une situation S est considérée en état d'incomplétude si :

$$\begin{cases} v(p_i) \geq \varepsilon_v & \forall p_i \in (P_1 \cap P_2) \\ i(p_j) \geq \varepsilon_i & \forall p_j \in (P_1 \,\Delta\, P_2) \end{cases}$$

avec :
$(P_1 \cap P_2) \neq \emptyset$ et $(P_1 \,\Delta\, P_2) \neq \emptyset$,
$(P_1 \,\Delta\, P_2)$ la différence symétrique entre les deux ensembles de préconditions,
ε_v le seuil à partir duquel une assertion est considérée comme vraie,
ε_i le seuil à partir duquel une assertion est considérée comme ignorée.

L'enjeu du moteur de génération est donc de pouvoir générer des objectifs scénaristiques, qui permettent d'atteindre l'état d'incomplétude défini ci-dessus. Nous définissons alors l'algorithme 7 qui raisonne sur le modèle de tâches et la représentation mentale incertaine de l'apprenant. Il prend en entrée un paramètre ε_v qui correspond à la croyance minimum que doit vérifier une assertion pour qu'elle soit considérée comme vraie. L'algorithme retourne en sortie une liste de groupes d'assertions à dissimuler. L'algorithme parcourt tout d'abord le modèle de tâches par paires. Ensuite pour chaque paire, il vérifie si elles disposent de préconditions communes. Si c'est le cas, il vérifie qu'elles sont toutes perçues par l'apprenant comme étant vraies. C'est à dire que la masse de croyance v est supérieure au seuil ε_v. Lorsque cette condition est vérifiée, l'algorithme extrait les assertions à dissimuler, qui correspondent à la différence symétrique entre les deux ensembles de préconditions. Chaque groupe d'assertions de cette liste correspond donc à un objectif scénaristique à **proscrire**.

Algorithm 7 Recherche de groupes d'assertions à dissimuler

Require: *Tasks* {liste des tâches du modèle de tâches}
Require: *Mental* {Représentation mentale de l'apprenant}
Require: $\varepsilon_v > 0$ {degré de certitude minimum}
  $assertionsList \leftarrow \emptyset$ {Liste des groupes d'assertions à dissimuler}
  for all $t_1 \in Tasks$ **do**
    for all $t_2 \in Tasks$ **do**
      if $(\forall p \in (t_1.preconditions \cap t_2.preconditions), v(Mental.p) \geq \varepsilon_v)$ **then**
        $assertionsList \leftarrow assertionsList \cup \{\{(t_1.preconditions \;\Delta\; t_2.preconditions)\}\}$
  return *assertionsList*

Production du flou

D'après la représentation tridimensionnelle de (Blockley, 2013), l'ambiguïté n'émerge pas uniquement de l'incomplétude mais plutôt d'une combinaison entre l'incomplétude et le flou. Dans la section précédente, nous avons montré comment le moteur de génération extrait les assertions pertinentes à dissimuler. Afin d'accentuer le caractère ambiguë, et en respect de la représentation tridimensionnelle de Blockley, nous proposons de révéler les informations pertinentes en les entachant d'incertitude. Si nous reprenons l'exemple en médecine décrit dans la section précédente, cela se traduit par la fourniture d'informations incertaines sur les symptômes trois et quatre, mettant ainsi le médecin simultanément sur deux pistes plausibles.

Il ne s'agit donc pas uniquement de prescrire un état du monde à atteindre, mais également de spécifier un degré de flou avec lequel cet état doit être perçu par l'apprenant. Par conséquent, nous devons disposer de mécanismes qui permettent de diminuer la fiabilité des sources d'informations, et donc de diminuer les croyances sur la véracité ou la fausseté des assertions. Dans le cadre formel des fonctions de croyances, nous disposons d'un ensemble de règles d'affaiblissement dont celle de (Shafer, 1976) :

Règle d'affaiblissement

$$\begin{aligned} m^{\alpha}(A) &= (1-\alpha) * m(A), \forall A \subset \Omega \\ m^{\alpha}(\Omega) &= (1-\alpha) * m(\Omega) + \alpha \end{aligned}$$

α correspond au degré d'affaiblissement,
(1-α) correspond au degré de fiabilité.

Nous définissons alors la fonction d'affaiblissement f_a qui prend en paramètre un objectif scénaristique (agrégat d'assertions) et un taux d'affaiblissement α. Cette fonction retourne un objectif scénaristique intermédiaire qui permet d'assurer que l'objectif final est instancié avec un taux d'affaiblissement α. Par exemple, supposons que l'information pertinente dissimulée correspond à la présence ou non d'un véhicule derrière celui de l'apprenant. Nous souhaitons révéler partiellement cette information en spécifiant un taux d'affaiblissement de 0.8. Cela peut se faire en mettant en place un temps brumeux qui limite fortement la visibilité :

$$f_a(\text{“LearnerCar BehindObject Car”}, 0.8) = \{\text{“ ?weather :has-state :foggy”}\}$$

Bien évidemment, lorsque le taux d'affaiblissement change, l'état du monde intermédiaire spécifié en sortie change éventuellement. Supposons que, dans l'exemple précédent, nous diminuions le taux d'affaiblissement à 0.2. L'état du monde en sortie, dans ce cas-là, peut être un temps brumeux, toutefois, avec le véhicule derrière ayant allumé les feux de brouillard.

Cela permet effectivement une meilleure visibilité, sans pour autant permettre d'identifier clairement s'il s'agit d'un camion, d'une voiture ou de deux véhicules à deux roues côte-à-côte.

Afin de spécifier ces objectifs intermédiaires, nous proposons pour cela l'ajout d'un nouveau type de barrières au modèle de causalité : les **barrières de perception**. Contrairement aux barrières de prévention ou de protection, leur rôle n'est pas d'empêcher le déclenchement des causes des événements ou leurs conséquences, mais plutôt de limiter leur perception par l'utilisateur de l'environnement. Cela peut se rapprocher de la notion de détectabilité dans certaines représentations de la criticité en analyse des risques. Ces représentations considèrent qu'un événement est plus critique s'il n'est pas détectable par les systèmes ou les opérateurs. Nous associons alors à chaque barrière de perception un taux d'affaiblissement α qui correspond au degré de flou avec lequel l'événement antérieur à la barrière est perçu. Lorsque $\alpha = 1$, l'événement est dissimulé totalement. Lorsque $\alpha = 0$, l'événement est perçu clairement par l'utilisateur de l'environnement. La figure 3.11 illustre un exemple d'une barrière de perception qui limite la perception d'un événement antérieur. Dans cet exemple, le modèle de causalité décrit, entre autres, l'événement où l'apprenant se fait dépasser par un un autre véhicule. Si nous supposons que le seul moyen de percevoir cet événement consiste à regarder dans les rétroviseurs, le fait que ces derniers soient flous (ou cassés) limite considérablement la perception du dépassement.

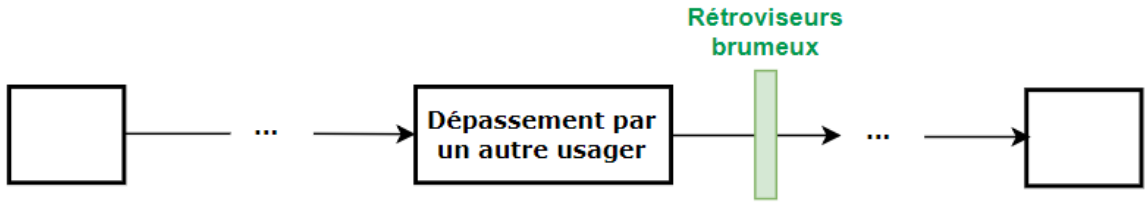

FIGURE 3.11 Exemple de barrière de perception

3.3.4 Génération de l'ambiguïté du contexte

Parmi les sources d'ambiguïté, nous considérons ici le fait que l'individu peut être confronté à des contextes incohérents qui laissent place à différentes interprétations. Il peut s'agir d'une incohérence au niveau même d'une assertion ou d'une incohérence entre différentes assertions entre elles. Le premier cas, se réfère aux situations où l'apprenant dispose de croyances conflictuelles qui ne lui permettent pas de trancher entre la véracité ou la fausseté d'une assertion. Tandis que le deuxième cas désigne les situations où des

assertions vraies impliquent des actions contradictoires. Dans cette section, nous expliquons comment nous générons ces deux types d'incohérence.

Conflit des croyances

S'agissant de l'incohérence d'une assertion, prenons un exemple en médecine. Le patient affirme qu'il n'est pas allergique à une certaine substance. Toutefois, le médecin s'aperçoit que le patient fait une allergie après lui avoir injecté un traitement comportant cette substance. L'ambiguïté dans cette situation naît de l'incohérence entre ce qu'a dit le patient, et ce qui s'est produit en réalité. Cela se représente dans notre cadre formel par le conflit dans la distribution de masses de croyance d'une assertion. Si ce conflit est très faible, il se peut qu'il ne soit pas pris en considération, écartant ainsi toute ambiguïté. Nous définissons alors un seuil ε_c qui définit le degré de conflit minimum qu'une masse de croyance c doit respecter pour qu'une assertion soit potentiellement une source d'ambiguïté. Une assertion A_i est potentiellement ambiguë si elle vérifie l'état de conflit suivant :

$$c(A_i) \geq \varepsilon_c$$

L'enjeu d'un moteur de génération est donc de pouvoir prescrire des objectifs scénaristiques qui permettent d'atteindre l'état de conflit défini ci-dessus, étant donné l'état courant du monde, et la représentation mentale de l'apprenant. Ici, cela se résume à prescrire un objectif scénaristique qui correspond à l'assertion à contredire avec une distribution de masses telle que la fusion des deux distributions entraîne un conflit. Si nous supposons, par exemple, que dans la représentation mentale de l'apprenant, l'assertion (`?patient :is-allergic-to :substance`) est fausse. En d'autres termes, que sa masse de croyance $f(A)$ est supérieure à un seuil ε_f. Le moteur de génération doit être capable de déduire l'état du monde comprenant la même assertion mais avec une distribution de masses qui maximise le conflit en cas de fusion. Dans cet exemple, de manière intuitive, l'état du monde propice à la génération du conflit correspond à une situation où l'assertion (`?patient :is-allergic-to :substance`) dispose d'une distribution de masses qui maximise $v(A)$.

Nous définissons alors l'algorithme 8 qui raisonne sur le modèle de causalité. Il prend en entrée un seuil ε_c qui correspond au seuil minimal du conflit à atteindre. L'algorithme retourne en sortie l'ensemble des événements dont les postconditions correspondent à des assertions qui sont présentes dans la représentation mentale de l'apprenant et dont la fusion de leur distribution de masses (fusion de la distribution d'une assertion qui correspond à une postcondition d'un événement et une assertion identique dans la représentation mentale) engendre un conflit supérieur à ε_c. Pour cela l'algorithme parcourt l'ensemble des assertions

de la représentation mentale de l'apprenant. Pour toutes les assertions dont la masse de croyance liée au conflit $c(A)$ est inférieure au seuil de conflit minimal ε_c, l'algorithme recherche les événements dont les postconditions incluent ces assertions. Ensuite, si la fusion des distributions de masse de croyance des deux assertions (celle de la représentation mentale et celle des postconditions de l'événement) produit une nouvelle distribution de masse dont le conflit est supérieur à ε_c, l'événement est préservé, sinon il est rejeté.

Algorithm 8 Recherche d'événement maximisant le conflit

Require: $Events \neq \emptyset$ {liste des événements du modèle de causalité}
Require: $Mental \neq \emptyset$ {Représentation mentale}
Require: $\varepsilon_c \neq 0$ {degré de conflit minimum}
  $eventList \leftarrow \emptyset$ {Liste des événements susceptibles de créer du conflit}
  for all $a \in Mental$ **do**
    if $c(a) \leq \varepsilon_c$ **then**
      for all $e \in Causality$ **do**
        for all $p \in e.postconditions$ **do**
          if $p = a$ **then**
            if $(m_p \bigoplus m_a) \Rightarrow c_{p \bigoplus a} \geq \varepsilon_c$ **then**
              $eventList \leftarrow eventList \cup \{e\}$
  return $eventList$

Incohérence des assertions

L'ambiguïté du contexte peut émerger également à cause de la présence de deux contextes incompatibles, nécessitant des actions contradictoires. Prenons par exemple le cas d'un feu de circulation défaillant, où les feux rouge et vert sont allumés simultanément. Si l'apprenant considère que le feu est vert, il roule. S'il considère qu'il est rouge, il s'arrête. Cependant, il ne peut pas considérer les deux contextes à la fois car ils sont incompatibles. Ici, les informations à disposition sont claires (feu rouge et feu vert). En d'autres termes, la masse de croyance sur la véracité de chaque assertion $v(A_i)$ est supérieure à un seuil ε_v. Une situation est alors ambiguë si, étant donnée deux tâches contradictoires, les préconditions pertinentes à leur réalisation sont vérifiées, et sont perçues comme l'étant ainsi dans la représentation mentale de l'apprenant. Nous définissons alors l'ambiguïté qui émerge par l'incohérence des assertions comme suit :

Etat d'incohérence des assertions

Soit P_1 et P_2 les deux ensembles de préconditions respectifs des tâches T_1 et T_2.
S_{T_1,T_2} est une situation ambiguë si :

$$v(P_1) \geq \varepsilon_v \ , \quad v(P_2) \geq \varepsilon_v \quad \text{et} \quad T_1 \wedge T_2 \Rightarrow \bot$$

avec ε_v, le degré de certitude minimum.

L'enjeu d'un moteur de génération est donc de pouvoir générer des objectifs scénaristiques permettant d'atteindre l'état d'incohérence défini ci-dessus. C'est-à-dire, si nous supposons que dans la représentation mentale de l'apprenant, l'assertion (`trafficLight has-color green`) est vérifiée, le moteur doit pouvoir déduire que pour produire une situation où deux contextes incompatibles sont présents, il doit prescrire l'objectif scénaristique : (`trafficLight has-color red`), car celui-ci implique la réalisation d'une action contradictoire à celle associée à l'assertion de la représentation mentale. Nous définissons alors l'algorithme 9, qui raisonne sur la représentation mentale de l'apprenant et le modèle de tâches. L'algorithme prend en entrée un seuil ε_v qui représente le degré minimum de certitude pour qu'une assertion soit considérée comme vraie. L'algorithme retourne en sortie une liste de groupes d'assertions incohérents avec des assertions du modèle mental. Pour cela, l'algorithme recherche tout d'abord l'ensemble des groupes d'assertions vraies dans la représentation mentale de l'apprenant (un groupe désigne un ensemble d'assertions constituant les préconditions d'une tâche). En d'autres termes, l'algorithme recherche les assertions dont la masse de croyance associée à leur véracité est supérieure au seuil minimal de certitude ε_v. Ensuite, pour chaque tâche ayant comme préconditions un des groupes d'assertions identifiés précédemment, l'algorithme recherche les tâches qui lui sont contradictoires. Il extrait ensuite les préconditions contextuelles de ces tâches, et les ajoute à la liste des groupes d'assertions retournées à la fin de l'algorithme. Chaque groupe d'assertions correspond à un contexte incohérent avec la représentation mentale courante de l'apprenant. C'est également un objectif scénaristique à prescrire qui est transmis par le moteur de génération au moteur de planification qui se chargera de diriger la simulation vers un état du monde qui respecte cet objectif.

3.3.5 Bilan

Nous avons proposé un ensemble de techniques et d'algorithmes qui raisonnent sur des modèles de connaissances afin de faire émerger l'ambiguïté. Nous avons tout d'abord proposé un modèle pour décrire la représentation mentale de l'apprenant. Cela permet de se rendre

Algorithm 9 Recherche d'assertions incompatibles

Require: $Tasks \neq \emptyset$ {liste des tâches du modèle de tâches}
Require: $Mental \neq \emptyset$ {Représentation mentale de l'apprenant}
Require: $\varepsilon_v \neq 0$ {degré de certitude minimum}
  $AssertionsList \leftarrow \emptyset$ {Liste de groupes assertions susceptibles de créer une incohérence}
  for all $A \in Mental$ **do**
    if $v(A) \geq \varepsilon_v$ **then**
      for all $T_1 \in Tasks$ **with** $T_1.precondition = A$ **do**
        for all $T_2 \in Tasks$ **do**
          if AREContradictory(T_1, T_2) **then**
            if $\{T_2.preconditions\} \notin AssertionsList$ **then**
              $AssertionsList \leftarrow AssertionsList \cup \{\{T_2.preconditions\}\}$
  return $AssertionsList$

compte du degré de connaissance qu'a l'apprenant sur l'ensemble des variables du monde. Cette représentation est construite particulièrement à partir des assertions pertinentes pour la réalisation des tâches. Afin d'enrichir cette représentation, nous avons proposé d'utiliser le cadre formel des fonctions de croyance qui permet de prendre en compte l'incertitude mais aussi de quantifier le conflit résultant de la fusion d'informations contradictoires. Pour la génération de l'ambiguïté, et en nous basant sur la représentation mentale incertaine de l'apprenant, nous avons tout d'abord proposé de générer des situations caractérisées par l'incomplétude, en dissimulant les informations pertinentes à la réalisation des tâches. L'utilisation des modèles sémantiques, en particulier le modèle de tâches, permet d'extraire ces informations pertinentes. Par ailleurs, nous avons proposé d'enrichir le modèle de causalité par la notion de barrières de perception. Cette notion permet d'identifier les éléments qui sont susceptibles de limiter la perception des événements déclenchés. Cela permet donc de diminuer le degré de fiabilité des sources et donc de tâcher les informations par du flou. La production de ces deux aspects — le flou et l'incomplétude — permet de générer un type spécifique d'ambiguïté : l'ambiguïté de l'information. Dans le cadre de nos travaux, nous nous sommes intéressés également à l'ambiguïté du contexte. Nous avons proposé de la générer en produisant des contextes incohérents, que ce soit en amplifiant le conflit dans les croyances liées aux assertions, ou en provoquant des événements mettant en jeu des assertions incompatibles avec la représentation mentale de l'apprenant. L'utilisation des modèles sémantiques, permet d'inférer des connaissances nouvelles, comme par exemple les relations de contradiction entre les tâches. Cela permet d'identifier les éléments qui sont susceptibles d'engendrer du conflit et/ou de produire des contextes incompatibles.

Chapitre 4

Implémentation

Sommaire

4.1 Moteur de génération de dilemmes 111
4.1.1 Les modèles de connaissances 111
4.1.2 Le moteur de génération KOBA 113
4.1.3 Le profil moral 114
4.2 Preuve de concept 114
4.2.1 Présentation de l'environnement virtuel 114
4.2.2 Les modèles utilisés 115
4.2.3 Les étapes de génération 115

4.1 Moteur de génération de dilemmes

Nos travaux sur la génération dynamique des dilemmes ont fait l'objet d'un démonstrateur en environnement virtuel afin de valider les capacités génératives de KOBA.

4.1.1 Les modèles de connaissances

Afin de faciliter l'interprétation des modèles de connaissances par le moteur de génération, il a été nécessaire de développer un modèle objet JAVA pour les modèles qui n'étaient pas codés dans ce langage, à savoir, le modèle de causalité et le modèle d'activité. Nous avons opté pour une représentation en JSON qui permet un parsing et une importation aisées grâce aux libraires disponibles. Les représentations JSON de ces deux modèles respectent bien évidemment les méta-modèles ACTIVITY-DL et CAUSALITY-DL. Nous avons alors

développé une classe générique `ModelLoader`, qui permet de charger un modèle quelconque à partir d'une représentation JSON.

Modèle d'activité

Afin de faciliter la manipulation du modèle d'activité, nous avons développé la classe ACTIVITYMODEL, qui en plus de permettre l'accès aux différents composants du modèle, procure un ensemble de méthodes nécessaires à l'extraction d'information pour la génération des dilemmes. Parmi les principales fonctions de cette classe, nous trouvons :

- `isContradictoryWith(Task task)` : cette méthode retourne si oui ou non la tâche à partir de laquelle la méthode est invoquée et la tâche donnée en argument sont contradictoires, en vérifiant leurs postconditions. Cette fonction est utilisée par le moteur de génération pour identifier les paires de tâches qui peuvent potentiellement servir pour la génération d'un dilemme d'obligation.
- `isCompatibleWith(Task task)` : cette méthode retourne si oui ou non la tâche à partir de laquelle la méthode est invoquée et la tâche donnée en argument sont contextuellement compatibles, en vérifiant leurs préconditions. Cette méthode est utilisée par le moteur de génération pour écarter les paires de tâches qui ne peuvent être instanciées dans une situation qui les met en jeu simultanément.
- `commonAncestor(TaskNode taskNode, TaskNode taskNode-2)` : cette méthode retourne le premier ancêtre commun des deux tâches dont les nœuds sont donnés en paramètres. Cette méthode est utilisée par le moteur de génération pour écarter les paires de tâches dont l'ancêtre commun contient un constructeur temporel séquentiel.

Modèle de causalité

Afin de faciliter la manipulation du modèle de causalité, à l'instar du modèle d'activité, nous avons développé la classe CAUSALITYMODEL, qui en plus de permettre l'accès aux différents composants du modèle, procure un ensemble de méthodes utilisées par le moteur de génération pour extraire les informations pertinentes pour la génération de dilemmes. Parmi les principales fonctions de cette classe, nous trouvons :

- `isNegativeConsequenceNode(CausalityNode node)` : cette méthode retourne si oui ou non le nœud du graphe de causalité donné en paramètre est considéré comme une conséquence négative selon la formalisation de la fonction $f_{cn}(e)$ donnée dans la section 3.2.1.
- `hasNegativeConsequence(CausalityNode node)` : cette méthode retourne si oui ou non le nœud du graphe de causalité donné en paramètre conduit à des conséquences

négatives. Elle vérifie notamment s'il existe un chemin entre ce nœud et un nœud de conséquences négatives identifié par la fonction précédente. Cette méthode est utilisée par le moteur de génération au tout début du processus de génération pour lister l'ensemble des actions, barrières et événements entraînant des conséquences négatives, et qui sont donc susceptibles d'être utilisés pour la génération de dilemmes.

— `commonDescendant(CausalityNode node, CausalityNode node-2)` : cette méthode retourne le premier descendant commun des deux nœuds donnés en paramètre. Elle est utilisée par le moteur de génération pour identifier les paires d'actions ou de barrières liées par une porte logique AND. Cela permet de déduire les paires d'actions qui conduisent à des conséquences négatives si elles sont réalisées ensemble, et les paires de barrières qui conduisent à des conséquences négatives si elles ne sont pas réalisées ensemble.

Modèle du monde

Le modèle du monde est géré par le WORLDMANAGER. Ce dernier, étant codé en JAVA, nous n'avions pas eu à réaliser de développements particuliers afin de pouvoir manipuler le modèle du monde. En effet, il suffisait d'envoyer à ce module les requêtes adéquates afin d'extraire les éléments nécessaires à la génération des dilemmes.

4.1.2 Le moteur de génération KOBA

KOBA a été développé en utilisant le langage JAVA. La modularité de la plateforme HUMANS a permis de faciliter son intégration au sein de la plateforme. En effet, la communication entre les différentes modules se faisant à travers des messages réseaux, cela a permis de faire abstraction des langages utilisés pour chaque module. S'agissant de la génération de dilemmes, nous avons implémenté la classe `DilemmaManager`. Cette classe fait principalement appel aux méthodes décrites précédemment. Ces méthodes permettent de construire au final deux listes de paires d'actions susceptibles de créer des dilemmes d'obligation ou des dilemmes de prohibition. Une fois que ces listes sont à la disposition du moteur de génération, ce dernier les classe en fonction d'un score qui correspond au degré de satisfaisabilité des contraintes pédagogiques et scénaristiques définies dans la section 3.2.4. Parmi les principales méthodes de la classe `DilemmaManager`, nous trouvons :

— `computeScore()` : cette méthode prend en paramètre une paire d'actions `Pair<Task>`, un ensemble de contraintes `Constraints`, et les poids attribués à ces contraintes `List<float>`. Elle retourne le score attribué à cette paire selon le degré de satisfaisabilité des contraintes, en respectant les formalisations définies dans 3.2.4. Cette

méthode est utilisée par le moteur de génération pour classer les paires de tâches afin de pouvoir proposer celle qui répond au mieux aux différents types de contraintes.

— `prescribeScenaristicGoals(Pair<Task>)` : cette méthode prend en paramètre une paire d'actions sélectionnée et extrait l'objectif scénaristique propice à la réalisation de ces deux actions.

— `proscribeScenaristicGoals(Pair<Task>)` : cette méthode prend en paramètre une paire d'actions sélectionnée et extrait l'objectif scénaristique à proscrire pour assurer un état du monde propice à la réalisation de ces deux actions.

4.1.3 Le profil moral

Pour la modélisation du profil moral de l'apprenant, nous avons opté pour l'utilisation de la théorie des valeurs universelles de (Schwartz, 2006). Cette théorie définit une taxonomie de valeurs dans une structure circulaire qui renseigne sur les relations de compatibilité et d'opposition qu'elles ont les unes avec les autres. Nous avons opérationnalisé cette théorie dans la classe `MoralProfil`. Pour chaque objet `Value`, nous avons défini les propriétés suivantes :

— `label` : correspond au nom de la valeur ;

— `dimension` : détermine la dimension à laquelle appartient la valeur ;

— `importance` : détermine l'importance qu'a la valeur pour l'apprenant.

La classe dispose également de la méthode `propagateImportance()` qui permet de propager l'importance des valeurs selon le mécanisme de propagation décrit dans la section 3.2.2.

4.2 Preuve de concept

Nos travaux sur la génération dynamique de dilemmes ont fait l'objet d'une preuve de concept en conduite automobile. Nous avons pour cela utilisé une première version du moteur de génération et développé un environnement virtuel qui communique directement avec ce moteur sans passer par le moteur de planification.

4.2.1 Présentation de l'environnement virtuel

L'environnement consiste en une ville virtuelle composée de bâtiments, panneaux de signalisation, de feux de circulation et d'agents autonomes — véhicules et piétons (cf. figure 4.1). Le joueur contrôle un véhicule en première personne en utilisant une configuration clavier-souris ou en utilisant un kit volant-pédale adapté. L'environnement a été développé

en utilisant le moteur de rendu Unity 3D. Les scripts développés dans le cadre de ce démonstrateur ont tous été codés en C#. Afin de pouvoir déclencher des événements par des directives du moteur de génération, ces événements doivent tout d'abord être implémentés visuellement du côté de l'environnement virtuel. A l'initialisation de la scène, tous les événements déclenchables sont dans un état `inactif`. Dans le cadre de ce démonstrateur, nous avons développé les événements suivants : `début de pluie`, `aquaplaning du véhicule du joueur` et `changement de la couleur du feu`.

FIGURE 4.1 Capture écran du démonstrateur

4.2.2 Les modèles utilisés

Pour ce démonstrateur, nous avons utilisé le modèle de tâches illustré dans la figure 4.2. Le modèle est volontairement simplifié pour des soucis de clarté. Il décrit trois tâches indépendantes : "Handle_aquaplaning", "Handle_red_light" et "Handle_stop".

Nous avons également utilisé le modèle de causalité illustré dans la Figure 4.3. Il décrit un ensemble d'événements comme par exemple "Running a Stop Sign" et "Running a Red Light" qui conduisent à un "Highway Code Violation". Ce dernier peut aussi être causé par les nœuds "Driving" et "Answer a phone call".

4.2.3 Les étapes de génération

Nous détaillons ici pas à pas, les différentes étapes du processus de génération illustrées plus tôt dans le mémoire dans la figure 3.6 :

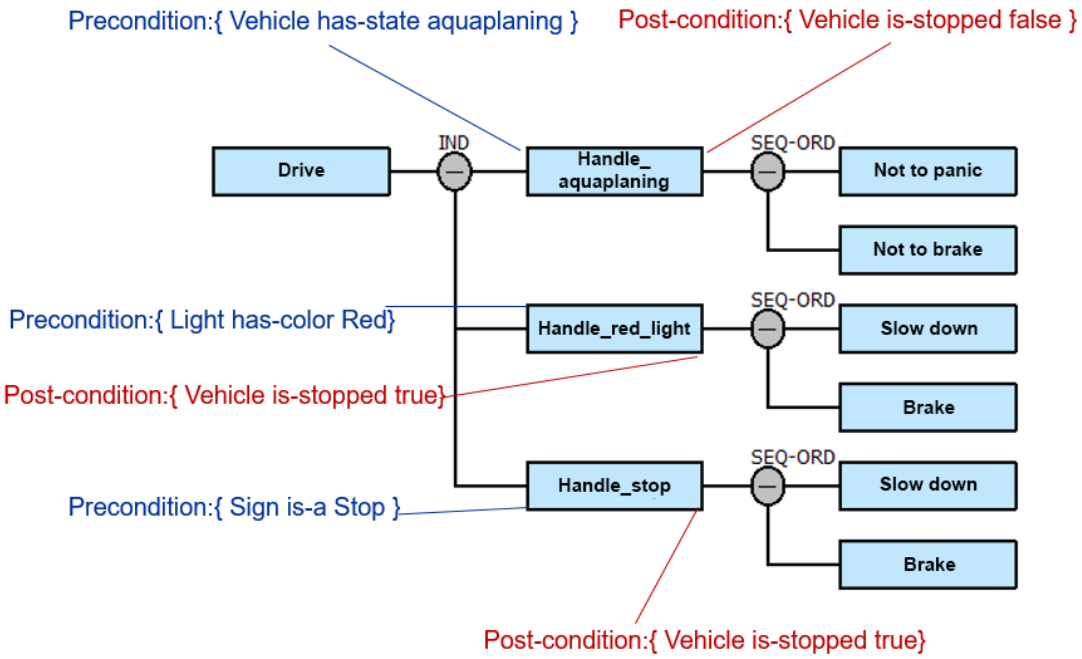

FIGURE 4.2 Modèle de tâches utilisé pour le démonstrateur

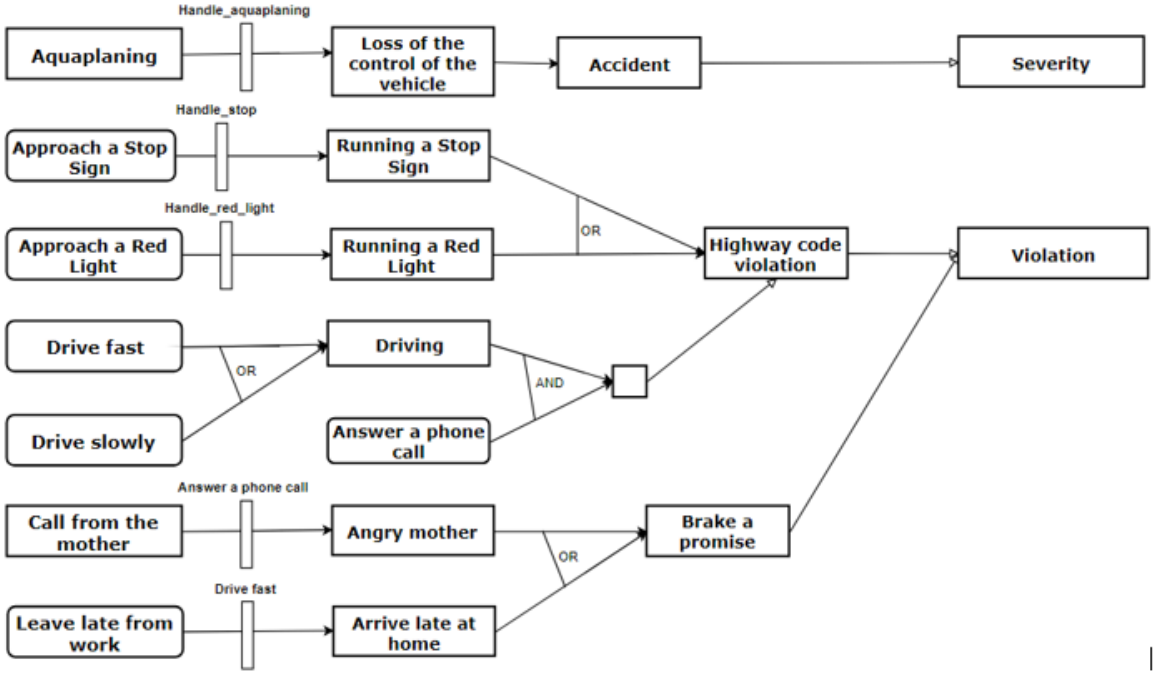

FIGURE 4.3 Modèle de causalité utilisé pour le démonstrateur

Etape 1.1 : recherche de barrières à conséquences négatives (dilemmes d'obligation)
D'après le modèle de causalité, les barrières qui, si elles ne sont pas activées, pourraient mener

à des conséquences négatives sont : “Handle_stop”, “Handle_red_light”, “Handle_aquaplaning”, “Answer a phone call” et “Drive fast”.

Etape 1.2 : recherche d’actions à conséquences négatives (dilemmes de prohibition)
D’après le modèle de causalité, les actions qui pourraient mener à des conséquences négatives sont : “Approach a Stop sign”, “Approach a Red light”, “Drive fast”, “Drive slowly”, “Answer a phone call” et “leave late from work”.

Etape 2.1 : recherche de paires d’actions contradictoires
En parcourant la liste retournée à la fin de l’étape 1.1, le moteur de génération recherche les actions contradictoires en s’appuyant sur le modèle de tâches. Il retourne les paires suivantes :

— `paire 1` : {“Handle_stop”, “Handle_aquaplaning”} ;
— `paire 2` : {“Handle_red_light”, “Handle_aquaplaning”}.

En effet, les postconditions des actions, pour les deux paires, sont incompatibles ("Vehicle is-stopped false" vs "Vehicle is-stopped true"). La paire {"Handle_stop", "Handle_red_light"} est rejetée car les postconditions sont compatibles ("Vehicle is-stopped true" vs "Vehicle is-stopped true").

Etape 2.3 : recherche de paires de barrières à conséquences négatives
A partir de la liste retournée à la fin de l’étape 1.2, le moteur de génération cherche, par paires, les barrières qui conduisent à des conséquences négatives si elles sont activées toutes les deux. Dans cet exemple, le moteur ne retourne aucune paire car il n’existe pas de barrières qui ont une porte logique AND comme descendant commun.

Etape 3 : vérification de la compatibilité d’instanciation
L’ancêtre commun des actions de la `paire 1` est l’action "Drive" qui dispose d’un constructeur temporel indépendant `IND`. C’est le cas également pour les actions de la `paire 2`. Par ailleurs, les préconditions des actions de `paire 1` sont compatibles ("Sign is-a Stop" vs "Vehicle has-state aquaplaning"). Pareillement pour `paire 2`, les préconditions de ses actions sont compatibles. Par conséquent, `paire 1` et `paire 2` sont toutes les deux retenues car leurs actions sont contextuellement et temporellement compatibles.

Etape 4 : Classement et sélection d’une paire d’actions
A ce stade, deux paires sont candidates. Les dilemmes issus de ces deux paires sont les suivants :

— `paire 1` : dans une situation d'aquaplaning, le conducteur doit-il freiner pour respecter le stop au risque de perdre le contrôle de son véhicule, ou doit-il éviter de freiner en grillant ainsi le stop, afin de garder le contrôle du véhicule.
— `paire 2` : dans une situation d'aquaplaning, le conducteur doit-il freiner pour respecter le feu rouge au risque de perdre le contrôle de son véhicule, ou doit-il éviter de freiner en grillant ainsi le feu, afin de garder le contrôle du véhicule.

Dans cette version du démonstrateur, le calcul des scores des paires d'actions n'avait pas été implémenté. La sélection de la meilleure paire s'est faite de manière aléatoire. Dans une version plus aboutie qui prendrait en compte les contraintes pédagogiques et scénaristiques, si on suppose qu'il y a dix fois plus de feux rouges que de panneaux stop, le moteur de génération pourrait préférer `paire 2` à `paire 1` car il y aurait plus de chances qu'une situation mettant en jeu les actions de `paire 2` se produise. Pour la suite de cet exemple, nous supposons alors que `paire 2` a été choisie.

Etape 5 : Extraction d'un état du monde

Finalement, le moteur de génération extrait un état du monde à partir de la paire d'action sélectionnée. Cet état consiste en une agrégation des préconditions de ces actions. Pour `paire 2`, cela correspond à :

{(Vehicle has-state aquaplaning) AND (Light has-color Red)}

Ces préconditions, qui correspondent à un état but à atteindre, sont transmises en théorie au moteur de planification qui se charge de diriger la simulation vers une situation où se trouve un feu rouge et où le véhicule du conducteur est en aquaplaning. L'environnement virtuel se charge ensuite d'instancier visuellement cette situation [1]. Le moteur de planification ne faisant pas l'objet de nos travaux, dans le cadre du démonstrateur, nous avons fait abstraction de ce moteur en transmettant les états du monde à atteindre directement à l'environnement virtuel.

1. Une vidéo de la situation dilemmatique peut être visionnée en suivant ce lien : https ://www.youtube.com/watch ?v=Qz80sBjasfU

Chapitre 5

Évaluations

Sommaire

5.1 Évaluation préliminaire 119

5.2 Évaluation en environnement virtuel 121

5.2.1 Protocole de l'évaluation 122

5.2.2 Résultats 124

5.2.3 Interprétation des résultats 126

Pour valider nos travaux sur la génération dynamique des dilemmes, nous avons réalisé deux évaluations. Nous avons conduit, dans un premier temps, une évaluation préliminaire sans environnement virtuel pour évaluer les capacités d'une première version de notre moteur de génération. L'objectif de cette évaluation consistait à comparer les situations dilemmatiques générées aux situations non-dilemmatiques scriptées. Nous avons réalisé ensuite une seconde expérience, cette fois-ci en salle de réalité virtuelle (CAVE [1]). Avec cette expérience, nous souhaitions ainsi valider que KOBA était capable de générer dynamiquement à partir des modèles de connaissances des dilemmes perçus comme tels, sans les avoir écrits à l'avance. Dans ce chapitre, nous présentons les protocoles de ces deux évaluations, et exposons les résultats issus de ces deux expériences.

5.1 Évaluation préliminaire

Afin d'évaluer notre moteur de génération de dilemmes, nous avons mené une évaluation préliminaire sans environnement virtuel. Par cette évaluation, nous souhaitons savoir si les situations générées sont dilemmatiques. Nous proposons alors de confronter les participants

1. Un CAVE (Cave Automatic Virtual Environment) ou Cube immersif 3D, est une salle de visualisation 3D composée de plusieurs écrans et de capteurs de mouvements

à un ensemble de situations qui contient à la fois des situations dilemmatiques générées par notre moteur, et des situations normales – non-dilemmatiques – non générées. Ne voulant pas induire tout de suite l'idée de dilemme, nous cherchons à savoir si les participants identifient dans les situations générées, les propriétés que nous avons associées aux dilemmes dans notre définition. Nous cherchons donc à vérifier l'hypothèse suivante :

— H1 : *Les participants retrouvent dans les situations générées les propriétés des situations dilemmatiques*

L'évaluation consiste en un questionnaire en ligne où les participants doivent répondre à un ensemble de questions par rapport à différentes situations de conduite. Le questionnaire porte sur neuf situations de conduite automobile décrites par deux phrases au maximum. Parmi ces situations, six d'entre elles correspondent à des situations générées par notre système. Elles mettent en jeu les paires d'actions suivantes :

— {"Respecter Stop", "Gérer Aquaplanage"}
— {"Respecter Feu rouge", "Gérer Aquaplanage"}
— {"Gérer Voiture qui colle", "Respecter Stop"}
— {"Gérer Voiture qui colle", "Gérer Traversée Piéton"}
— {"Gérer Voiture qui colle", "Respecter Feu Rouge"}
— {"Gérer Aquaplanage", "Gérer Traversée Piéton"}

Les trois autres paires d'actions correspondent à des situations dites normales (non-dilemmatiques) :

— {"Respecter Feu Rouge", "Gérer Route Dégagée"}
— {"Respecter Sens Interdit", "Suivre Conseil du Passager"}
— {"Gérer Feu Vert", "Gérer Traversée d'un Piéton"}

Les neuf situations sont présentées dans un ordre aléatoire. Pour chacune d'elles, les participants doivent répondre aux questions suivantes :

— Que feriez-vous dans cette situation ?
— Avez-vous hésité avant de prendre votre décision ? Pourquoi ?
— Pensez-vous qu'il existe une solution sans conséquences négatives ?
— Pensez-vous qu'il existe une bonne solution ?

Nous avons eu au total, 67 participants. La figure 5.1 présente les résultats recueillis.

Nous constatons que les participants sont plus hésitants dans les situations générées que dans les situations normales. Cependant, nous nous attendions à une valeur d'hésitation plus élevée. En analysant les réponses des participants, il s'avère que cela est dû à quelques éléments qui n'étaient pas pris en compte dans les modèles, ce qui a permis à certaines décisions d'être plus ou moins simples et immédiates (e.g. "la plupart des véhicules de nos jours sont équipés de technologies qui font éviter les pertes de contrôle lors d'aquaplaning. Donc, devant un feu rouge je freine"). Dans les situations générées, 60% des participants

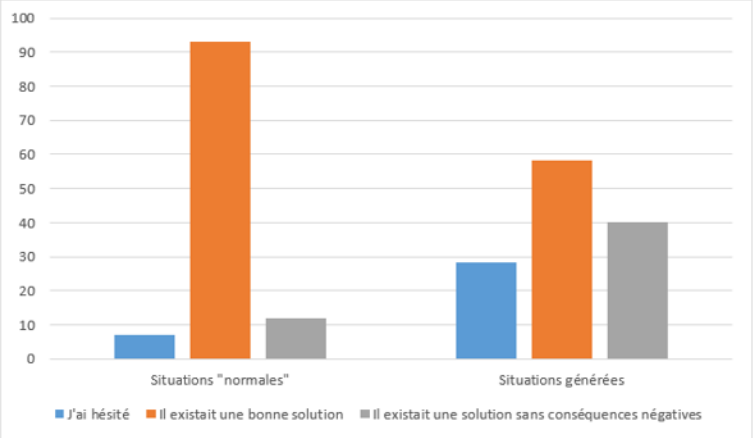

FIGURE 5.1 Résultat de l'évaluation préliminaire

estiment qu'il n'y a pas de solution sans conséquences négatives, tandis que pour les situations normales 12% seulement estiment cela. Par ailleurs, 93% des participants déclarent qu'il existe une bonne solution lorsqu'il s'agit de situations normales contre 58.25% lorsqu'il s'agit de situations générées. Par conséquent, nous pouvons en déduire que notre système a pu générer des situations qui, comparées à des situations normales, sont plus compliquées en termes de prise de décision, sont perçues comme ayant des conséquences négatives quel que soit le choix et ne présentent pas de bonne solution, ce qui correspond aux situations dilemmatiques.

5.2 Évaluation en environnement virtuel

Afin de valider les capacités de notre moteur de génération, il a été nécessaire de réaliser une expérience en environnement virtuel afin de déterminer si les situations générées par notre moteur, et instanciées visuellement, sont bien perçues comme étant des dilemmes. Pour cela, nous avons conduit une expérience en environnement virtuel qui consiste à confronter les participants à des situations de différentes natures. Nous souhaitons voir si au milieu de ces situations, les participants arrivent à percevoir celles qui sont générées comme étant des dilemmes. Plus précisément, nous cherchons à savoir si les situations dilemmatiques générées sont perçues de la même manière que les situations dilemmatiques scriptées. Nous cherchons donc à vérifier l'hypothèse générale suivante : *notre moteur de génération génère bien des dilemmes*. Pour la vérifier, nous proposons de la diviser en trois sous-hypothèses :

— H0.1 : *les individus perçoivent qu'ils ont le choix entre deux options dans les deux conditions (générée et scriptée)*

— H0.2 : *les individus perçoivent qu'il n'y a pas d'option sans conséquences négatives dans les deux conditions (générée et scriptée)*
— H0.3 : *les individus considèrent qu'ils ont été confrontés à un dilemme dans les deux conditions (générée et scriptée)*

Afin de maximiser l'immersion, nous avons conduit cette expérience dans une salle de réalité virtuelle composée de trois grands écrans. Au centre de la salle nous avons installé un siège volant-pédales utilisé pour piloter la voiture virtuelle (cf. figure 5.2).

FIGURE 5.2 Salle de réalité virtuelle

5.2.1 Protocole de l'évaluation

Les participants à cette évaluation sont au volant d'une voiture automatique. Les seuls fonctionnalités dont ils ont besoin sont l'accélérateur, le frein et le volant. Au début de l'expérience, les participants jouent dans deux scénarios préliminaires afin de se familiariser avec l'environnement virtuel et le volant-pédales. Dans le premier scénario de familiarisation, les participants doivent récupérer une personne qui les attend au bout de la rue. C'est une route urbaine où la vitesse est limitée à 50 km/h. La présence d'un stop lors du trajet permet aux participants de se familiariser avec les freins. Par ailleurs, à un moment donné, ils doivent éviter un obstacle en se plaçant sur la voie du sens contraire. Le deuxième scénario de familiarisation reprend les mêmes éléments du premier scénario en ajoutant un événement particulier qui est le passage d'un piéton lorsque la voiture du participant est à l'approche. Par cette expérience, nous cherchons à savoir si une situation générée par notre moteur de scénarisation est perçue de la même manière qu'une situation dilemmatique écrite manuellement. Nous avons donc proposé une expérience où les participants sont confrontés à quatre scénarios : un scénario dilemmatique généré, un scénario dilemmatique scripté, un scénario critique (sans dilemme), et un scénario non-critique. Les deux derniers scénarios

servent à camoufler plus ou moins les scénarios dilemmatiques. Au début de chaque scénario, un objectif est énoncé au participant. Nous détaillons ci-après ces quatre scénarios :

- **scénario non-critique :** l'objectif du participant dans ce scénario est d'aller chercher une collègue qui l'attend à côté du premier stop. L'événement déclenché durant ce scénario correspond à un piéton qui traverse normalement en utilisant le passage prévu à cet effet. Le participant a le temps de voir le piéton et de freiner sans urgence même en étant en excès de vitesse ;
- **scénario critique :** dans ce scénario, l'objectif du participant est d'aller chercher une collègue qui l'attend juste avant le premier stop. L'événement déclenché durant ce scénario correspond à un piéton qui surgit subitement derrière une cabine téléphonique. Si le participant ne freine pas à temps, il risque d'y avoir collision. Par ailleurs, si le participant est en excès de vitesse, même en réalisant un freinage d'urgence, la collision est inévitable ;
- **scénario du dilemme généré :** dans ce scénario, l'objectif du participant est d'aller récupérer une collègue qui l'attend au bout de la rue. Dans ce scénario, le participant se trouve dans une situation de collision inévitable. Il doit choisir entre percuter un piéton qui traverse ou des éléments du décor ;
- **scénario du dilemme scripté :** dans ce scénario, l'objectif du participant est de suivre la voiture d'un ami car ce dernier est le seul à connaître le lieu du rendez-vous. Lors du trajet, l'ami grille un sens interdit. Le participant doit choisir entre respecter le code de la route et suivre son ami.

Les scénarios sont présentés dans un ordre aléatoire. Ils se terminent si le participant réalise l'objectif demandé ou s'il y a eu collision (des captures d'écran des scénarios sont présentées dans l'annexe C). A la fin de chaque scénario, le participant répond à une série de questions identiques pour les quatre scénarios :

- *Q1.1. Y a-t-il eu une collision ?*
  Oui / Non
- *Q1.2. Y a-t-il eu une violation du code de la route ?*
  Oui / Non
- *Q1.3. Avez-vous échoué dans votre objectif ?*
  Oui / Non
- *Q2. Juste avant l'incident (collision, violation du code de la route ou échec dans la réalisation de l'objectif), j'avais le choix entre au moins deux options.*
  (1) pas du tout d'accord … (7) tout à fait d'accord

— *Q3. Il existait une option sans conséquences négatives (sans collision, sans violation du code de la route et sans échouer dans mon objectif).*
(1) pas du tout d'accord ... (7) tout à fait d'accord
— *Q4. J'ai été confronté à un choix difficile*
(1) pas du tout d'accord ... (7) tout à fait d'accord
— *Q5. J'ai été face à un dilemme*
(1) pas du tout d'accord ... (7) tout à fait d'accord
— *Q6. J'ai hésité avant d'agir*
(1) pas du tout d'accord ... (7) tout à fait d'accord
— *Q7. Si c'était à refaire, j'aurais agi différemment*
(1) pas du tout d'accord ... (7) tout à fait d'accord

Si les participants répondent "non" aux trois premières questions, ils vont directement à la question 6.

Notre objectif est de voir si chaque participant perçoit le dilemme généré de la même manière que le dilemme scripté. Nous nous sommes intéressés particulièrement aux réponses des questions 2, 3 et 5. L'objectif des questions 2 et 3, est de savoir si dans l'absolu, le moteur de génération a réussi à produire une situation qui respecte les propriétés d'un dilemme. La question 5, quant à elle, est plus subjective. Son but est de collecter le ressenti des participants lorsqu'ils sont confrontés à une telle situation.

5.2.2 Résultats

A l'écriture de ces lignes, nous avons eu 20 participants. Les figures 5.3 et 5.4 représentent respectivement les résultats associés au scénario généré et au scénario scripté. Nous notons que pour le dilemme scripté, nous n'avons eu que 17 réponses parmi 20 pour ces trois questions. Il s'avère que trois participants ont répondu "non" aux trois premières questions. Selon eux, il n'y avait pas de violation du code de la route. Cela les exonère donc des questions 2,3,4 et 5. Lors d'une brève phase de débriefing post-expérience, ces participants ont mentionné qu'ils n'avaient tout simplement pas vu le sens interdit tellement ils étaient concentrés sur leur objectif. Par conséquent, nous avons écarté les trois réponses de ces participants pour les autres scénarios afin de ne pas fausser les résultats.

Après le prétraitement des résultats, nous souhaitons déterminer s'il existe une différence significative entre les réponses du scénario scripté et ceux du scénario généré. Nous proposons d'utiliser pour cela le test des rangs signés de Wilcoxon. Ce test permet de comparer deux mesures d'une variable quantitative effectuées sur les mêmes sujets. L'hypothèse que nous cherchons à vérifier grâce à ce test est que **la différence moyenne entre deux mesures est nulle**. Nous réalisons le test de Wilcoxon sur chacune de ces hypothèses en nous basant sur

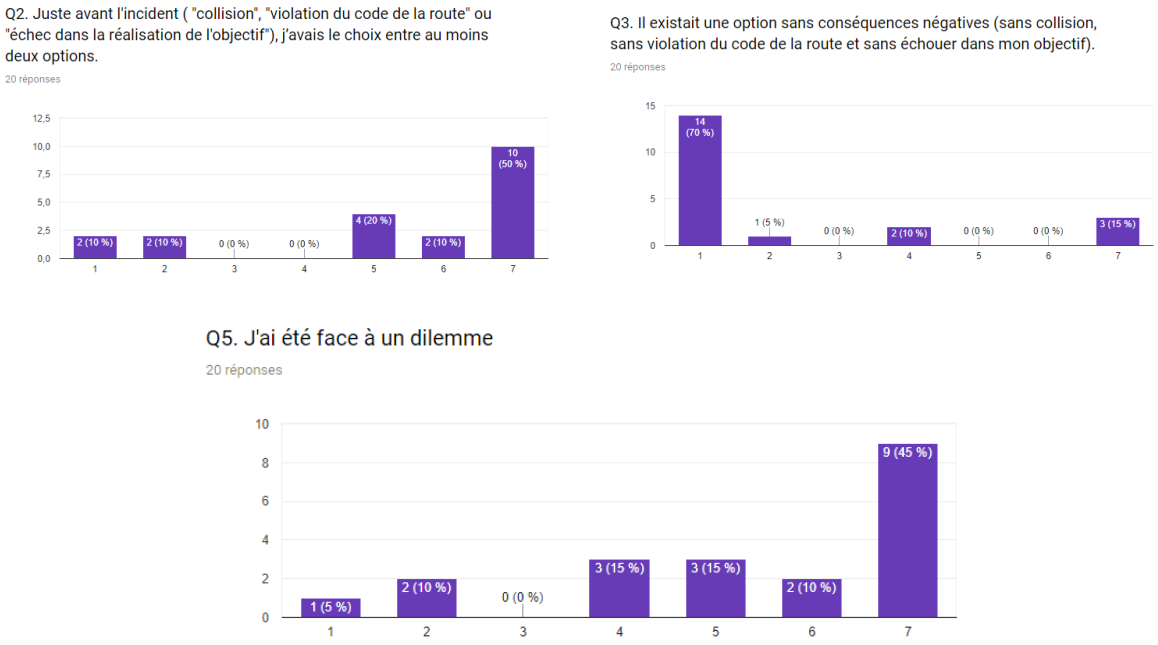

FIGURE 5.3 Réponses aux questions 2,3 et 5 du scénario généré

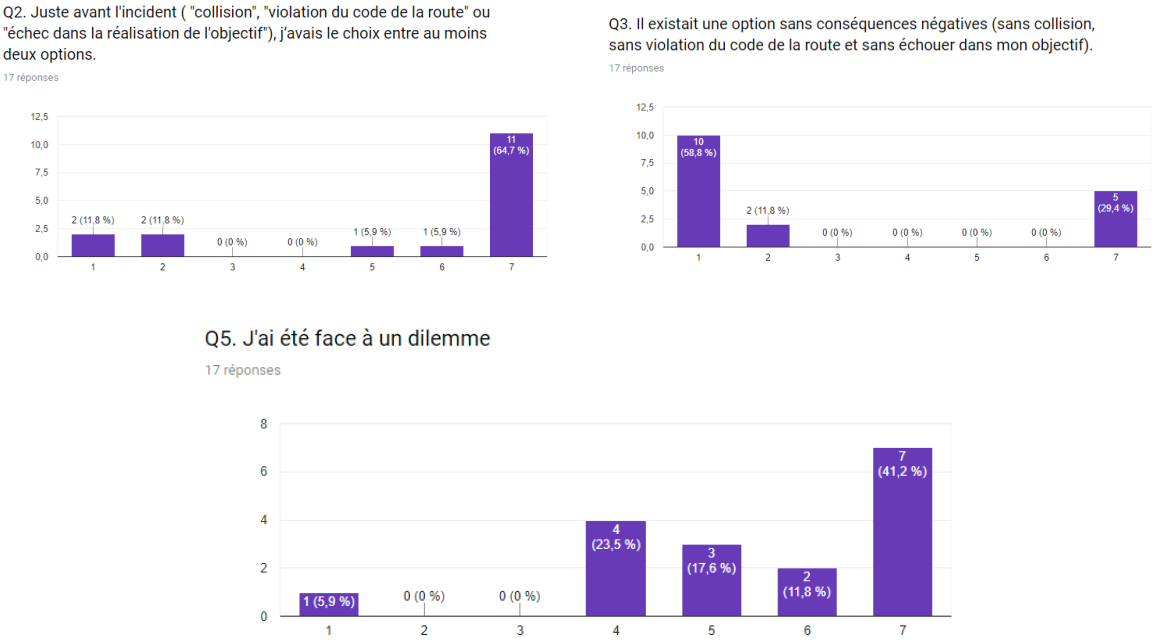

FIGURE 5.4 Réponses aux questions 2,3 et 5 du scénario scripté

les réponses des questions 2,3 et 5. Une *p-value* ≤ 0.1 indiquerait un rejet de l'hypothèse. Le tableau 5.1 dresse un récapitulatif des résultats.

	H0.1	**H0.2**	**H0.3**
p-value	0.41	0.13	0.68

TABLE 5.1 p-value associée à chaque hypothèse

5.2.3 Interprétation des résultats

Dans le dilemme généré, 80% des participants ont affirmé qu'ils avaient le choix entre au moins deux options. Nous pouvons en déduire que dans l'absolu, notre moteur de génération a réussi à générer des situations qui présentent différentes options à l'utilisateur. De plus, en comparant les résultats avec ceux du dilemme scripté, le test de Wilcoxon retourne une *p-value*=0.41 supérieure à 0.1. Nous déduisons donc qu'il n'y a pas de différence significative entre les deux scénarios. L'hypothèse H0.1 est donc vérifiée. Par ailleurs, dans le dilemme généré, 75% des participants ont affirmé que le choix auquel ils étaient confrontés ne proposait aucune issue sans conséquences négatives. Nous pouvons en déduire que dans l'absolu, notre moteur de génération a réussi à produire des situations qui entraînent des conséquences négatives quel que soit le choix de l'utilisateur. De plus, en comparant les résultats du dilemme généré avec ceux du dilemme scripté, le test de Wilcoxon retourne une *p-value*=0.13 supérieure à 0.1. Nous déduisons qu'il n'y a pas de différence significative entre les deux scénarios. L'hypothèse H0.2 est donc vérifiée.

Les éléments précédents ont permis de valider les hypothèses associées aux critères objectifs que nous avons définies pour la description d'un dilemme, à savoir **l'existence de plusieurs options possibles** et **l'absence d'issue sans conséquences négatives**. Nous cherchons également à vérifier l'hypothèse H0.3 qui est plutôt subjective, et qui se rapporte au ressenti des participants face à ces situations. Elle cherche en particulier à vérifier si les participants considèrent les situations rencontrées comme étant des dilemmes. Dans le dilemme généré, 70% des participants ont affirmé qu'ils considéraient qu'ils étaient face à un dilemme. Nous pouvons en déduire que dans l'absolu, notre moteur de génération a réussi à générer des situations perçues comme étant des dilemmes. De plus, en comparant les résultats avec ceux du dilemme scripté, le test de Wilcoxon retourne une *p-value*=0.68 supérieure à 0.1. Nous déduisons qu'il n'y a pas de différence significative entre les deux scénarios. L'hypothèse H0.3 est donc vérifiée.

Grâce à ces éléments nous avons pu vérifier les trois sous-hypothèses H0.1, H0.2 et H0.3. Par conséquent, nous déduisons que l'hypothèse générale est également vérifiée. Autrement dit, que KOBA génère bien des dilemmes.

Bien que ce ne soit pas en relation directe avec la vérification de notre hypothèse de départ, nous présentons ci-après quelques éléments connexes issus de cette expérience. L'analyse des résultats, ainsi que la brève phase de débriefing, nous a permis de mettre en évidence l'importance du poids de chaque issue dans la prise de décision dans un dilemme. En effet, dans le dilemme scripté, la plupart des participants qui ont choisi de griller le sens interdit ont précisé qu'en situation réelle, ils ne le feraient pas. Aussi, ils ont précisé que leur décision pouvait dépendre de l'urgence du rendez-vous auquel ils allaient se présenter. Cela a par conséquent soulevé la question suivante : pourquoi alors avoir grillé le sens interdit en simulation ? Nous pensons que cela est dû au fait que les participants sont conscients qu'il n'y a pas véritablement de sanctions pour cette violation, et qu'en virtuel, ils peuvent alors se permettre de transgresser les lois pour réaliser leur objectif. Cela a d'ailleurs été confirmé par quelques participants — amateurs de jeux vidéo — qui ont affirmé que ce qui était important pour eux dans chaque scénario, c'est la réalisation à tout prix de l'objectif énoncé au départ. De plus, les participants qui ont choisi de ne pas griller le sens-interdit, ont affirmé pour la plupart, que c'était exactement ce qu'ils auraient fait en situation réelle. Certains d'entre eux ont mentionné le fait que leur choix était guidé par une expérience antérieure. Nous pensons particulièrement à un père de famille qui nous a raconté comment il s'était retrouvé dans une situation semblable où il s'agissait plutôt d'un feu rouge à la place d'un sens interdit, et que son choix s'est fait sans hésitation en repensant à cette situation. Ce dernier point met en évidence l'intérêt de confronter les individus à des situations diverses et variées qui vont leur permettre d'acquérir, entre autres, des compétences pour gérer les situations futures. Cela s'est également confirmé par les résultats du scénario généré, où 65% des personnes — pourcentage des personnes ayant percuté le piéton — ont mentionné qu'elles auraient agi différemment si c'était à refaire. En d'autres termes, elles auraient choisi de percuter des éléments du décor au lieu du piéton.

Chapitre 6

Conclusion et limites

Sommaire

6.1 Résumé des travaux 129
6.2 Limites et perspectives 130
6.2.1 Génération des dilemmes 130
6.2.2 Génération de l'ambiguïté 132
6.2.3 L'aspect apprentissage : limites et discussions 134

6.1 Résumé des travaux

Nos travaux portent sur la génération dynamique des situations critiques en environnements virtuels. Nous nous sommes intéressés particulièrement à la génération de deux dimensions critiques en particulier : les dilemmes et les situations ambiguës. Nous avons proposé de reposer le processus de génération sur des modèles de connaissances qui sous-tendent la simulation.

S'agissant de la génération des dilemmes, nous avons proposé KOBA : un moteur de génération qui raisonne sur des modèles de connaissances afin d'extraire les paires d'actions susceptibles de produire des situations dilemmatiques. Nous avons distingué les dilemmes de prohibition et les dilemmes d'obligation. Pour chaque type de dilemme, nous avons proposé une formalisation qui décrit les conditions nécessaires que doivent vérifier les situations dilemmatiques de chaque type. A partir de cette formalisation, nous avons proposé des algorithmes génériques qui raisonnent sur des modèles sémantiques expressifs. L'utilisation de tels modèles permet la génération d'une variabilité de situations tout en minimisant l'effort d'écriture. Pour proposer des situations adaptées à l'apprenant et qui sont en phase avec le diagnostic établi, nous avons proposé d'attribuer un score à chaque paire d'actions

sélectionnée. Ce score correspond au degré de satisfaisabilité des contraintes pédagogiques et scénaristiques. Par ailleurs, afin de prendre en compte les valeurs morales de l'apprenant, nous avons proposé une opérationnalisation de la théorie des valeurs universelles de (Schwartz, 2006). En tenant compte des relations de compatibilité et d'opposition énoncées par cette théorie, nous avons proposé un mécanisme simple de propagation de l'importance des valeurs pour l'apprenant.

S'agissant de la génération des situations ambiguës, nous avons distingué l'ambiguïté de l'information et l'ambiguïté du contexte. Afin de pouvoir générer ces deux types d'ambiguïté, il a fallu tout d'abord proposer une modélisation de la représentation mentale de l'apprenant. Cette représentation correspond à l'ensemble des éléments pertinents pour la réalisation des tâches tels qu'ils sont perçus par l'apprenant. Nous avons proposé d'utiliser les fonctions de croyances afin de représenter l'incertitude sur cette représentation tout en quantifiant l'ignorance et le conflit qui sont deux éléments essentiels pour le raisonnement du moteur de génération. Nous avons alors proposé des algorithmes génériques qui raisonnent sur des modèles de connaissances, et qui prennent en considération ces deux éléments, afin de créer des états d'incomplétude, des états de flou et des états d'incohérence. Les deux premiers états permettent au moteur de générer l'ambiguïté de l'information, tandis que le dernier permet au moteur de générer l'ambiguïté du contexte. Enfin, nous avons proposé la notion de barrière de perception qui permet de décrire les éléments qui limitent la perception des événements déclenchés. Cela permet ainsi de créer des représentations floues et de mettre en doute les sources.

6.2 Limites et perspectives

Que ce soit pour la génération des dilemmes ou pour la génération de l'ambiguïté, si les résultats sont plutôt positifs, il reste toutefois un certain nombre de limites que nous avons à priori identifiées. Dans les sections 6.2.1 et 6.2.2, nous décrivons les limites liées, respectivement, à la génération des dilemmes et de l'ambiguïté, tout en ouvrant sur des perspectives et des pistes d'amélioration.

6.2.1 Génération des dilemmes

Bien que notre proposition sur la génération des dilemmes constitue un apport original par rapport aux systèmes de l'état de l'art, elle présente quelques limites. Dans cette section, nous présentons ces limites en détaillant particulièrement celles liées à la détection des tâches contradictoires, ainsi que celles liées à la manière d'exploiter le modèle de causalité. Nous

soulevons également quelques problèmes sur la modélisation du profil moral de l'apprenant, ainsi que les limites liées à nos évaluations.

Les tâches contradictoires

Une première limite de notre approche de génération de dilemmes réside dans la détection des tâches contradictoires. Pour déterminer que deux tâches sont contradictoires, KOBA raisonne sur les postconditions des tâches, en vérifiant si elles sont incompatibles. Les conditions étant des assertions sous forme de triplets, le moteur considère qu'elles sont incompatibles si pour les mêmes sujet et prédicat, l'objet est différent. Or, cela suppose que le prédicat ne correspond pas à une relation de cardinalité multiple. Par exemple, les assertions `(?Car :has-passenger :Alain)` et `(?Car :has-passenger :Gaston)` sont considérées incompatibles par notre moteur alors qu'elles ne le sont pas en réalité. Ces assertions renseignent en effet sur le fait qu'il y a deux passagers dans la voiture. Si la relation `:has-passenger` était unaire — le cas d'un véhicule mono-siège —, les assertions seraient incompatibles dans ce cas-là. Afin de ne pas retourner des résultats absurdes, il faudrait alors prendre en compte les cardinalités dans l'algorithme de détermination des tâches contradictoires.

L'exploitation du modèle de causalité

Dans le cadre de nos travaux, le moteur de génération considère que l'activation d'une barrière empêche systématiquement le déclenchement des événements postérieurs. Aussi, il considère que la non-réalisation des barrières déclenche systématiquement l'événement postérieur. Or, ce n'est pas vraiment le cas. En effet, le meta-modèle CAUSALITY-DL que nous utilisons, affecte des probabilités aux événements, ainsi qu'un degré de confiance aux barrières. Ces deux aspects ne sont pas considérés par le moteur de génération. Dans une version plus aboutie, le moteur devrait prendre en compte ces éléments en les intégrant dans les contraintes scénaristiques. Plus un événement serait probable, plus la situation le mettant en jeu aurait des chances d'être instanciée. Inversement, plus la confiance sur une barrière serait élevée, moins la situation mettant en jeu l'événement postérieur aurait des chances d'être instanciée.

Profil moral de l'apprenant

Afin de prendre en compte le système de valeurs de chaque individu, nous avons proposé une opérationnalisation de la théorie des valeurs universelles de (Schwartz, 2006). Nous avons proposé une modélisation simple de la structure circulaire en prenant en compte

l'importance de chaque valeur pour l'individu. Cette modélisation mérite d'être enrichie par la prise en compte des incertitudes sur les valeurs. Par ailleurs, le mécanisme de propagation de l'importance des valeurs morales est relativement simple. Nous pensons que l'utilisation d'un cadre formel de l'incertitude permettrait d'améliorer considérablement ce modèle. Nous pensons particulièrement à la théorie des fonctions de croyance. En plus de représenter les croyances sur les valeurs, cette théorie propose également différentes techniques de propagation des croyances en quantifiant si nécessaire le conflit résultant de la fusion des informations. Par conséquent, le moteur de génération peut raisonner sur les différentes masses de croyances pour déterminer quelles valeurs morales sont les plus pertinentes pour la génération d'un dilemme. Ce profil peut également être mis à jour dynamiquement à chaque fois que l'apprenant réalise une action mettant en jeu une ou plusieurs valeurs morales, en prenant en compte les éventuels conflits liés à la fusion de données.

Evaluation

Les premiers résultats issus des évaluations que nous avons réalisées, montrent que notre moteur génère bel et bien des situations dilemmatiques. Toutefois, certains aspects méritent d'être évalués également. Nous pensons particulièrement à la génération de dilemmes moraux en prenant en compte les valeurs morales de chaque individu. Cette évaluation nécessite l'établissement d'un profil moral du participant — grâce à un questionnaire par exemple — avant le début de l'expérience. Ce profil est ensuite transmis en entrée au moteur de génération qui se charge de présenter des situations à conflits moraux. Par ailleurs, dans le cadre de cette thèse, nous n'avons pas évalué l'aspect apprentissage. Cette évaluation permettrait de voir quels enseignements les individus peuvent tirer suite à la confrontation à des dilemmes divers et variés. Aussi, il serait intéressant de voir si les dilemmes sont corrélés avec la mobilisation d'une ou de plusieurs compétences non-techniques particulières.

6.2.2 Génération de l'ambiguïté

La limite principale liée à nos travaux sur la génération de l'ambiguïté concerne l'absence d'évaluation sur le sujet. Nous avons proposé un modèle théorique de génération de situations ambiguës dont les travaux futurs consisteraient à l'implémenter au sein du moteur de génération. Toutefois, cette fondation théorique n'est pas exempte de limites. Dans cette section, nous détaillons en particulier celles liées à la représentation mentale de l'apprenant, aux incertitudes et à la prise en compte des niveaux diégétiques.

Incertitudes et représentation mentale

Dans le cadre de nos travaux, pour la génération de l'ambiguïté, nous avons proposé des algorithmes et mécanismes pour réduire la fiabilité des sources et limiter la perception des éléments selon un taux d'affaiblissement α. La difficulté réside dans le fait de lier ce taux abstrait à des représentations concrètes. La question qu'on pourrait se poser est : est-ce que le fait de définir un taux d'affaiblissement de 50% sur la perception d'un piéton, revient à occulter la moitié du piéton ? La réponse est non, car le simple fait d'apercevoir la tête du piéton est un indice sûr de sa présence. De la même manière un immeuble occulté partiellement par un autre bâtiment n'implique pas forcement que l'incertitude quant à la présence de l'immeuble est proportionnelle à la taille du bâtiment occultant. Par conséquent, la transformation de ce taux abstrait en représentation concrète est un problème complexe qui mérite de plus profondes réflexions.

La principale limite de notre moteur de génération de l'ambiguïté réside dans la représentation mentale de l'apprenant. Par cette représentation, nous souhaitons modéliser comment l'apprenant se représente l'état du monde. Notre proposition constitue une première étape sur le sujet. Toutefois, pour être plus réaliste, il faudrait également prendre en compte les incertitudes sur cette représentation. En d'autres termes, nous devons modéliser les incertitudes qu'a le système sur les incertitudes qu'a l'apprenant sur l'état du monde. L'utilisation du même cadre formel peut être envisagé, à savoir l'utilisation des fonctions de croyance, qui répondent parfaitement aux besoins de représentations incertaines et de fusions d'informations. Les croyances sur les croyances de l'apprenant seraient alors mises à jour après chaque comportement observé chez l'apprenant.

Ambiguïté extradiégétique

Dans le cadre de la génération des situations ambiguës, nous nous sommes concentrés sur l'aspect intradiégétique. Toutefois, nous pensons qu'il est également possible d'exploiter les capacités extradiégétiques des systèmes de simulations pour créer de l'ambiguïté. Nous rappelons que les ajustements extradiégétiques correspondent à tous les éléments qui ne font pas partie du scénario —de l'histoire—, mais qui sont plutôt des propriétés visuelles intrinsèques aux environnements virtuels comme par exemple le fait de mettre en surbrillance des éléments ou afficher des messages d'assistance. Parmi les techniques de production d'ambiguïté, (Gaver et al., 2003) propose d'attirer l'attention sur des choses sans expliquer pourquoi. Cela peut très bien se faire au niveau extradiégétique en mettant, par exemple, en surbrillance — à l'instar des retours multimodaux de (Fricoteaux, 2012) — des éléments non pertinents à la réalisation de la tâche. Cela amènerait l'utilisateur à se questionner sur

la pertinence de cette rétroaction, produisant éventuellement une déstabilisation dans la réalisation de la tâche en cours.

Evaluation

Finalement, afin de valider ces travaux, à l'instar des dilemmes, nous proposons de réaliser à l'avenir une expérience en environnement virtuel. De la même manière, nous proposons de faire passer les participants sur quatre scénarios : un scénario non critique, un scénario critique, un scénario ambigu généré et un scénario ambigu scripté. L'objectif est de voir si le scénario ambigu généré est perçu de la même manière que le scénario scripté. Nous proposons d'étudier cela à l'aide du test utilisé pour les dilemmes : le test des rangs signés de Wilcoxon. Dans les questions présentées aux participants, il serait intéressant d'investiguer si les participants retrouvent dans les situations générées, les propriétés énoncées dans la formalisation de l'ambiguïté, à savoir les états d'incomplétude, de flou ou d'incohérence.

6.2.3 L'aspect apprentissage : limites et discussions

Que ce soit pour les dilemmes ou l'ambiguïté, l'aspect apprentissage n'a malheureusement pas été suffisamment intégré dans nos travaux comme nous l'espérions au tout début de la thèse. Nous rappelons que nous nous situons dans un cadre constructiviste, où l'apprentissage passe par la confrontation à des situations à la fois variées, et pertinentes pour l'apprenant. En d'autres termes, les situations présentées doivent être adaptées à chaque apprenant. Dans le cadre de nos travaux, l'enjeu primordial consistait à générer automatiquement des situations critiques pertinentes. Étant conscients que ce qui est critique pour l'un ne l'est pas forcément pour l'autre, il a été indispensable de prendre en compte le profil apprenant pour adapter les situations afin qu'elles soient perçues comme critiques pour ce profil. En particulier, dans le cadre de la génération des dilemmes, nous avons pris en compte un profil moral de l'apprenant afin que le moteur de génération puisse identifier quelles valeurs morales importent pour l'apprenant. A partir de cela, le moteur est capable d'identifier celles qui sont susceptibles de créer des dilemmes — adaptés à l'apprenant — si elles sont mises en jeu au sein d'une situation. S'agissant de l'ambiguïté, nous avons pris en compte la représentation mentale de l'apprenant afin que le moteur de génération puisse identifier quel est le degré de connaissance de l'apprenant de l'ensemble des variables du monde. A partir de cela, le moteur est capable d'adapter les situations afin de proposer un état du monde propice à la production de l'ambiguïté étant donné l'état de connaissance courant de l'apprenant. Que ce soit pour les dilemmes ou l'ambiguïté, cette adaptation ne prend pas en compte l'aspect apprentissage. En effet, cette adaptation garantit seulement

le fait que les situations sont critiques par rapport à l'apprenant, mais ne garantit en aucun cas le développement des compétences ciblées chez l'apprenant. Nous pouvons alors nous poser la question suivante : quel est l'intérêt de générer automatiquement une multitude de situations si au final il se pourrait qu'aucune d'entre elles ne soit intéressante d'un point de vue pédagogique ? Nous répondons à cette question par le fait que ce qui nous a intéressé principalement dans le cadre de ces travaux, c'est tout d'abord le challenge informatique que suscite la génération automatique de notions complexes telles que les dilemmes et l'ambiguïté. Dans de meilleures circonstances, nous aurions espéré ajouter une brique à notre moteur de génération qui permette de spécifier parmi les situations générées, celles qui sont les plus adaptées pour le développement des compétences. Il aurait fallu dans ce cas, proposer des situations qui prennent en compte le niveau de développement courant de l'apprenant. A partir de ce niveau, le moteur devrait pouvoir proposer des situations qui ne soient pas très complexes au risque de frustrer l'apprenant, et qui ne soient pas très faciles au risque de l'ennuyer. Selon (Vygotsky, 1978), l'enjeu de la formation est de proposer des situations adaptées à l'individu pour favoriser son apprentissage individuel. C'est-à-dire proposer des situations qui appartiennent à ce qu'il définit comme étant la Zone Proximale de Développement [1]. Cette problématique a fait l'objet des travaux de recherche de (Carpentier, 2015) qui a proposé une représentation du profil de l'apprenant comme une opérationnalisation de la zone proximale de développement. Dans cette représentation, les situations sont catégorisées dans des classes qui représentent un espace de situations. Chaque axe de cet espace multidimensionnel correspond à un descripteur. Il peut s'agir d'une variable cognitive, d'une compétence, d'un critère de performance ou d'une tâche. Nous pensons qu'il serait tout à fait pertinent d'étendre la notion de descripteur aux dimensions de criticité. Cela permettrait donc d'identifier dans l'espace de situations, celles qui seraient à la fois critiques pour l'apprenant et pertinentes pour lui du point de vue apprentissage.

Toujours en lien avec l'apprentissage, nous pouvons nous interroger sur l'intérêt de confronter les personnes à des situations dilemmatiques. Autrement dit, quelle est l'utilité de confronter des individus à des situations qui, de toute façon, ne proposent aucune issue positive ? Avant de répondre à cette question, nous tenons à distinguer les dilemmes — ou les situations critiques de façon générale — qui sont imposés par l'environnement (e.g. le choix de Sophie), et les dilemmes qui sont causés par à un comportement non optimal de l'individu (cf. l'exemple de l'aquaplaning et le feu rouge). Dans le premier cas, nous pensons que l'intérêt de confronter les individus à ces dilemmes, consiste à leur permettre de gérer

1. La Zone Proximale de Développement est définie comme étant "la distance entre le niveau de développement actuel, tel qu'on pourrait le déterminer par les capacités de l'enfant à résoudre seul des problèmes, et le niveau de développement potentiel, tel qu'on pourrait le déterminer à travers la résolution de problèmes par cet enfant, lorsqu'il est aidé par des adultes ou collabore avec des pairs initiés"

ces situations plus sereinement. Par conséquent, ils sont plus confiants dans leur choix et moins hésitants lorsqu'il s'agit de gérer des situations semblables. Cela permet notamment de limiter le comportement passif — par hésitation — chez les individus qu'ils pourraient regretter par la suite lorsqu'ils auront pris plus de recul. Dans le deuxième cas, nous pensons que l'intérêt majeur de confronter les individus à ce genre de dilemmes réside non pas dans la gestion du dilemme en soi, mais plutôt dans son anticipation. En effet, l'objectif ici est de montrer aux individus que leur comportement non-optimal a été la cause d'une situation sans issue positive. L'intérêt consiste par conséquent à entraîner les personnes à **anticiper** les situations dilemmatiques, et donc à les pousser à changer leur comportement pour empêcher la survenue de telles situations. Prenons l'exemple d'une voiture qui colle le véhicule de l'apprenant. Ce dernier ne ralentit pas et ne laisse pas la voiture derrière lui le dépasser. Le moteur de génération décide dans ce cas, de faire traverser soudainement un piéton occulté. L'apprenant se retrouve par conséquent face au dilemme suivant : percuter le piéton, ou freiner en urgence et se faire percuter par la voiture de derrière. Ce scénario représente une situation typique où le dilemme se produit à cause d'un comportement non-optimal de l'apprenant. Nous pensons que le confronter à ce genre dilemme est un bon moyen de l'entraîner à anticiper la survenue de situations sans issues positives, et donc de le pousser à changer de comportement avant que le mal ne se produise (e.g. en ralentissant ou en laissant la voiture de derrière le dépasser).

Il arrive également que certaines situations, qui sont considérées comme des dilemmes pour la plupart, ne le soient pas pour certaines personnes. Nous pensons qu'il serait tout de même intéressant de confronter ces personnes à ces dilemmes pour voir si elles ont conscience des retombées de leur choix. Prenons l'exemple d'une personne non conforme aux règles qui n'hésite pas à violer le code de la route lorsque l'occasion se présente. Dans ce cas-là, un dilemme mettant en jeu une infraction au code de la route ne serait pas vraiment un dilemme pour cette personne. Pour voir si elle a pleinement conscience des conséquences de ses décisions, nous pouvons augmenter l'intensité du dilemme et le pousser à l'extrême. Nous pouvons, par exemple, provoquer un accident de la route impliquant l'apprenant si un stop ou un feu rouge sont grillés. Par la suite, si dans des situations semblables, le moteur de génération s'aperçoit que l'apprenant agit de la même manière, il déduit que l'apprenant a déjà préalablement conscience des conséquences et assume totalement sa non conformité aux règles. En revanche, si le moteur s'aperçoit que le comportement de l'apprenant a changé, il peut en déduire que l'apprenant n'avait pas totalement conscience des conséquences négatives de ses choix, et que son système de valeurs a potentiellement changé. Par conséquent, en mettant l'apprenant face à cette situation, et en lui montrant les conséquences extrêmes potentielles auxquelles il n'aurait pas forcément pensé, nous pensons que l'apprenant devient

plus conscient des conséquences de ses choix, l'amenant éventuellement à repenser tout son système de valeurs.

Annexe A

Glossaire

automatique Réalisé par un système informatique sans l'intervention d'un humain dans le processus.

dynamique Dont le résultat change en fonction du temps et des paramètres en entrée.

extradiégétique Un événement extradiégétique est un événément qui n'appartient pas au domaine simulé, se situant donc en dehors de la diégèse.

intradiégétique Un événement intradiégétique est un événément qui appartient au domaine simulé et se situe donc dans la diégèse.

méta-modèle Un méta-modèle est un modèle qui a pour particularité de modéliser d'autres modèles.

module Un module est un un ensemble de composants (e.g. moteurs, modèles, base de données) formant une entité indépendante, dont l'objectif est la réalisation d'une tâche spécifique bien définie.

scénarisation 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-réel.

situation Une situation correspond à un agrégation d'assertions sur le monde.

Annexe B

Les méta-modèles

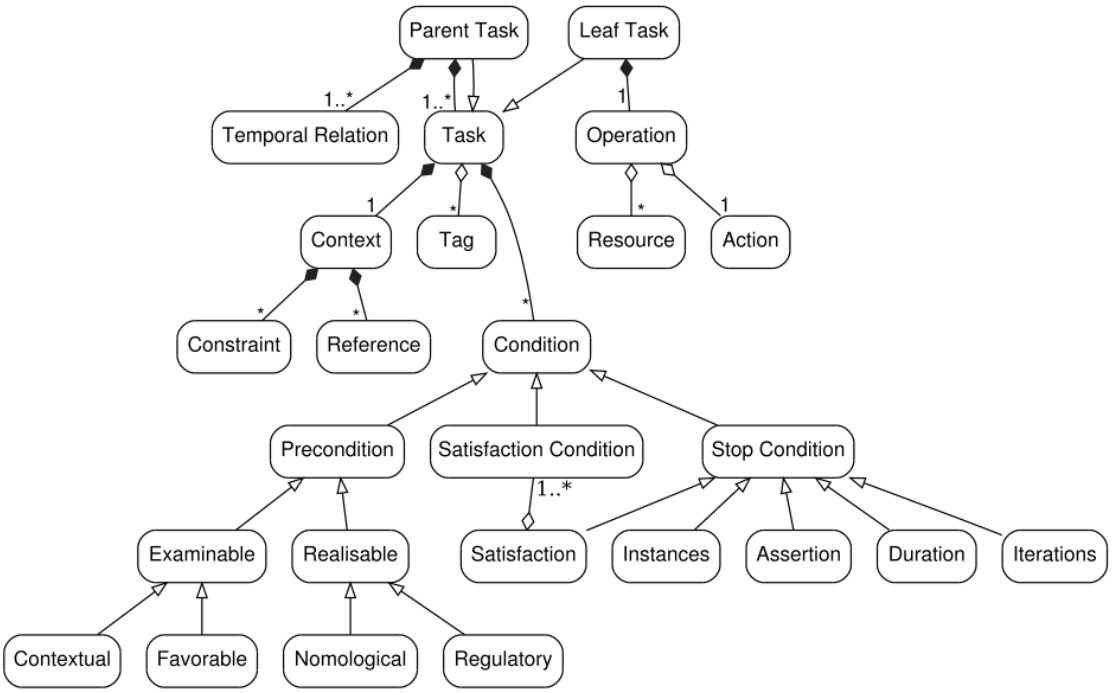

FIGURE B.1 Méta-modèle Activity-DL

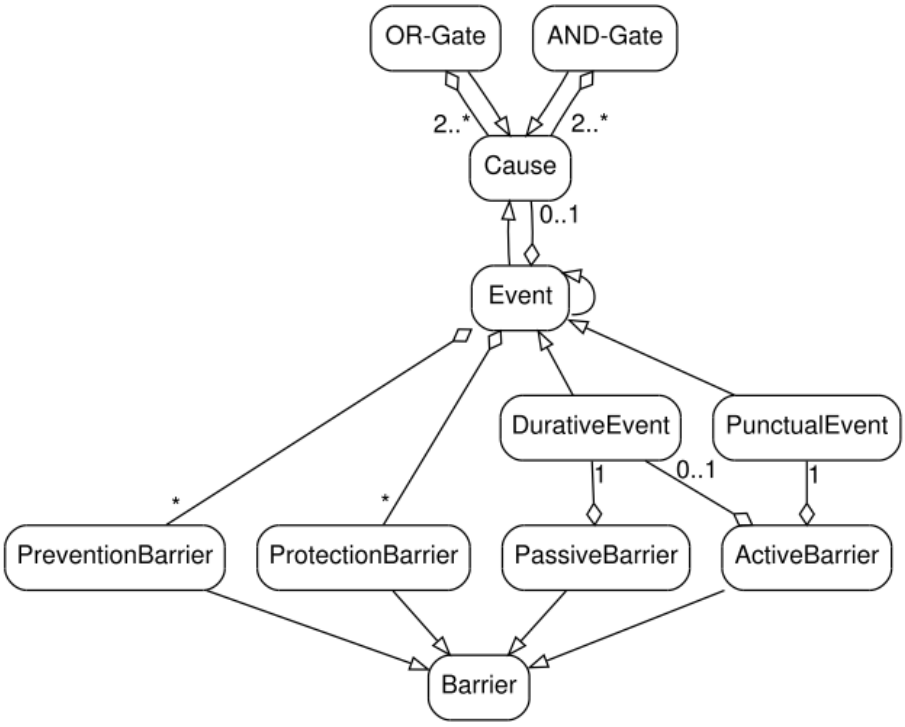

FIGURE B.2 Méta-modèle Causality-DL

Annexe C

Capture d'écrans des scénarios de l'évaluation

FIGURE C.1 Capture d'écran du scénario non-critique de l'évaluation

FIGURE C.2 Capture d'écran du scénario critique de l'évaluation

FIGURE C.3 Capture d'écran du scénario dilemmatique généré de l'évaluation

FIGURE C.4 Capture d'écran du scénario dilemmatique scripté de l'évaluation

Table des figures

1.1 Classification des simulations proposées par (Chiniara, 2007) 16
1.2 Architecture du projet 21
1.3 Architecture de la plateforme HUMANS 22

2.1 Matrice d'évaluation du risque 28
2.2 Zone de simulation de la matrice Acuity-Opportunity 29
2.3 Niveaux de performance (norme ISO 13849-1 :2008) 30
2.4 Les valeurs de bases et items de la théorie des Valeurs Universelles Schwartz (2006) 36
2.5 Valeurs de bases et dimensions 36
2.6 Matrice incertitude-ambiguïté (Schrader et al., 1993) 47
2.7 Espace Fuzziness-Incompleteness-Randomness (FIR) de (Blockley, 2013) . 48
2.8 Plan de la bibliothèque. (1) petite distante, (2) moyenne distance et (3) grande distance. 51
2.9 Lieux visuellement quasi-identiques 52

3.1 Architecture globale 57
3.2 Fragment d'un modèle du monde 64
3.3 Fragment d'un modèle de causalité 69
3.4 Noeuds fictifs de conséquences négatives 75
3.5 Exemple de mise à jour du profil moral suite à deux actions observées . . . 77
3.6 Étapes de génération des dilemmes 82
3.7 Recherche de barrières dont les conséquences sont négatives 84
3.8 Recherche d'actions dont la réalisation simultanée conduit à des conséquences négatives 85
3.9 Recherche de barrières dont la non-réalisation simultanée conduit à des conséquences négatives 86
3.10 Extraction d'un état du monde 96

3.11 Exemple de barrière de perception 106

4.1 Capture écran du démonstrateur 115
4.2 Modèle de tâches utilisé pour le démonstrateur 116
4.3 Modèle de causalité utilisé pour le démonstrateur 116

5.1 Résultat de l'évaluation préliminaire 121
5.2 Salle de réalité virtuelle 122
5.3 Réponses aux questions 2,3 et 5 du scénario généré 125
5.4 Réponses aux questions 2,3 et 5 du scénario scripté 125

B.1 Méta-modèle Activity-DL 141
B.2 Méta-modèle Causality-DL 142

C.1 Capture d'écran du scénario non-critique de l'évaluation 143
C.2 Capture d'écran du scénario critique de l'évaluation 144
C.3 Capture d'écran du scénario dilemmatique généré de l'évaluation 144
C.4 Capture d'écran du scénario dilemmatique scripté de l'évaluation 145

Liste des tableaux

2.1 Tableau comparatif de notre système avec les approches génératives de l'état de l'art 43

3.1 Ensemble des constructeurs disponibles dans ACTIVITY-DL 65
3.2 Préconditions définies dans ACTIVITY-DL (tableau adapté de la source : Carpentier (2015)) 66
3.3 Exemple de tâche modélisée avec ACTIVITY-DL 67

5.1 p-value associée à chaque hypothèse 126

Publications

— **Barre , J. and Benabbou, A. and Corneloup, V. and Bourrier, Y. and Job, A.**
Simulation et Réalité Virtuelle pour l'apprentissage des Compétences Non-Techniques en conduite et en médecine des situations d'urgence
Journées J-RV (journées de l'AFRV), Rennes, France, Oct, 2017

— **Benabbou, A. and Lourdeaux, D. and Lenne, D.** *Generation of Obligation and Prohibition Dilemmas Using Knowledge Models*
29th International Conference on Tools with Artificial Intelligence, Boston, United States, Nov, 2017

— **Benabbou, A. and Lourdeaux, D. and Lenne, D.**
Génération dynamique de dilemmes en environnement virtuel à partir de modèles de connaissances
8ème Conférence sur les Environnements Informatiques pour l'Apprentissage Humain, Strasbourg, France, Jun, 2017

— **Lourdeaux, D. and Benabbou, A. and Huguet, L. and Lacaze-Labadie, R.**
HUMANS : suite logicielle pour la scénarisation d'environnements virtuels pour la formation à des situations socio-techniques complexes Conférence Nationale sur les Applications Pratiques de l'Intelligence Artificielle, Caen, France, pp. 61-68, Jul, 2017

— **Benabbou, A. and Lourdeaux, D. and Lenne, D.**
Dynamic generation of dilemmas in virtual learning environments for non-technical skills training
15th IEEE International Conference on Cognitive Informatics and Cognitive Computing (ICCI*CC 2016), Palo Alto CA, United States, pp. 231-234, Aug, 2016

— **Benabbou, A. and Lourdeaux, D. and Lenne, D.**
Dynamic generation of dilemma-based situations in virtual environments
13th International Conference, ITS 2016, Zagreb, pp. 526-527, Poster, Jun, 2016

— **Burkhardt, J. M., Corneloup, V., Garbay, C., Bourrier, Y., Jambon, F., Luengo, V., ... & Lourdeaux, D.**.
Simulation and virtual reality-based learning of non-technical skills in driving : critical situations, diagnostic and adaptation.
IFAC-PapersOnLine, 49(32), 66-71 (2016).

Bibliographie

Amokrane, K. (2010). *Suivi de l'apprenant en environnement virtuel pour la formation à la prévention des risques sur des sites {SEVESO}*. PhD thesis, Université de Technologie de Compiègne.

Badawi, M. and Donikian, S. (2004). Autonomous Agents Interacting With Their Virtual Environments Through Synoptic Objects. In *Proceedings of {CASA} 2004*, pages 179–187.

Baker, D., Prince, C., Shrestha, L., Oser, R., and Salas, E. (1993). Aviation Computer Games for Crew Resource Management Training. *The International Journal of Aviation Psychology*, 3(2) :143–156.

Barber, H. (2008). Generator of Adaptive Dilemma-based Interactive Narratives. (October) :1–18.

Barot, C. (2014). *Scénarisation d'environnements virtuels. Vers un équilibre entre contrôle, cohérence et adaptabilité*. PhD thesis, Université de technologie de Compiègne.

Barre, J., Benabbou, A., Corneloup, V., Bourrier, Y., and Job, A. (2017). Simulation et Réalité Virtuelle pour l'apprentissage des Compétences Non-Techniques en conduite et en médecine des situations d'urgence.

Bertholet, C. (2000). *Du jugement a la construction de la competence ethique en soins infirmiers. Role de la formation initiale et de l'experience professionnelle*. PhD thesis, Université de Nantes.

Bille, W., De Troyer, O., Kleinermann, F., Pellens, B., and Romero, R. (2004). Using Ontologies to Build Virtual Worlds for the Web. In *ICWI*, pages 683–690.

Blockley, D. (2013). Analysing uncertainties : Towards comparing Bayesian and interval probabilities'. *Mechanical Systems and Signal Processing*, 37(1-2) :30–42.

Bobillier, P., Kahn, B., and Probst, A. (1976). *Simulation With GPSS and GPSS/V*. Prentice-Hall.

Bonnefon, J.-F., Shariff, A., and Rahwan, I. (2016). The social dilemma of autonomous vehicles. *Science*, 352(6293) :1573–1576.

Bourrier, Y., Francis, J., Garbay, C., and Luengo, V. (2018). A Hybrid Architecture for Non-Technical Skills Diagnosis. In *Intelligent Tutoring Systems*, Montreal.

Brink, D. O. (1994). Moral Conflict and Its Structure. *The Philosophical Review*, 103(2) :215.

Burkhardt, J.-M., Corneloup, V., Garbay, C., Bourrier, Y., Jambon, F., Luengo, V., Job, A., Cabon, P., Benabbou, A., and Lourdeaux, D. (2016). Simulation and virtual reality-based learning of non-technical skills in driving : critical situations, diagnostic and adaptation. In *IFAC-PapersOnLine*, volume 49, pages 66–71.

Burkhardt, J.-M., Lourdeaux, D., and Mellet-d'Huart, D. (2006). La réalité virtuelle pour l'apprentissage humain. *G. Moreau (Coord.), B. Arnaldi & P. Guitton (Coords). Le Traité de la réalité virtuelle*, 4 :43–100.

Camerer, C. and Weber, M. (1992). Recent Developments in Modelling Preferences : Uncertainty and Ambiguity. *Journal of Risk and Uncertainty*, 5 :325–370.

Carpentier, K. (2015). *Scénarisation personnalisée dynamique dans les environnements virtuels pour la formation*. PhD thesis, Université de Technologie de Compiègne.

Catrinu, M. and Nordgård, D. (2011). Integrating risk analysis and multi-criteria decision support under uncertainty in electricity distribution system asset management. *Reliability Engineering & System Safety*, 96(6) :663–670.

Chataigne, C. (2014). *Psychologie des valeurs*. De Boeck.

Chevaillier, P., Querrec, R., and Septseault, C. (2009). {VEHA}, un métamodèle d'environnement virtuel informé et structuré. *Techniques et sciences informatiques*, 28(6-7) :715–740.

Chiniara, G. (2007). Simulation médicale pour acquisition des compétences en anesthésie. In *Congrès national d'anesthésie et de réanimation*, pages 41–49.

Chiniara, G., Cole, G., Brisbin, K., Huffman, D., Cragg, B., Lamacchia, M., Norman, D., and Canadian Network For Simulation In Healthcare, G. W. G. (2013). Simulation in healthcare : A taxonomy and a conceptual framework for instructional design and media selection. *Medical Teacher*, 35(8) :e1380–e1395.

Clement, J. (1989). Learning via Model Construction and Criticism. In *Handbook of Creativity*, pages 341–381. Springer US, Boston, MA.

Conee, E. (1982). Against Moral Dilemmas. *The Philosophical Review*, 91(1) :87.

Cooper, J. B., Long, C. D., Newbower, R. S., and Philip, J. H. (1982). Critical incidents associated with intraoperative exchanges of anesthesia personnel. *Anesthesiology*, 56(6) :456–61.

Cottone, P., Pieti, L., Schiavinato, V., Soru, D., Martinelli, M., Varotto, D., and Mantovani, G. (2009). "solving" ambiguity in the virtual space : Communication strategies in a collaborative virtual environment. *Cognition, Technology and Work*, 11(2) :151–163.

Cushman, F., Gray, K., Gaffey, A., and Mendes, W. B. (2012). Simulating murder : The aversion to harmful action. *Emotion*, 1.

Debray, B., Chaumette, S., Descouriere, S., and Trommeter, V. (2006). OMEGA 7 - Méthodes d'analyse des risques générés par une installation industrielle. Technical report.

Dubois, D. and Prade, H. (1988). *Possibility theory : An approach to computerized processing of uncertainty*. Plenum Press, New York.

Duncan, R. B. (1972). Characteristics of Organizational Environments and Perceived Environmental Uncertainty. *Administrative Science Quarterly*, 17(3) :313.

Edward, L. (2011). *Modélisation décisionnelle de personnages virtuels autonomes évoluant dans un environnement pour la présentation des risques sur les sites {SEVESO}*. PhD thesis, Université de Technologie de Compiègne.

Edward, L., Amokrane, K., Lourdeaux, D., and Barthès, J.-P. (2010). An Ontology for Managing a Virtual Environment for Risk Prevention. In *2010 First International Conference on Integrated Intelligent Computing*, pages 62–67, Bangalore, India.

Ellsberg, D. (1961). Risk, Ambiguity, and the Savage Axioms. *The Quarterly Journal of Economics*, 75(4) :643.

Fellner, W. (1961). Distortion of Subjective Probabilities as a Reaction to Uncertainty. *The Quarterly Journal of Economics*, 75(4) :670.

Festinger, L. (1957). *A theory of cognitive dissonance*.

Flanagan, J. C. and C., J. (1954). The critical incident technique. *Psychological Bulletin*, 51(4) :327–358.

Flin, R., O'Connor, P., and Crichton, M. (2008). Safety at the sharp end : A guide to non-technical skills ; Chapter 2. In *Safety at The Sharp End : A Guide to Non-Technical Skills*, pages 17–40.

Flin, R., Winter, J., and Cakil Sarac, M. R. (2009). Human factors in patient safety : review of topics and tools. *World Health*, 2.

Foot, P. (1967). The problem of abortion and the doctrine of double effect.

Francis, K. B., Howard, C., Howard, I. S., Gummerum, M., Ganis, G., Anderson, G., and Terbeck, S. (2016). Virtual Morality : Transitioning from Moral Judgment to Moral Action ? *PLOS ONE*, 11(10) :e0164374.

Fricoteaux, L. (2012). *Interaction informée par un système décisionnel pour un retour multimodal en environnement virtuel : application à la formation au transport fluvial*.

Frisch, D. and Baron, J. (1988). Ambiguity and rationality. *Journal of Behavioral Decision Making*, 1(3) :149–157.

Galbraith, J. R. and R., J. (1973). *Designing complex organizations.* Addison-Wesley Pub. Co.

Garner, W. R. (1962). *Uncertainty and Struture as Psychological Concepts*.

Gaver, W. W., Beaver, J., and Benford, S. (2003). Ambiguity as a resource for design. In *Proceedings of the conference on Human factors in computing systems - CHI '03*, number January 2003, page 233.

Gratch, J. and Marsella, S. (2004). A Domain-independent Framework for Modeling Emotion. *Journal of Cognitive Systems Research*, 5(4) :296–306.

Greene, J. D., Sommerville, R. B., Nystrom, L. E., Darley, J. M., and Cohen, J. D. (2001). An fMRI Investigation of Emotional Engagement in Moral Judgment. *Science*, 293(5537).

Gremler, D. D. (2004). The Critical Incident Technique in Service Research. *Journal of Service Research*, 7(1) :65–89.

Harmon, S. (2016). An Expressive Dilemma Generation Model for Players and Artificial Agents. In *The Twelfth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE-16)*, pages 176–182.

Hauser, M., Cushman, F., Young, L., Jin, R. K. X., and Mikhail, J. (2007). A dissociation between moral judgments and justications. *Mind & Language*, 22(1) :1–21.

Heath, C. and Tversky, A. (1991). Preference and belief : Ambiguity and competence in choice under uncertainty. *Journal of Risk and Uncertainty*, 4(1) :5–28.

Helmreich, R. L. (2000). On error management : lessons from aviation. *BMJ (Clinical research ed.)*, 320(7237) :781–5.

Huguet, L., Lourdeaux, D., Sabouret, N., and Ferrer, M.-H. (2016). Perturbed Communication in a Virtual Environment to Train Medical Team Leaders. *Studies in health technology and informatics*, 220 :146–9.

Irwin, A. and Poots, J. (2015). The human factor in agriculture : An interview study to identify farmers' non-technical skills. *Safety Science*, 74 :114–121.

Johansen, I. L. and Rausand, M. (2015). Ambiguity in risk assessment. *Safety Science*, 80 :243–251.

Kallmann, M. and Thalmann, D. (1999a). Direct 3d interaction with smart objects. In *Proceedings of the {ACM} symposium on Virtual reality software and technology*, pages 124–130.

Kallmann, M. and Thalmann, D. (1999b). Modeling objects for interaction tasks. In *Computer animation and simulation'98 : proceedings of the Eurographics Workshop in Lisbon, Portugal, August 31-September 1, 1998*, page 73.

Klinke, A. and Renn, O. (2002). A New Approach to Risk Evaluation and Management : Risk-Based, Precaution-Based, and Discourse-Based Strategies. *Risk Analysis*, 22(6) :1071–1094.

Kohlberg, L. (1958). *The development of modes of moral thinking and choice in the years 10 to 16*. PhD thesis, University of Chicago.

Lacaze-Labadie, R., Lourdeaux, D., and Sallak, M. (2017). Heuristic approach to guarantee safe solutions in probabilistic planning. In *29th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2017)*, 2017 IEEE 29th International Conference on Tools with Artificial Intelligence Proceedings, Boston, United States.

Lecomte, M.-A. (2006). La formation à l'éthique des étudiants en soins infirmiers (Belgique). *Recherche en soins infirmiers*, 86(3) :4.

Lourdeaux, D., Benabbou, A., Huguet, L., and Lacaze-Labadie, R. (2017). HUMANS : suite logicielle pour la scénarisation d'environnements virtuels pour la formation à des situations socio-techniques complexes. In *Conférence Nationale sur les Applications Pratiques de l'Intelligence Artificielle*.

Luce, R. D. R. D. and Raiffa, H. (1989). *Games and decisions introduction and critical survey*. Dover Publications.

Luengo, V. (2009). *Les rétroactions épistémiques dans les Environnement Informatique pour l'Apprentissage Humain*. PhD thesis, Université Joseph Fourier - Grenoble I.

Mantovani, G., Gamberini, L., Martinelli, M., and Varotto, D. (2001). Exploring the Suitability of Virtual Environments for Safety Training : Signals, Norms and Ambiguity in a Simulated Emergency Escape. *Cognition Technology Work*, 3(1) :33–41.

March, J. G. (1987). Ambiguity and accounting : The elusive link between information and decision making. *Accounting, Organizations and Society*, 12(2) :153–168.

Marcus, R. B. (1980). Moral Dilemmas and Consistency. *The Journal of Philosophy*, 77(3) :121.

McConnell, T. (1988). Interpersonal Moral Conflicts. *American Philosophical Quarterly*, 25(1) :25–35.

McConnell, T. C. (1978). Moral Dilemmas and Consistency in Ethics.

Milgram, S. (1963). Behavioral Study of obedience. *Journal of Abnormal and Social Psychology*, 67(4) :371–378.

Miller, G. A. and Frick, F. C. (1949). Statistical behavioristics and sequences of responses. *Psychological Review*, 56(6) :311–324.

Mintzberg, H., Raisinghani, D., and Theoret, A. (1976). The Structure of &quot ;Unstructured&quot ; Decision Processes. *Administrative Science Quarterly*, 21(2) :246.

Mollet, N. (2005). *De l'Objet-Relation au Construire en Faisant : application à la spécification de scénarios de formation à la maintenance en réalité virtuelle.* PhD thesis.

Mollet, N. and Arnaldi, B. (2006). Storytelling in Virtual Reality for Training. In Pan, Z., Aylett, R., Diener, H., Jin, X., Göbel, S., and Li, L., editors, *Technologies for E-Learning and Digital Entertainment*, volume 3942, pages 334–347. Springer Berlin Heidelberg, Berlin, Heidelberg.

Nabil Bin NASIR, A., Farzeeha ALI, D., Khair Bin NOORDIN, M., and Safarin Bin NORDIN, M. (2011). Technical skills and non-technical skills : predefinition concept. In *Proceedings of the IETEC'11 Conference*.

Navarrete, C. D., McDonald, M. M., Mott, M. L., and Asher, B. (2012). Virtual morality : emotion and action in a simulated three-dimensional "trolley problem". *Emotion (Washington, D.C.)*, 12(2) :364–70.

Newbower, R. S., Cooper, J. B., and Long, C. D. (1981). Learning from anesthesia mishaps : analysis of critical incidents in anesthesia helps reduce patient risk. *QRB. Quality review bulletin*, 7(3) :10–6.

Noothigattu, R., Gaikwad, S. N. S., Awad, E., Dsouza, S., Rahwan, I., Ravikumar, P., and Procaccia, A. D. (2017). A Voting-Based System for Ethical Decision Making.

OMG (2014). {OMG} Systems Modeling Language.

Pernin, J.-P., Lejeune, A., and Professionnelle, A. (2004). MODELES POUR LA REUTILISATION DE SCENARIOS D'APPRENTISSAGE Modèles pour la réutilisation de scénarios d'apprentissage.

Pourin, C., Daucourt, V., and Barberger-Gateau, P. (2001). Utilisation de la méthode des incidents critiques dans la construction d'un outil de mesure de la satisfaction en psychiatrie. *Santé Publique*, 13(2) :169.

Raybourn, E. M., Deagles, E., Mendini, K., and Heneghan, J. (2005). Adaptive Thinking & Leadership Simulation Game Training for Special Forces Officers. In *Interservice/Industry Training, Simulation, and Education Conference*, Orlando.

Renn, O. (2008). White Paper on Risk Governance : Toward an Integrative Framework. In *Global Risk Governance*, pages 3–73. Springer Netherlands, Dordrecht.

Reznek, M., Harter, P., and Krummel, T. (2002). Virtual Reality and Simulation : Training the Future Emergency Physician. *Academic Emergency Medicine*, 9(1) :78–87.

Rickel, J., Gratch, J., Hill, R., Marsella, S., and Swartout, W. (2001). Steve Goes to Bosnia : Towards a New Generation of Virtual Humans for Interactive Experiences.

Rokeach, M. (1973). *The Nature of Human Values*.

Rouillé, M., Bilavarn, J., Burkhardt, J.-M., Couix, S., Lourdeaux, D., and Amokrane, K. (2008). Prototype du langage de modélisation de l'activité humaine. Technical report.

Rovira, A., Swapp, D., Spanlang, B., and Slater, M. (2009). The Use of Virtual Reality in the Study of People's Responses to Violent Incidents. *Frontiers in behavioral neuroscience*, 3 :59.

Salas, E. and Burke, C. S. (2002). Simulation for training is effective when.... *Quality & safety in health care*, 11(2) :119–20.

Schrader, S., Riggs, W. M., and Smith, R. P. (1993). Choice over uncertainty and ambiguity in technical problem solving. *Journal of Engineering and Technology Management*, 10(1-2) :73–99.

Schwartz, S. H. (1992). Universals in the Content and Structure of Values : Theoretical Advances and Empirical Tests in 20 Countries. *Advances in Experimental Social Psychology*, 25 :1–65.

Schwartz, S. H. (2006). *Les valeurs de base de la personne : Théorie, mesures et applications*, volume 47.

Schwartz, S. H., Cieciuch, J., Vecchione, M., Davidov, E., Fischer, R., Beierlein, C., Ramos, A., Verkasalo, M., Lönnqvist, J.-E., Demirutku, K., Dirilen-Gumus, O., and Konty, M. (2012). Refining the theory of basic individual values. *Journal of Personality and Social Psychology*, 103(4) :663–688.

Shafer, G. (1976). *A mathematical theory of evidence*, volume 1. Princeton university press Princeton.

Simon, H. A. (1987). Making Management Decisions : The Role of Intuition and Emotion.

Sinnott-Armstrong, W. (1988). *Moral dilemmas*. B. Blackwell.

Skulmowski, A., Bunge, A., Kaspar, K., and Pipa, G. (2014). Forced-choice decision-making in modified trolley dilemma situations : a virtual reality and eye tracking study. *Frontiers in Behavioral Neuroscience*, 8(December) :426.

Slater, M., Antley, A., Davison, A., Swapp, D., Guger, C., Barker, C., Pistrang, N., and Sanchez-Vives, M. V. (2006). A virtual reprise of the Stanley Milgram obedience experiments. *PloS one*, 1(1) :e39.

Spierling, U. and Szilas, N. (2009). Authoring issues beyond tools. In *Interactive Storytelling*, pages 50–61. Springer.

Sütfeld, L. R., Gast, R., König, P., and Pipa, G. (2017). Using Virtual Reality to Assess Ethical Decisions in Road Traffic Scenarios : Applicability of Value-of-Life-Based Models and Influences of Time Pressure. *Frontiers in Behavioral Neuroscience*, 11 :122.

Tardif, J. (2006). *L'évaluation des compétences : documenter le parcours de développement*. Cheneliere.

Tardif, J. and Meirieu, P. (1996). Stratégie pour favoriser le transfert des connaissances. *Vie pédagogique*, 98(7) :4–7.

Thue, D., Bulitko, V., Spetch, M., and Romanuik, T. (2010). Player agency and the relevance of decisions. *Interactive Storytelling*, pages 210–215.

Valdesolo, P. and Desteno, D. (2006). Manipulations of emotional context shape moral judgment. *Psychological Science*, 17(6) :476–477.

Vallentyne, P. (1989). Two Types of Moral Dilemmas. *Erkenntnis (1975-)*, 30 :301–318.

Van Rijsselbergen, D., Van De Keer, B., Verwaest, M., Mannens, E., and Van de Walle, R. (2009). Movie script markup language. In *Proceedings of the 9th ACM symposium on Document engineering - DocEng '09*, page 161, New York, New York, USA. ACM Press.

Vanderhaegen, F. (2016). A rule-based support system for dissonance discovery and control applied to car driving. *Expert Systems with Applications*, 65 :361–371.

Vygotsky, L. S. (1978). *Mind in Society*. Harvard University Press, Cambridge, {MA}.

Weick, K. E. (1995). *Sensemaking in organizations*. Sage Publications.

Westrum, R. (2006). A typology of resilience situations. In *Resilience engineering : Concepts and percepts*, pages 55–65.

Williams, B. A. O. and Atkinson, W. F. (1965). Symposium : Ethical Consistency. *Proceedings of the Aristotelian Society, Supplementary Volumes*, 39 :103–138.

Williams, R. M. J. (1968). Values. *International encyclopedia of the social sciences*.

Zadeh, L. A. (1965). Fuzzy sets. *Information and control*, 8(3) :338–353.

Zadeh, L. A. (1978). Fuzzy sets as a basis for a theory of possibility. *Fuzzy sets and systems*, 1 :3–28.

Zimmerman, M. J. (1995). Prima Facie Obligation and Doing the Best One Can. *Philosophical Studies : An International Journal for Philosophy in the Analytic Tradition*, 78 :87–123.