UNIVERSITÉ D'AIX-MARSEILLE

ECOLE DOCTORALE ED 184

FACULTÉ DES SCIENCES ET TECHNIQUE

LSIS UMR CNRS 7296 / DIMAG

THESE DE DOCTORAT

Spécialité : Informatique

M'hammed Ali OULHACI

Évaluation individuelle et collective dans les jeux sérieux collaboratifs : Application à la gestion de crise

Soutenue le 20 juin 2014

Composition du jury :

Domitile LOURDEAUX	Université Technologique de Compiènne	Rapporteur
Chihab HANACHI	Université Toulouse 1, Capitole	Rapporteur
Julie DUGDALE	Université Pierre Mendes, Grenoble 2	Examinateur
Gilles GESQUIERE	Université Lumière, Lyon 2	Examinateur
Bernard ESPINASSE	Aix-Marseille Université (AMU)	Directeur de thèse
Erwan TRANVOUEZ	Aix-Marseille Université (AMU)	Co-Directeur de thèse
Sébastien FOURNIER	Aix-Marseille Université (AMU)	Co-Directeur de thèse

Remerciements

Mes remerciements vont d'abord à mes directeurs de thèse, Messieurs Bernard ESPINASSE, Erwan TRANVOUEZ et Sébastien FOURNIER pour m'avoir offert la possibilité de travailler sur cette thèse, ainsi que la société SII et ses dirigeants sans le soutien desquels cette thèse n'aurait eu lieu.

J'aimerais exprimer ma gratitude à Monsieur TRANVOUEZ pour la confiance qu'il m'a accordé, et pour toutes ces années durant lesquelles j'ai appris énormément de choses grâce à ses conseils avisés, à Monsieur FOURNIER pour les retours dont j'ai pu bénéficier et enfin à Monsieur ESPINASSE dont le recul et l'expérience m'ont permis de prendre plus de hauteur sur mes propre travaux.

Je remercie aussi Monsieur Benoît POULARD pour l'expertise et l'aide appréciables qu'il m'a apporté dans le projet SIMFOR, et je remercie aussi Monsieur Joël HALLAKOUN pour son soutien au sein de la société SII.

Je tiens aussi à remercier Madame Domitile LOURDEAUX et Monsieur Chihab HANACHI pour avoir rapporté cette thèse. Je les remercie pour leurs remarques constructives et détaillées qui ont contribué à améliorer considérablement le manuscrit.

Je remercie aussi Madame Julie DUGDALE et Monsieur Gilles GESQUIERE pour avoir accepté d'examiner les travaux de cette thèse.

Bien évidemment, je remercie l'ensemble des doctorants, chercheurs, enseignants, personnels que j'ai eu l'occasion de côtoyer au sein du laboratoire des sciences de l'information et des systèmes.

Enfin, toute ma reconnaissance va vers mes parents, mon épouse Faïza et tout ceux qui m'ont soutenu et encouragé durant ces années.

Table des matières

Table des matières iii

Introduction générale 1

Contexte, problématique et état de l'art 5

I Contexte et problématique de recherche 7
- I.1 Introduction . . . 7
- I.2 Les jeux sérieux et l'enjeu de l'apprentissage . . . 8
- I.3 Vers un véritable environnement de formation . . . 9
  - I.3.1 Quels besoins pour les jeux sérieux . . . 9
  - I.3.2 Comment compléter les besoins des jeux sérieux . . . 9
    - I.3.2.1 L'évaluation des apprenants . . . 10
    - I.3.2.2 Représentation des connaissances . . . 11
    - I.3.2.3 Analyse des traces . . . 12
- I.4 Un domaine d'application privilégié : la gestion de crise . . . 12
  - I.4.1 Définition de la gestion de crise . . . 13
  - I.4.2 Évaluation des apprenants et la gestion de crise . . . 14
- I.5 Approche méthodologique de notre recherche . . . 15
- I.6 Objectif de recherche . . . 15

II Jeux sérieux et formation 17
- II.1 Introduction . . . 18
- II.2 Les jeux sérieux . . . 18
  - II.2.1 Quelques modélisations du concept de jeu sérieux . . . 20

II.2.2 Quelques exemples de jeux sérieux 22
II.2.2.1 Cadre d'analyse 22
II.2.2.2 Un jeu sérieux pour l'apprentissage des langues 26
II.2.2.3 FORMAT-STORE Application à un supermarché virtuel . 28
II.2.2.4 Un jeu sérieux pour les accidents de la route 31
II.2.2.5 Un jeu sérieux de formation des infirmières aux urgences 33
II.2.2.6 Escape from Wilson Island (EFWI) 36
II.2.2.7 Autres travaux 38
II.2.3 Synthèse et conclusion 39
II.3 Les systèmes tutoriels intelligents 39
II.3.1 Définition et architecture 40
II.3.2 Quelques travaux dans le domaine des STIs 43
II.3.2.1 Cadre d'analyse 43
II.3.2.2 Help Agent for Learning (HAL) 43
II.3.2.3 PEGASE (PEdagogical Generic and Adaptive SystEm) . 46
II.3.2.4 Helpful agent for safEty leaRning in virtuAl environment (HERA) 51
II.3.3 Synthèse et conclusion 55
II.4 Les SMA dans les jeux sérieux 55
II.4.1 Simulation 56
II.4.2 Architecture et intégration 57
II.5 L'évaluation des apprenants 57
II.6 Étude comparative 59
II.7 Conclusion 62

Contributions **63**

III Un cadre méthodologique pour l'évaluation individuelle et collective dans les jeux sérieux collaboratifs **65**
III.1 Introduction 66
III.2 Problématique de l'évaluation collective dans les jeux sérieux 66
III.3 Le concept d'espace d'évaluation 68
III.3.1 Définition du concept d'espace d'évaluation 68
III.3.2 Illustration du concept d'espace d'évaluation dans un jeu sérieux collaboratif pour la gestion de crise 72
III.3.2.1 L'évaluation des apprenants dans la formation à la gestion de crise 73
III.3.2.2 Définition des espaces d'évaluation pour la formation à la gestion de crise 74

III.4 Les espaces d'évaluation pour la formation à la gestion de crise 74
III.4.1 Espace physique 75
III.4.1.1 Les composants de l'espace physique 75
III.4.1.2 L'évaluation dans l'espace physique 76
III.4.2 Espace comportemental 76
III.4.2.1 Les composants de l'espace comportemental 77
III.4.2.2 L'évaluation dans l'espace comportemental 79
III.4.3 Espace social 82
III.4.3.1 Les composants de l'espace social 82
III.4.3.2 L'évaluation dans l'espace social 83
III.5 Conclusion 86

IV Une architecture multi-agents pour l'évaluation individuelle et collective dans les jeux sérieux 87
IV.1 Introduction 88
IV.2 Pourquoi les SMA ? 88
IV.3 Évolution d'un jeu sérieux vers un Système Tutoriel Intelligent 89
IV.3.1 Rôle du jeu sérieux 90
IV.3.2 Présentation générale de l'architecture multi-agents 90
IV.4 Module d'évaluation 92
IV.4.1 Agent Source de Données 93
IV.4.2 Agent Indicateur 94
IV.4.3 Agent évaluateur 95
IV.5 Module de simulation des comportements 96
IV.5.1 Modélisation des agents de jeu 97
IV.5.2 Modélisation des actions 98
IV.5.3 Modélisation des dialogues 100
IV.6 Agent de l'apprenant 100
IV.7 Agent pédagogique 101
IV.8 Représentation des connaissances 102
IV.8.1 Modèle du domaine 102
IV.8.2 Modèle de l'apprenant 103
IV.9 Interaction entre les différents agents 104
IV.10 Conclusion 106

Applications 107

V Implémentation 109
V.1 Introduction 109

V.2 Présentation du projet SIMFOR . . . 110
V.3 Architecture de SIMFOR . . . 110
V.4 Simulation de comportement . . . 113
V.4.1 Éditeur d'agent . . . 114
V.4.2 Éditeur de dialogue . . . 115
V.5 Architecture . . . 116
V.5.1 Serveur SIMFOR . . . 116
V.5.2 Client SIMFOR . . . 118
V.6 Suivi et interaction des agents . . . 121
V.7 Évaluation . . . 122
V.7.1 Éditeur du modèle du domaine . . . 123
V.7.2 Moniteur d'évaluation . . . 124
V.8 Les agents . . . 126
V.9 Conclusion . . . 131

VI Illustration et validation sur un exemple de scénario de gestion de crise 133
VI.1 Introduction . . . 134
VI.2 Quelques définitions . . . 134
VI.2.1 Définition d'un scénario dans SIMFOR . . . 134
VI.2.2 Les attributs d'un scénario SIMFOR . . . 135
VI.2.2.1 Les acteurs . . . 135
VI.2.2.2 Les actions . . . 135
VI.2.2.3 Les moyens . . . 135
VI.2.2.4 Les environnements . . . 135
VI.2.2.5 Les sinistres . . . 136
VI.3 Présentation du scénario TMD . . . 137
VI.3.1 Fonctionnement général . . . 137
VI.3.2 Déroulement du scénario . . . 139
VI.4 Validation sur le Scénario TMD . . . 139
VI.4.1 Protocole de validation . . . 139
VI.4.2 Résultats et discussions . . . 141
VI.5 Conclusion . . . 144

Conclusion et perspectives 145

Annexes 155

A Les moteurs de jeux vidéo 155
A.1 Introduction . . . 155

A.2 Qu'est qu'un moteur de jeu ? . . . 156
A.3 Quelque exemple de moteur de jeu . . . 156
A.3.1 Delta 3D . . . 157
A.3.2 Unreal engine . . . 158
A.3.3 CryEngine . . . 159
A.3.4 Unity 3D . . . 160
A.4 Conclusion . . . 161

B Les interfaces de SIMFOR 163
B.1 Introduction . . . 163
B.2 Les actions dans SIMFOR . . . 163
B.2.1 L'interface téléphone . . . 164
B.2.2 L'interface fax . . . 165
B.2.3 L'interface radio . . . 166
B.2.4 L'interface carte . . . 167
B.2.5 L'interface main courante . . . 168
B.2.6 L'interface de déplacement . . . 169
B.3 Autres interfaces . . . 170
B.4 Conclusion . . . 171

Liste des publications 175

Références bibliographiques 184

Index des illustrations 188

Index des tableaux 189

Introduction générale

L'intérêt croissant des jeux sérieux, notamment dans le domaine de la formation, a soulevé des besoins nouveaux en termes d'évaluation des joueurs [Nieborg, 2004] et de simulation de comportement [Mathieu et al., 2011]. Un jeu sérieux est généralement défini comme un jeu privilégiant l'éducation (au sens large) plutôt que le divertissement [Michael et Chen, 2005], et basé sur des techniques issues des jeux vidéo (environnement 3D, modes d'interaction). Les jeux sérieux utilisent aussi la simulation pour reproduire des phénomènes complexes ou couteux (phénomènes physiques, conséquences catastrophes naturelles...). Lorsqu'un nombre important d'acteurs est impliqué il peut être nécessaire de simuler aussi des comportements d'acteurs humains pouvant être absents lors d'une session de jeu (appelés Personnages Non Joueurs PNJ).

Dans un jeu sérieux, le but est d'apprendre une ou plusieurs compétences (liée(s) à un métier ou à des qualités plus générales) à un ou plusieurs apprenants/joueurs. Chaque jeu sérieux répond en général à un objectif particulier de formation, en rapport à un contexte de formation spécifique lié à un corps de métier (ex. sécurité [Amokrane, 2010], pilotage [Lourdeaux, 2001] etc.), ou à un corpus de connaissance (type scolaire ou universitaire) [Baudouin et al., 2007], ou en encore de qualités sociales (gestion des conflits, coopération...) [Wendel et al., 2012]. Mesurer qualitativement ou quantitativement le succès ou non de cet apprentissage peut tout à la fois mettre en œuvre des solutions ad hoc ou générique. Ainsi, ce qui relève *in fine* d'un processus d'évaluation constitue ces dernières années un enjeu majeur dans les jeux sérieux [Nieborg, 2004] s'ajoutant aux problématiques plus classiques de conception et de scénarisation.

Dans cette thèse, nous nous intéressons plus particulièrement à l'évaluation des apprenants d'un jeu sérieux dans un contexte d'apprentissage collaboratif. Ce contexte présente la particularité d'intégrer non seulement l'évaluation individuelle des acteurs, selon leurs propres objectifs de formation, mais également leurs interactions avec d'autres

acteurs pouvant influer sur l'issue de la séance de formation. Ce travail de recherche est illustré avec le projet SIMFOR, un jeu sérieux pour la formation des non professionnels à la gestion de crise. Le projet SIMFOR se caractérise par l'implication d'un nombre d'acteurs significatif, aux compétences variées, engagés dans des processus coopératifs. Il s'agit à la fois de proposer des évaluations individuelles particulières, propres à chaque acteur, ainsi qu'une évaluation du bon déroulement des interactions entre ces acteurs. Nous montrons en quoi les systèmes multi-agents (SMA), relevant de l'Intelligence Artificielle Distribuée, peuvent améliorer l'apprentissage humain dans un contexte collectif, au travers de la simulation de comportements humains et de l'évaluation des joueurs apprenants.

Dans le premier chapitre nous détaillerons le contexte de notre problématique de recherche qui est représentée par l'évaluation des apprenants dans les jeux sérieux. Nous présentons tous d'abord le concept de jeu sérieux, ensuite nous présentons en quoi l'évaluation des apprenants représente un véritable enjeu de recherche dans les jeux sérieux. Nous verrons les différents types d'évaluation (individuelle, collective, temps réel, final), nous verrons aussi les besoins des jeux sérieux pour l'évaluation des apprenants, et enfin nous montrons en quoi la gestion de crise peut représenter un cadre d'application intéressant. Dans le deuxième chapitre nous présentons différents travaux dans le domaine des Jeux Sérieux (JS) et des Systèmes Tutoriels Intelligents (STI), nous analyserons ces travaux selon différents points de vue : la représentation des connaissances, l'évaluation des apprenants et leur architecture. Nous verrons aussi en quoi les systèmes multi-agents peuvent aider à améliorer la conception de jeux sérieux (autant d'un point de vue simulation qu'architectural). Enfin nous dressons un tableau comparatif (JS et ITS) de tous les travaux présentés, et nous essayons d'extraire les points faibles et les points forts de chaque outil. Notre conclusion est que l'évaluation individuelle et collective est peu abordée dans le contexte de l'apprentissage collaboratif, surtout lorsque cet apprentissage est hétérogène.

Dans le troisième chapitre nous présentons notre contribution pour répondre à la problématique de l'évaluation individuelle et collective dans un jeu sérieux collaboratif. Notre première contribution consiste à proposer un cadre méthodologique pour l'évaluation des apprenants basé sur le concept d'espace d'évaluation. Un espace d'évaluation est un quadruplé composé d'une représentation des connaissances (Kw), d'un ensemble d'indicateurs (I), de métriques (M), et enfin d'un ensemble de modèles d'évaluation (AM). Nous illustrons ensuite le concept d'espace d'évaluation dans le cadre de la gestion de crise (dans le cadre du projet SIMFOR), en définissant trois espaces d'évaluation : un espace physique, un espace comportemental, et un espace social.

Dans le quatrième chapitre nous présentons notre deuxième contribution consistant en l'opérationnalisation de notre cadre méthodologique pour l'évaluation des appre-

nants dans un jeu sérieux qui pose en soit des enjeux de modélisation indépendamment de questions purement techniques. Cette opérationnalisation est réalisée au travers d'un système multi-agent reprenant les différents modules d'un STI. Notre but étant de renforcer l'aspect évaluation des apprenants en gardant l'aspect immersif et ludique d'un jeu sérieux pour obtenir un outil de formation optimal. L'architecture générale de notre système s'articule autour d'un jeu sérieux (SIMFOR en l'occurrence), d'un module d'évaluation (qui est composé de plusieurs agents responsable de l'évaluation), d'un agent pédagogique, du modèle de l'apprenant, du modèle du domaine, et enfin d'un module de simulation de comportements pour simuler les personnages non joueurs (PNJ).

Le cinquième chapitre présente le projet SIMFOR, un jeu sérieux pour la formation à la gestion de crise ainsi que les différents outils d'édition (éditeur d'agent pour la simulation de PNJ, éditeur de dialogue, éditeur du modèle du domaine) et de suivi (suivi des PNJ, suivi des apprenants) pour le suivi et l'évaluation des apprenants.

Dans le sixième chapitre nous présentons un scénario de gestion de crise pour illustrer l'évaluation individuelle et collective des apprenants selon le cadre méthodologique proposé et le système logiciel le supportant. Le scénario de l'exercice concerne un accident TMD (transport de matières dangereuses) dans la ville d'Arles. Les résultats de l'exercice sont discutés et analysés pour mieux comprendre le comportement des apprenants.

Enfin en conclusion nous dressons un bilan sur nos contributions, notre cadre méthodologique pour l'évaluation individuelle et collective des apprenants dans un jeu sérieux collaboratif. Nous présentons les différentes perspectives à cette recherche notamment liées à l'usage de cette évaluation pour améliorer les rétroactions de l'agent pédagogique et ainsi améliorer encore plus l'apprentissage des apprenants.

Première partie

Contexte, problématique et état de l'art

CHAPITRE I

Contexte et problématique de recherche

Sommaire

I.1 Introduction 7
I.2 Les jeux sérieux et l'enjeu de l'apprentissage 8
I.3 Vers un véritable environnement de formation 9
I.3.1 Quels besoins pour les jeux sérieux 9
I.3.2 Comment compléter les besoins des jeux sérieux 9
I.3.2.1 L'évaluation des apprenants 10
I.3.2.2 Représentation des connaissances 11
I.3.2.3 Analyse des traces 12
I.4 Un domaine d'application privilégié : la gestion de crise 12
I.4.1 Définition de la gestion de crise 13
I.4.2 Évaluation des apprenants et la gestion de crise 14
I.5 Approche méthodologique de notre recherche 15
I.6 Objectif de recherche 15

I.1 Introduction

Dans ce chapitre nous présentons le contexte et la problématique générale de la thèse. Celle ci tourne autour de l'apprentissage dans les jeux sérieux ainsi que l'apprentissage dans les environnements virtuels de formation. Dans la section I.2, nous présentons le concept de jeu sérieux, dans la section I.3, nous présentons les besoins

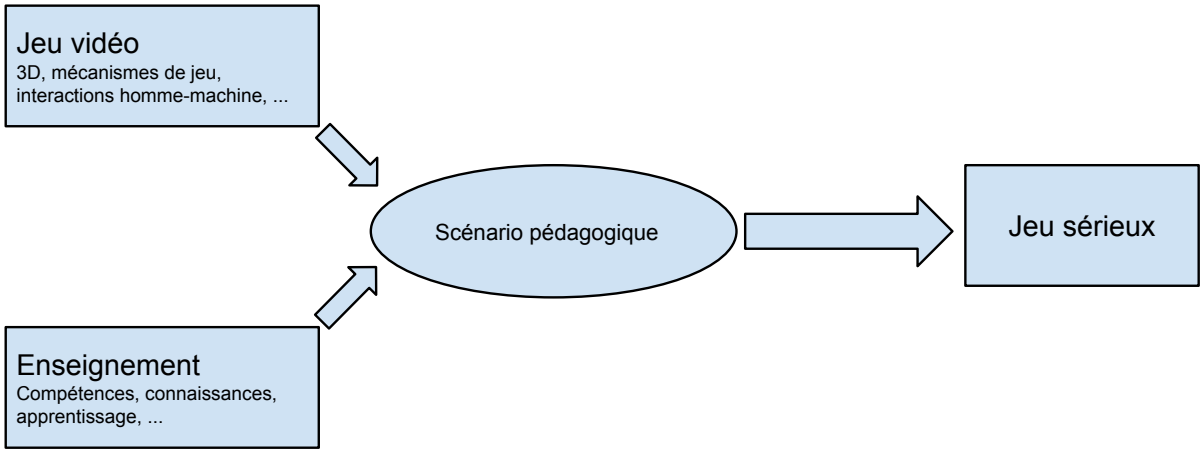


Figure I.1 – Schéma général du concept de jeu sérieux.

des jeux sérieux en terme d'apprentissage et comment compléter ces besoins. Dans la section I.4 nous présentons un domaine d'application intéressant pour les jeux sérieux qui est la gestion de crise, nous verrons ensuite (section I.4.2) comment les jeux sérieux peuvent aider à améliorer l'apprentissage de la gestion de crise, et enfin nous détaillons nos objectifs et notre approche méthodologique respectivement dans la section I.5 et I.6.

I.2 Les jeux sérieux et l'enjeu de l'apprentissage

Le concept de jeu sérieux est quelque peu difficile à cerner car les deux mots sont contradictoires (expression oxymoron "*jeu*" et "*sérieux*"). Les professionnels du domaine définissent un jeu sérieux comme étant un jeu qui privilégie l'éducation (au sens large) plutôt que le divertissement [Michael et Chen, 2005]. Il existe plusieurs autres définitions pour le concept de jeu sérieux, dans le domaine informatique on peut citer la définition de Micheal Zyda [Zyda et al., 2003] qui définit un jeu sérieux par un défi cérébral joué avec un ordinateur qui utilise le divertissement en tant que valeur ajoutée. On retrouve aussi des approches issues des Sciences Psychologiques et de l'Education, notamment celle de André Tricot [Tricot et Alain, 1999] qui met l'accent sur le scénario pédagogique. Dans [Alvarez, 2007], Julian Alvarez propose une définition unifiée d'un jeu sérieux : *Application informatique, dont l'objectif est de combiner à la fois des aspects sérieux (Serious) tels, de manière non exhaustive, l'enseignement, l'apprentissage, la communication, ou encore l'information, avec des ressorts ludiques issus du jeu vidéo (Game). Cette association doit s'opérer par l'implémentation d'un scénario pédagogique*.

La figure I.1 montre un schéma très général d'un jeu sérieux selon la définition de Julian Alvarez. Le jeu vidéo est associé aux techniques d'enseignement afin d'aboutir

à un outil d'apprentissage en environnent virtuel, et cela à travers un scénario pédagogique. Michael Zyda [Zyda, 2005] est l'un des premiers à parler de pédagogie (voir section II.2.1) en précisant qu'en plus des jeux vidéo classiques, les jeux sérieux intègrent en plus une dimension pédagogique. Zyda définit la pédagogie par des activités qui éduquent ou instruisent, et diffusant de ce fait de la connaissance ou de la compétence et cela rend les jeux sérieux [Alvarez, 2007].

Alvarez reprend les travaux d'André Tricot [Tricot et Alain, 1999] et propose la définition suivante pour le "***scénario pédagogique***" : *Fonction dédiée à un "objectif pédagogique", dont la propriété est de susciter l'envie d'apprendre et dont la réalisation dépend d'un jeu vidéo avec lequel elle puisse s'intégrer.*

Après la définition des jeux sérieux, nous aborderons dans la section suivante les besoins et les outils nécessaires à l'amélioration de l'apprentissage dans les jeux sérieux.

I.3 Vers un véritable environnement de formation

Dans cette section nous essayons d'extraire les besoins en terme d'apprentissage en environnent virtuel et comment y répondre notamment dans le cadre de jeux sérieux afin d'aboutir à un véritable environnement de formation.

I.3.1 Quels besoins pour les jeux sérieux

Lors de la conception d'un jeu sérieux, une part très importante des ressources est allouée à la partie ludique du jeu sérieux (environnement 3D, mécanismes de jeu, réalité virtuelle, ...) alors que l'évaluation des apprenants hérite d'une petite partie des ressources [Johnson et Wu, 2008] [Stone, 2009], voire ignorée tout simplement (comme dans [Vidani et Chittaro, 2009]). Cela s'explique par la nature même des jeux sérieux, qui peuvent être définis comme étant avant toute chose un jeu vidéo conçu dans le but d'enseigner quelque chose [Stone, 2009]. Dans la plupart des jeux sérieux, l'évaluation des apprenants est ainsi réalisée de façon manuelle via un tuteur qui supervise les apprenants, ou bien par l'apprenant lui même, ce qui peut être problématique, car cela oblige la présence d'un moniteur (formateur) durant l'apprentissage, et l'évaluation peut différer d'un moniteur à un autre et ainsi compromettre le résultat d'apprentissage.

I.3.2 Comment compléter les besoins des jeux sérieux

L'accompagnement des apprenants et leur évaluation est un sujet déjà traité dans la littérature [Johnson et al., 1998] [Lourdeaux, 2001]. Les besoins d'évaluation des jeux

sérieux ciblent plus particulièrement l'évaluation durant l'apprentissage (aide aux apprenants) et après l'apprentissage (diagnostic sur les compétences apprises) impliquant éventuellement un "débriefing" entre l'apprenant et le moniteur. Cette évaluation pose des questions sur les données ou connaissances nécessaires et les moyens de les obtenir de l'environnement informatique. Les sections suivantes développent ces différents points. Nous verrons d'abord qu'est ce que l'évaluation ainsi que les différents types (durant la partie, final, individuelle, collective). L'évaluation sert à évaluer l'apprentissage d'une compétence ou d'une connaissance. Pour réaliser l'évaluation il faut donc d'abord représenter cette connaissance, dans la section I.3.2.2 nous verrons différents types de représentation de ces connaissances. Enfin dans la section I.3.2.3, nous verrons comment recueillir l'activité de l'apprenant dans l'environnement virtuel, appelé dans la littérature "*recueil de traces*".

I.3.2.1 L'évaluation des apprenants

L'évaluation dans les jeux sérieux est une problématique de recherche en soit [Thomas Benjamin et al., 2012] [Nieborg, 2004]. Les travaux [de Freitas et Jarvis, 2006] [Yusoff et al., 2009], proposent des frameworks pour les jeux sérieux et aborde la notion de l'évaluation et de son importance, mais l'évaluation en elle-même est effectuée soit par l'apprenant lui-même (autoévaluation) soit par des moniteurs qui suivent les apprenants. Dans [Raybourn, 2007], Raybourn parle de l'évaluation durant la partie qui représente la base de la rétroaction faite aux apprenants (feedback). Dans [Johnson et Wu, 2008], Johnson présente un modèle d'évaluation individuelle dans le cadre du projet TLCTS (Tactical Language and Culture Training System). Ce modèle d'évaluation est assez simple et se base sur la réponse des apprenants dans différents exercices, où chaque exercice reçoit une pondération selon son importance. Dans [Raybourn, 2005] et [Wendel et al., 2012], les auteurs abordent le potentiel de l'aspect multi-apprenants dans les jeux sérieux et s'inspirent des jeux de rôle massivement multi-joueurs (MMO RPG, Massively Multiplayer Online Role Playing Game) pour la conception de jeux sérieux. Dans [Wendel et al., 2012], Wendel pointe le manque de jeux sérieux pour l'apprentissage de tâches collaboratives et propose un modèle de jeu sérieux collaboratif (escape from Wilson island) basé sur l'apprentissage coopératif assisté par ordinateur (Computer Supported Collaborative Learning, CSCL) [Stahl et al., 2006]. Néanmoins, l'évaluation de la performance collective n'est pas abordée, et l'évaluation se fait de manière individuelle via un questionnaire. Il est difficile de produire une évaluation collective dans ce genre de jeu sérieux, d'autant plus si l'objectif d'apprentissage diffère d'un apprenant à un autre (tout en convergeant vers un but global commun).

L'analyse de l'état de l'art nous a permis de distinguer plusieurs types d'évaluations,

le Tableau 1 résume ces différents types d'évaluation sur deux dimensions (durant/final, individuelle/collective).

Tableau I.1 – Les différents types dévaluation.

Évaluation	Durant la partie	Finale
Individuelle	IT	IF
Collective	CT	CF

L'évaluation Individuelle durant la partie (IT) représente une évaluation formatrice et/ou diagnostique de l'apprenant [Dintilhac et Ignace, 2005]. L'évaluation IT permet d'informer l'apprenant sur sa performance et éventuellement servir de base pour corriger et aider l'apprenant. L'évaluation individuelle finale (IF) représente une évaluation sommative de l'apprenant. L'évaluation IF permet de valider (ou pas) les compétences à apprendre par l'apprenant. L'évaluation collective durant la partie (CT) permet d'évaluer la performance globale du groupe d'apprenants à l'instant t. Cette évaluation va soit donner des indications à l'instructeur ou bien servir à déclencher certains événements dans la partie (via des scripts basés sur un scénario donné). L'évaluation collective finale (CF) permet d'évaluer la performance globale du groupe ainsi que la collaboration entre les apprenants. L'évaluation CF peut impliquer tous les types d'évaluation (IT, IF, CT). Dans notre recherche bibliographique (chapitre II), nous avons constaté que la plupart des travaux traitent la problématique de l'évaluation individuelle et la question de l'évaluation collective et de l'apprentissage de tâches collaboratives est très peu abordée.

L'évaluation dans les jeux sérieux est une problématique à laquelle peut répondre les Systèmes Tutoriels Intelligents (STI), pour qui l'évaluation des apprenants constitue un enjeu théorique et expérimental majeur. En combinant les possibilités d'évaluation d'un STI avec un jeu sérieux, nous pouvons ainsi améliorer les résultats de formation en utilisant un jeu sérieux. Nous verrons plus en détails le concept de STI dans la section II.3.

I.3.2.2 Représentation des connaissances

Définir et mettre en œuvre des moyens d'évaluation des acteurs d'un jeu sérieux, implique de caractériser les connaissances et ou les données contribuant à cette évaluation ainsi que leurs sources. Cette section aborde brièvement les enjeux qui en découlent.

L'évaluation a pour objectif de certifier l'apprentissage de l'apprenant. Cet apprentissage a pour but d'acquérir une certaine compétence ou connaissance, dont la certification dans un environnement informatique nécessite une représentation (symbolique ou non) manipulable par cet environnement. La représentation et la manipulation

des connaissances représente en elle-même un domaine vaste de recherche, avec plusieurs paradigmes de modélisation en fonction de la nature des connaissances ainsi que des objectifs de manipulation [Anderson, 1988]. Dans la section II.2.2.1, nous verrons quelques exemples de modèle de représentation des connaissances.

Dans un contexte d'évaluation de compétences hétérogènes, on peut s'attendre à des besoins de représentation variés, ainsi que des capacités de manipulation de ces représentations adaptées. Il peut alors être nécessaire de recourir à plusieurs modèles de représentation de connaissances au sein d'un même environnement en fonction des besoins d'évaluation posés par les différents rôles joués par les acteurs du jeu sérieux.

Pour obtenir les connaissances liées à l'apprenant et à son apprentissage, il faut les recueillir depuis l'environnent d'apprentissage. Dans la littérature, on appelle ce processus le recueil des traces, que nous allons traiter dans la section suivante.

I.3.2.3 Analyse des traces

Pour évaluer l'apprenant, il faut récupérer son activité dans l'environnement virtuel, autrement appelé dans la littérature « recueil de traces ». Le tuteur (au sens large) exploitera ces traces pour savoir si l'activité proposée est réussie, ou bien pour repérer où sont les difficultés des apprenants. Dans un ITS cela servira de base pour fournir un support à l'apprenant, pour qu'il puisse visualiser son activité et détecter des informations signifiantes pour lui [Baudouin et al., 2007].

Dans un ITS, le recueil des traces est assuré à l'aide du modèle de l'apprenant [VanLehn, 1988]. Le modèle de l'apprenant contient les connaissances de l'apprenant sur le domaine ainsi que les actions réalisées dans l'environnement, ce qui peut de fait constituer une trace. On peut avoir des traces de type déplacement, par exemple avec l'utilisation d'un dispositif de pointage avec des coordonnées. On peut avoir aussi des interactions de type communication (entre un ou plusieurs apprenants), etc.

Ayant introduit les enjeux des jeux sérieux que ce travail de thèse entend adresser, la section suivante les illustre sur le domaine de la gestion des crises.

I.4 Un domaine d'application privilégié : la gestion de crise

La gestion de crise pose en soit des questions qui intéressent les jeux sérieux par les réponses qu'ils peuvent (en partie) apporter, tout en soulevant des questions scientifiques qu'il convient de traiter. Cette section les introduit brièvement.

I.4.1 Définition de la gestion de crise

La gestion de crise est une activité coopérative dans laquelle plusieurs acteurs (intervenants) ayant différents objectifs à plusieurs niveaux (allant de l'opérationnel au stratégique) coopèrent pour gérer une crise. Une crise est généralement définie comme une situation difficilement maîtrisable.

En France, la gestion de crise est basée sur le plan ORSEC (Organisation de la Réponse la de SEcurité Civile). Le plan est conçu pour mobiliser et coordonner, sous l'autorité unique du préfet, les acteurs de la sécurité civile au delà du niveau de réponse courant ou quotidien des services de la sécurité civile. Le but est de préparer tous les acteurs, publics ou privés, pouvant intervenir dans le champ de la protection des populations [DDSC, 2006]. Chaque acteur doit s'approprier les missions relevant de sa compétence et les retranscrire dans son organisation interne. Le processus de développement du dispositif ORSEC doit prendre en compte, de manière très pragmatique et précise, tous les aspects nécessaires à son fonctionnement : organisationnels, techniques et humains. Mais surtout, pour être pleinement efficace, cette organisation doit s'appuyer sur des acteurs formés. L'organisation (regroupant les différents intervenants de la gestion de crise) ne doit pas se découvrir lorsque l'événement survient mais être rôdée par des entraînements et des exercices. Le but final est d'aboutir à une maîtrise partagée (inter-services) et pérenne d'un savoir-faire opérationnel. Néanmoins, le seul moyen de tester ces plans est la mise en place d'exercices en conditions réelles, ce qui peut devenir très lourd en termes d'organisation et très coûteux. Dans ce contexte, des outils informatiques de formation comme les jeux sérieux apparaissent comme une solution alternative pertinente.

Pour réduire les coûts et gagner du temps, certains organismes utilisent déjà les jeux sérieux pour la formation. Avec l'ère du numérique, de nombreuses écoles ont intégré ces outils pour la formation des élèves. Les jeux sérieux apportent un aspect ludique à l'apprentissage et permet de mieux appréhender le problème.

Bien évidemment, les jeux sérieux ne permettent pas de remplacer un exercice réel de plan ORSEC, car il n'y a pas de pression ni de manipulation d'outils concrets (principalement pour les pompiers). Toutefois, les jeux sérieux permettent aux acteurs potentiels de la gestion de crise de s'habituer aux procédures du plan ORSEC ainsi qu'à se confronter au travail d'équipe. En effet, la gestion de crise est avant tout un processus collaboratif qui a pour objectif le retour à la situation normale. Ainsi l'objectif de l'utilisation des jeux sérieux est avant tout d'apprendre à communiquer (entre les différents intervenants de la gestion de crise).

Pour valider l'apprentissage, nous devons évaluer non seulement individuellement les apprenants mais aussi collectivement. L'évaluation individuelle est déjà problématique en soit car dans le domaine de la gestion de crise nous avons plusieurs intervenants avec des cœurs de métier différents qui collaborent (compétences hétérogènes). En plus de cela, l'évaluation collective doit évaluer la performance globale du groupe ainsi que la qualité de la collaboration des différents apprenants (aspect social du groupe) . Nous allons voir plus en détails cet aspect dans la section suivante.

I.4.2 Évaluation des apprenants et la gestion de crise

La formation à la gestion de crise est un processus collaboratif permettant à plusieurs personnes d'acquérir une compétence (différente ou identique), avec la particularité de ne pas cibler uniquement les spécialistes du domaine mais également des non professionnels du risque (préfet, maire, directeur d'établissement, etc.). La gestion d'une crise majeure peut mobiliser plusieurs centaines d'intervenants, du Préfet dans son bureau au pompier sur le terrain. Ces intervenants sont amenés à communiquer et à collaborer pour ramener la situation à un état normal.

L'apprentissage collaboratif cible majoritairement l'apprentissage de compétences homogènes (à l'image d'une classe où les élèves apprennent la même leçon [Jermann et al., 2004] [Giordani et al., 2005]), alors que la gestion de crise est un processus collaboratif qui passe par la réalisation de plusieurs tâches différentes (dépendantes du rôle de l'acteur ainsi que du sinistre). Donc, la gestion de crise vise un apprentissage de tâches collaboratives hétérogènes, réalisées par des acteurs différents ayant un objectif commun. Ainsi la gestion de crise présente des besoins spécifiques du fait de son caractère multi-acteurs qui requiert deux types d'évaluation : une évaluation individuelle et une évaluation collective. La résolution d'une crise (incendie, pollution...) passe par la résolution de toutes les procédures des intervenants : dès lors l'évaluation individuelle peut avoir une influence sur l'évaluation collective, et réciproquement. Considérons, par exemple, un apprenant qui a bien effectué ses procédures, mais dont le but global n'a pas été atteint (e.g. pertes matérielles et humaines). Cet apprenant doit être évalué sur sa performance individuelle et collective pour définir la raison de l'échec (manque de communication, procédure manquante d'un autre apprenant, ...).

Les jeux sérieux visent aussi à immerger l'apprenant dans un monde virtuel qui lui permet d'agir comme il le ferait dans une situation d'urgence. Les connaissances et les compétences impliquées dans une telle situation sont diverses dans leur nature et leur source. Ainsi, se pose la question de comment évaluer et certifier (ou non) les connaissances et les compétences acquises par les joueurs.

I.5 Approche méthodologique de notre recherche

Comme nous l'avons vu précédemment, l'évaluation des apprenants implique l'acquisition d'un ensemble hétérogène d'informations à la fois par leur nature (données brutes, connaissances, procédures) et par leur source (niveau logiciel). Dans ce mémoire nous proposons une approche multi-agent répondant à deux objectifs : i) permettre la production et l'exploitation de ces informations hétérogènes en vue de produire des indicateurs qui alimenteront une évaluation multicritères ; et ii) mettre en œuvre le processus d'évaluation par un système multi-agents accompagnant la production d'évaluations individuelles et collectives. Cette approche est développée et illustrée sur le jeu sérieux SIMFOR (Simulation & Formation) dédié à la gestion de crise.

I.6 Objectif de recherche

L'objectif de notre travail est de répondre à l'aspect hétérogène et distribué de l'évaluation dans le cadre d'un jeu sérieux collaboratif dédié à la gestion de crise. Cette évaluation doit intégrer l'évaluation individuelle (durant la partie et finale) pour chaque apprenant ainsi que l'évaluation collective qui permet de mesurer la performance globale du groupe ainsi que sa capacité de communication. L'évaluation durant la partie servira de base pour la rétroaction (aide aux apprenants) et l'évaluation finale sera utilisée pour certifier (ou non) l'apprentissage de l'apprenant. Comme nous l'avons vu dans la section I.4.2, cette évaluation doit réunir des données de natures hétérogènes et de sources hétérogènes pour calculer la performance globale du groupe. L'objectif de notre travail de recherche vise à :

- *Définir les comportements nominaux* : c'est-à-dire, tel que prescrits dans les procédures officielles de gestion de crise ;
- *Observer les actions entreprises par les joueurs et en garder une trace* : il est alors nécessaire de les traduire en séquences d'actions ;
- *Comparer les actions des joueurs aux comportements nominaux* : afin de déterminer la justesse ou non de leurs actions, les types d'erreurs faites (*où*, *quand* et *comment*) ;
- *Identifier des comportements inadéquats récurrents* : soit constatés en situation réelle ou lors de session de jeux ;
- *Proposer des indicateurs qualitatifs et quantitatifs* mesurant l'adéquation des comportements des joueurs humains ;
- *Fournir des outils d'évaluation individuelle et collective*, ainsi qu'une évaluation durant la partie qui servira de base pour le support et l'aide aux apprenants.

En conséquence, le chapitre suivant présente un état de l'art centré sur les jeux sérieux (JS) et les systèmes tutoriels intelligents (STI), afin de rappeler d'une part les solutions existantes aux besoins identifiés dans I.3 ainsi que les lacunes qu'il convient de remplir pour renforcer la fonction d'apprentissage d'un jeu sérieux.

CHAPITRE II

Jeux sérieux et formation

Sommaire

II.1 Introduction 18
II.2 Les jeux sérieux 18
II.2.1 Quelques modélisations du concept de jeu sérieux 20
II.2.2 Quelques exemples de jeux sérieux 22
II.2.2.1 Cadre d'analyse 22
II.2.2.2 Un jeu sérieux pour l'apprentissage des langues 26
II.2.2.3 FORMAT-STORE Application à un supermarché virtuel . . 28
II.2.2.4 Un jeu sérieux pour les accidents de la route 31
II.2.2.5 Un jeu sérieux de formation des infirmières aux urgences . 33
II.2.2.6 Escape from Wilson Island (EFWI) 36
II.2.2.7 Autres travaux 38
II.2.3 Synthèse et conclusion 39
II.3 Les systèmes tutoriels intelligents 39
II.3.1 Définition et architecture 40
II.3.2 Quelques travaux dans le domaine des STIs 43
II.3.2.1 Cadre d'analyse 43
II.3.2.2 Help Agent for Learning (HAL) 43
II.3.2.3 PEGASE (PEdagogical Generic and Adaptive SystEm) . . 46
II.3.2.4 Helpful agent for safEty leaRning in virtuAl environment (HERA) 51
II.3.3 Synthèse et conclusion 55
II.4 Les SMA dans les jeux sérieux 55
II.4.1 Simulation 56
II.4.2 Architecture et intégration 57

II.5 L'évaluation des apprenants 57
II.6 Étude comparative 59
II.7 Conclusion 62

II.1 Introduction

Dans ce chapitre, nous verrons différents travaux dans le domaine des jeux sérieux et des systèmes tutoriels intelligents (STI). Nous verrons les points forts et les points faibles des jeux sérieux, notamment l'aspect modélisation et le suivi des apprenants et en quoi les STI peuvent répondre à ces points faibles. Dans la section II.2, nous verrons différentes architectures de jeux sérieux et nous présentons quelques travaux dans ce domaine. Dans la section II.3, nous présentons l'architecture d'un STI et nous présentons quelques travaux dans ce domaine. Dans la section II.4, nous illustrons l'intérêt des systèmes multi-agents (SMA) dans la conception et l'implémentation d'environnement virtuel pour l'apprentissage. Dans la section II.5, nous verrons le rôle de l'évaluation des apprenants et son importance dans les environnements virtuels pour l'apprentissage. Enfin dans la section II.6 nous comparons différents travaux entre STI et jeux sérieux et nous discutons des points forts et des points faibles des uns et des autres.

II.2 Les jeux sérieux

En reprenant la définition de Julian Alvarez [Alvarez, 2007] (section I.2), un jeu sérieux est une application informatique, et englobe ainsi la conception informatique avec les aspects de spécification, de modélisation, d'architecture et d'intégration. Cette application doit combiner à la fois l'aspect sérieux associé à l'apprentissage des apprenants (éducation, pédagogie, évaluation, etc.) et l'aspect ludique associé aux les jeux vidéo (mécanisme de jeux, 3D, immersion, etc.), et cette combinaison se fait via l'implémentation de scénario pédagogique. On peut donc représenter un jeu sérieux selon trois axes, un axe pour la conception informatique, un axe pour l'aspect apprentissage, et enfin un axe pour l'aspect ludique (jeu vidéo). Enfin ces axes reposent sur un socle qui représente l'architecture logicielle qui permet d'englober l'ensemble. Cette représentation est illustrée dans la figure II.1.

- **Conception :** l'axe de la conception relève du domaine de l'informatique et comprend tout ce qui concerne la gestion de projet définition des besoins, faisabilité, conception, implémentation, test, etc.
- **Apprentissage :** L'axe de l'apprentissage relève du domaine de l'éducation et l'enseignement, et correspond à la connaissance ou la compétence que l'on veut en-

seigner.

- **Jeux :** l'axe jeu relève du domaine du divertissement et correspond aux différents outils et techniques utilisés dans le domaine des jeux vidéo. Le jeux vidéo a cumulé plus de quarante ans d'expérience dans le divertissement et se traduit par des techniques 3D (immersion), mécanismes de jeu, interface homme-machine, etc.
- **Scénario pédagogique :** le scénario pédagogique a pour propriété de susciter l'envie d'apprendre ainsi que l'étude et la compréhension de l'apprenant.
- **Architecture logicielle :** le socle de l'architecture logicielle permet d'englober l'ensemble des axes et s'assurer du bon fonctionnement et l'interopérabilité des différents axes.

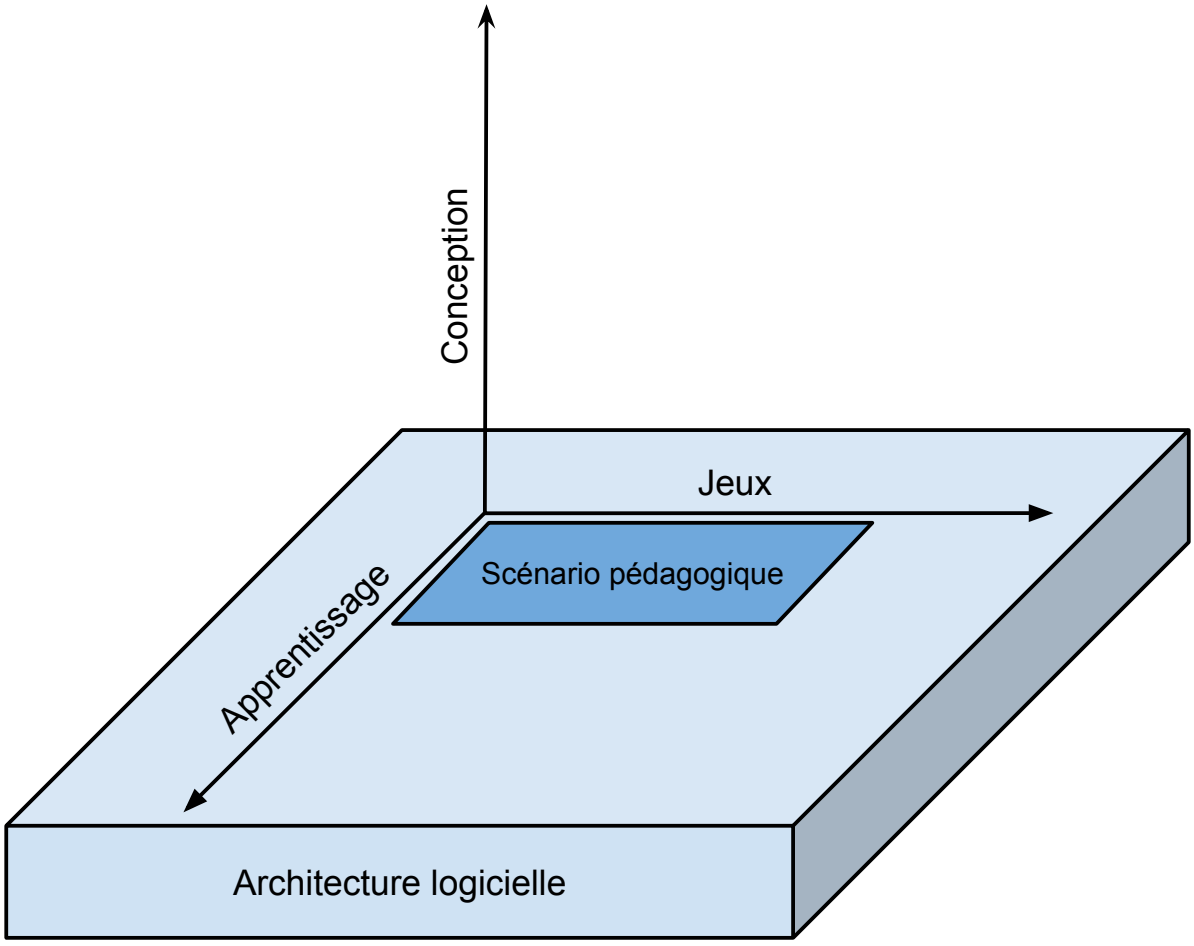


Figure II.1 – Le concept de jeu sérieux défini selon trois axes.

Comme nous l'avons vu dans la section I.2, il existe plusieurs définitions du concept de jeu sérieux, mais il existe aussi plusieurs modélisations de l'architecture des jeux sérieux. Dans la section suivante nous allons voir différents cadres conceptuels et modélisations de jeux sérieux et dans la section II.2.2, nous présentons quelques travaux de jeu sérieux.

II.2.1 Quelques modélisations du concept de jeu sérieux

Il existe plusieurs modélisations de l'architecture des jeux sérieux, une des plus citée dans la littérature est celle de Garris [Garris et al., 2002]. Garris définit un modèle simple retenu par la plupart des jeux sérieux. L'objectif est de concevoir un programme d'enseignement qui intègre certaines fonctions ou caractéristiques issues des jeux vidéo (mécanismes de jeu, environnement virtuel 3D, etc). Ces caractéristiques déclenchent un cycle qui comprend les jugements et les comportements des utilisateurs, ainsi qu'un mécanisme de rétroaction. L'intégration du contenu pédagogique avec les fonctionnalités du jeu se traduit par un cycle de jeu récurrent et auto-motivant pour l'apprenant. Cet engagement dans le jeu conduit à la réalisation des objectifs de formation. Ce modèle d'enseignement est illustré par la figure II.2.

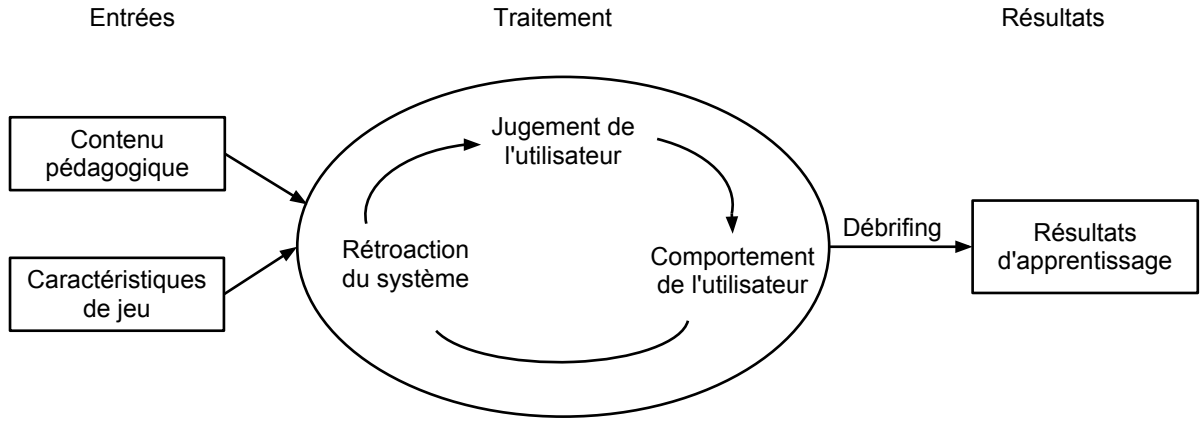


Figure II.2 – Modèle de jeux sérieux selon Garris [Garris et al., 2002]

Michael Zyda [Zyda, 2005] prend l'exemple du jeu sérieux American Army [Nieborg, 2004] qui est l'un des jeux sérieux les plus populaires et qui est à l'origine de l'expansion des jeux sérieux. Zyda présente un modèle de jeu sérieux basé sur une extension du jeu vidéo traditionnel, comme le montre la figure II.3, en intégrant la pédagogie (les activités qui éduquent et instruisent et confèrent ainsi une connaissance ou une compétence), cela permet de rendre les jeux plus sérieux (dans la mesure où cela facilite l'apprentissage des apprenants). La pédagogie doit toutefois être liée à l'histoire du jeu (scénario).

Il existe également des travaux qui présentent des cadres conceptuels (*framework*) pour la conception de jeux sérieux comme [de Freitas et Jarvis, 2006] et [Yusoff et al., 2009]. Dans [Yusoff et al., 2009], Yusoff propose un cadre conceptuel qui inclut la théorie de l'apprentissage, pédagogie en accord avec les exigences du jeu vidéo. Ce cadre conceptuel vise à établir un modèle conceptuel qui sera utilisé par le concepteur du jeu ou le professionnel de l'enseignement lors de la conception de jeux sérieux. Le

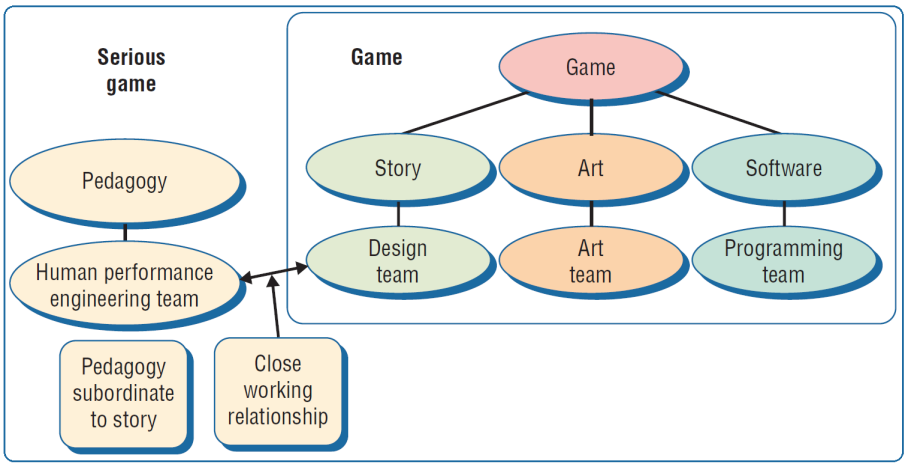


Figure II.3 – Du jeu vidéo au jeu sérieux [Zyda, 2005]

cadre conceptuel (présenté dans la figure II.4) est présenté par son concepteur comme une évolution du modèle de jeu entrée-processus-résultats proposé par Garris [Garris et al., 2002].

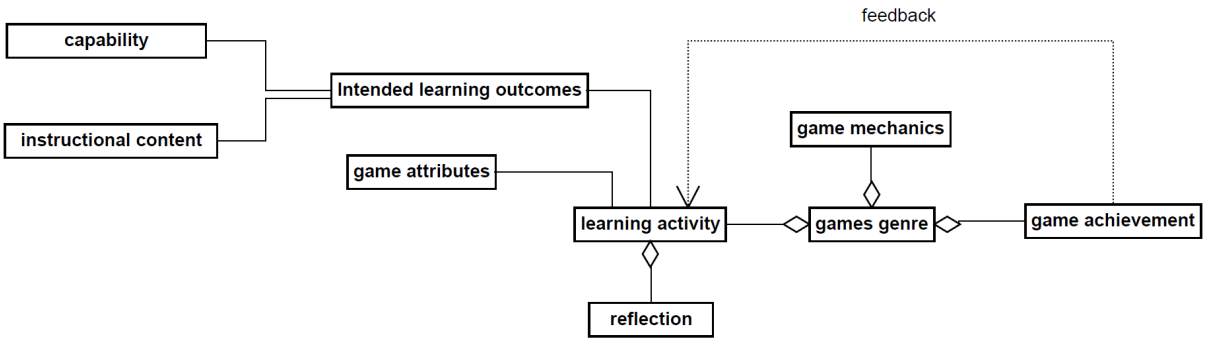


Figure II.4 – Diagramme de classes du cadre conceptuel présenté par Yusoff [Yusoff et al., 2009]

Dans [de Freitas et Jarvis, 2006], de Freitas décrit un processus de développement qui vise à englober les spécificités de l'apprenant et les résultats d'apprentissage ciblés afin de s'assurer de la réussite du jeu sérieux. Le cadre conceptuel met en évidence l'importance de quatre principaux aspects d'un jeu sérieux qui sont résumés par le **contexte** (permet de définir le type de jeu, ex : simulation jeu de rôle, etc.), la **spécification de l'apprenant** (défini les préférences et les particularités de l'apprenant ou groupe d'apprenants), **représentation de l'environnement** (défini le niveau d'immersion, la fidélité et l'interactivité du jeu), et enfin le **modèle pédagogique** (assurer le suivi et support des apprenants). Le cadre conceptuel proposé par de Freitas est décrit dans la figure II.5.

Dans certain travaux sur les jeux sérieux, la notion de jeu sérieux adaptatif est souvent utilisée. L'adaptation peut être définie comme une caractéristique exprimée au

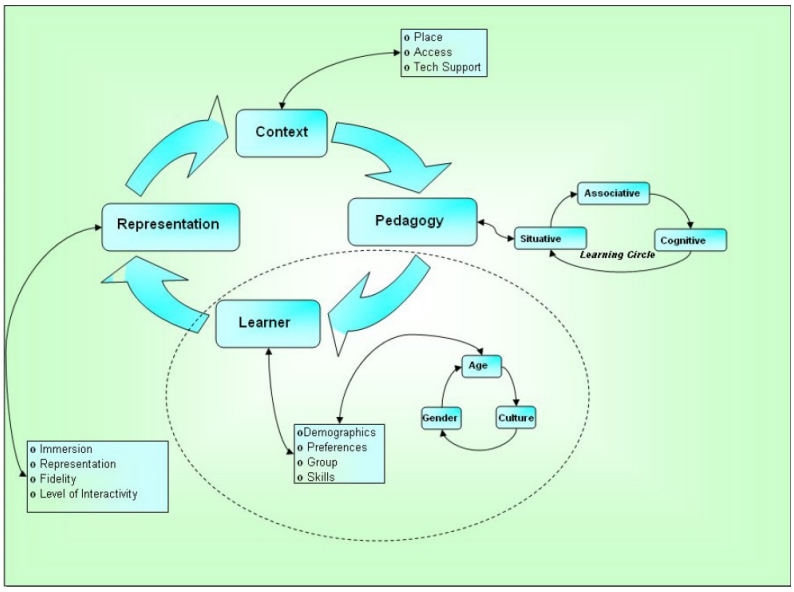


Figure II.5 – Le cadre conceptuel à quatre dimensions pour la conception de jeu sérieux [de Freitas et Jarvis, 2006]

niveau du jeu sérieux, qui reflète sa capacité à se modifier structurellement en réaction à certains évènements bien identifiés. [Hocine et al., 2011a] fait une synthèse générale de l'état de l'art sur les différentes approches sur l'adaptation dans les jeux sérieux. Cette notion d'adaptation peut être vue comme un modèle pédagogique qui permet de scénariser le processus de formation (enchainement des exercices, difficulté, rétroaction etc.).

Nous remarquons qu'il existe plusieurs approches de modélisation des jeux sérieux, mais dans chacune de ces approches on retrouve les concepts de base définis par Julian Alvarez qui sont le jeu, l'apprentissage, et le scénario pédagogique. Dans les sections suivantes nous présentons quelques exemples de jeux sérieux.

II.2.2 Quelques exemples de jeux sérieux

Dans cette section nous étudions quelques exemples de jeux sérieux et nous étudierons leur architecture ainsi que leur fonctionnement et plus particulièrement leur module d'évaluation.

II.2.2.1 Cadre d'analyse

Comme nous l'avons vu dans la section I.3, les jeux sérieux ont des points forts et des points faibles que nous avons représenté en tant que besoins d'apprentissage. Dans la section suivante nous allons étudier différents travaux dans le domaine des jeux sérieux

et des STIs, et en extraire les points forts et leurs faiblesses relatif à ces besoins d'apprentissage. Le cadre de notre étude consiste à analyser les différents travaux en termes d'architecture, de représentation des connaissances, et surtout du suivi et l'évaluation des apprenants (individuelle et collective).

Représentation des connaissances L'aspect représentation des connaissances consiste à analyser comment sont représentées les connaissances liées au domaine et à l'apprenant. Dans les jeux sérieux cet aspect est souvent négligé, surtout les connaissances du domaine étudié. Dans les STI la représentation des connaissances est modélisée par le modèle du domaine et le modèle de l'apprenant (voir section II.3). Par ailleurs, il existe plusieurs façons de modéliser les connaissances, certaines utilisent des ontologies, d'autres des méta-modèles ou d'autres types de modélisation (modèle Rule Skill Knowledge [Rasmussen, 1983], Concur Task Trees [Paternò et al., 1997], etc.).

Connaissances liées au domaine : Cette représentation va dépendre du domaine étudié. Par exemple, dans [Tchétagni et Nkambou, 2002], Tchétgani présente une représentation hiérarchique des connaissances. L'évaluation hiérarchique consiste à évaluer l'état des connaissances de l'apprenant à plusieurs niveaux de granularité. Cette représentation permet un diagnostic précis et flexible des compétences de l'apprenant. Nous pouvons aussi avoir d'autres types de représentations tel que les ontologies. Dans [Provitolo et al., 2009] et [Bénaben et al., 2008], les auteurs présentent respectivement une ontologie pour représenter le domaine du risque et les catastrophes naturelles et une ontologie pour la gestion de crise et la collaboration. Les ontologies peuvent apporter une certaine sémantique grâce à la définition de concepts et de relations qui permettent de produire un raisonnement sur l'ontologie. Cette ontologie constitue alors une première étape dans la constitution d'un modèle du domaine voire du modèle de l'apprenant (modèle « overlay »). Sans oublier les autres formes de représentation (réseau bayésien, diagrammes graphiques), citons également le recours à un méta-modèle comme dans [Buche et al., 2010].

Connaissances liées à la représentation de l'apprenant : Il existe plusieurs modèles de classification des connaissances, le plus utilisé en EIAH (Environnement Informatique pour l'Apprentissage Humain) est celui de Jacques Tardif [Tardif, 1993]. Tardif représente les connaissances avec les connaissances déclaratives, procédurales et conditionnelles. Les connaissances déclaratives sont essentiellement des connaissances théoriques. Il s'agit de la connaissance de faits, de règles, de lois, de principes. Les connaissances procédurales correspondent au "*comment*" de l'action, aux étapes pour réaliser une action ; il s'agit de séquences d'actions. Les connaissances conditionnelles

concernent le "*quand*" et le "*pourquoi*" des actions. Dans un autre registre, le modèle SRK (S-Skill based behaviour, R-Rule base behaviour, K-Knowledge based behaviour) de Rasmussen [Rasmussen, 1983] divise la connaissance de l'apprenant en trois niveaux :

- Niveau basé sur le savoir-faire : Automatismes sensori-moteurs, exécution sans contrôle conscient, situations familières.
- Niveau basé sur des règles : Procédures contrôlées par des règles intuitives, procédures acquises par l'apprentissage, situations familières.
- Niveau basé sur des connaissances : Procédures contrôlées par des buts, mécanismes de prise de décision, situation d'apprentissage, situations inhabituelles, exceptionnelles ou inconnues.

Évaluation des apprenants Pour évaluer l'apprenant, il faut aussi récupérer son activité de l'environnement de formation. Ce processus est appelé dans la littérature le recueil des traces. Dans [Baudouin et al., 2007], Baudouin et al. propose un modèle de recueil des traces dans un environnement virtuel pour les travaux pratiques. Le recueil de traces est basé sur le modèle MASCARET (Multi-Agent System for Collaborative, Adaptive and Realistic Environment for Training). MASCARET est une plate-forme logicielle permettant à plusieurs utilisateurs, apprenants et formateurs, d'agir ensemble dans un environnement virtuel. MASCARET représente les traces via un méta-modèle couvrant différents types de traces (communication, motrice, changement d'état, etc.). Il permet aussi, via des outils, de visualiser et d'analyser les traces sous forme de chronogramme afin de manipuler explicitement ces observables.

Dans les jeux sérieux impliquant un environnement virtuel, la notion de trace est multiforme (actions des joueurs, modalité d'interaction avec l'environnement virtuel, interactions entre joueurs etc.), et peut se dérouler dans un contexte distribué (multi-joueurs). Elle peut également soulever des enjeux techniques dans la capacité à obtenir les données traduisant ces traces.

L'aspect suivi et évaluation des apprenants consiste à analyser comment les actions et les connaissances des apprenants sont collectées et évaluées, et cela de manière individuelle et collective.

Comme nous l'avons vu dans la section I.3.2.1, L'évaluation dans les jeux sérieux est une problématique de recherche en soit [Thomas Benjamin et al., 2012] [Nieborg, 2004]. L'évaluation des apprenants est plus particulièrement problématique dans le cadre de l'apprentissage de tâches collaboratives. L'apprentissage de tâches collaboratives vise à apprendre à un groupe d'apprenants à travailler ensemble pour améliorer l'apprentissage individuel et les résultats de leur collaboration. Woolf [Woolf, 2010] fait un tour d'horizon des environnements informatiques contribuant à ce domaine selon

quatre niveaux de soutien à la collaboration [Soller, 2006] :

- **Structuration de la collaboration :** soit comment constituer l'équipe qui va collaborer (choix d'un partenaire par exemple).
- **Retour d'information (mirror) sur la collaboration :** ou comment rendre compte aux acteurs de la dynamique de leurs interactions au travers de tableaux de bord graphiques (par ex. un graphe d'interaction [Kay et al., 2006]).
- **Analyse métacognitive :** il s'agit d'analyser, voire de diagnostiquer les interactions et l'avancement des apprenants. Dans le projet COMET [Suebnukarn et Haddawy, 2004] des étudiants en médecine doivent diagnostiquer un malade via un ensemble de symptômes. Durant leur diagnostic, le système agrège les données d'interaction des apprenants en indicateurs de haut niveau et les en informe.
- **Guidage dans la collaboration (coaching) :** ceci peut revenir à exploiter les résultats précédents pour expliquer et guider les acteurs sur leur processus collaboratif. Ainsi le projet COLER [de los Angeles Constantino-Gonzalez et al., 2003] (dans le cadre de l'apprentissage de la conception de base de données) détecte les opportunités d'interaction dans un groupe (différence de résultats entre groupes, niveau d'implication, et qualité des solutions) et peut faire des recommandations d'interaction.

On retrouve dans les jeux sérieux le même besoin d'évaluer ou d'apprécier le niveau des connaissances acquises par un ou plusieurs joueurs/apprenants. Il est à noter que ces évaluations considèrent dans la majorité des cas la validation de compétences homogènes y compris dans un contexte collaboratif. Que ce soit dans un système mono apprenant ou d'apprentissage de groupe, le parcours de chacun des participants peut varier, mais l'objectif de formation est unique même lorsqu'il intègre la dimension collective. La question d'évaluer à la fois individuellement et collectivement des compétences hétérogènes est peu abordée.

Architecture L'aspect architecture consiste à étudier la nature de l'architecture utilisé pour la conception du jeu sérieux (ou STI). Comme l'avons vu dans la section II.2.1, il existe plusieurs modélisations du concept de jeu sérieux. De plus certains jeux sérieux utilisent des moteurs de jeu tel que Unreal Ungine [1], Unity 3D [2] (voir annexe A), etc. De même pour les STIs, certains utilisent l'architecture de base d'un STI et d'autres ajoutent d'autres composantes liées au domaine (module du risque [Amokrane, 2010]) ou bien à l'objectif de formation (module d'erreur [Buche, 2005]).

1. http ://www.unrealengine.com/
2. http ://unity3d.com/

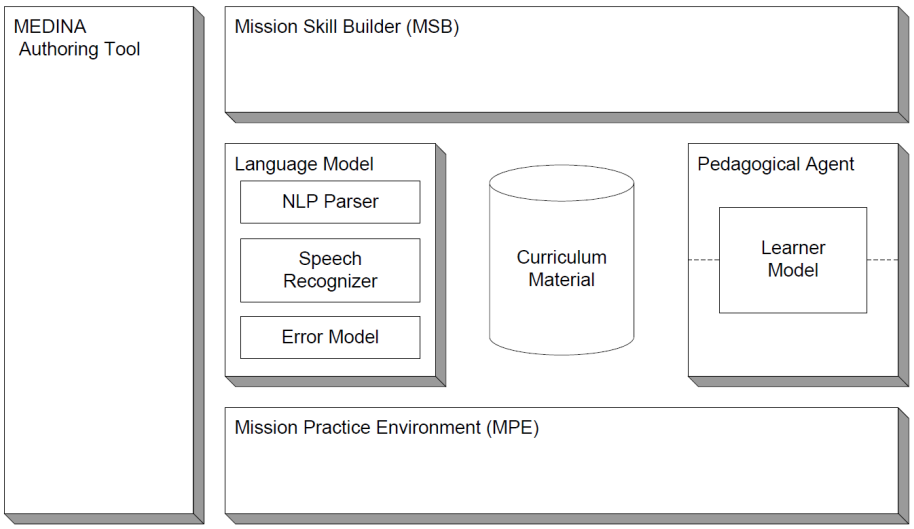


Figure II.6 – Architecture de TLCTS [Johnson et al., 2004]

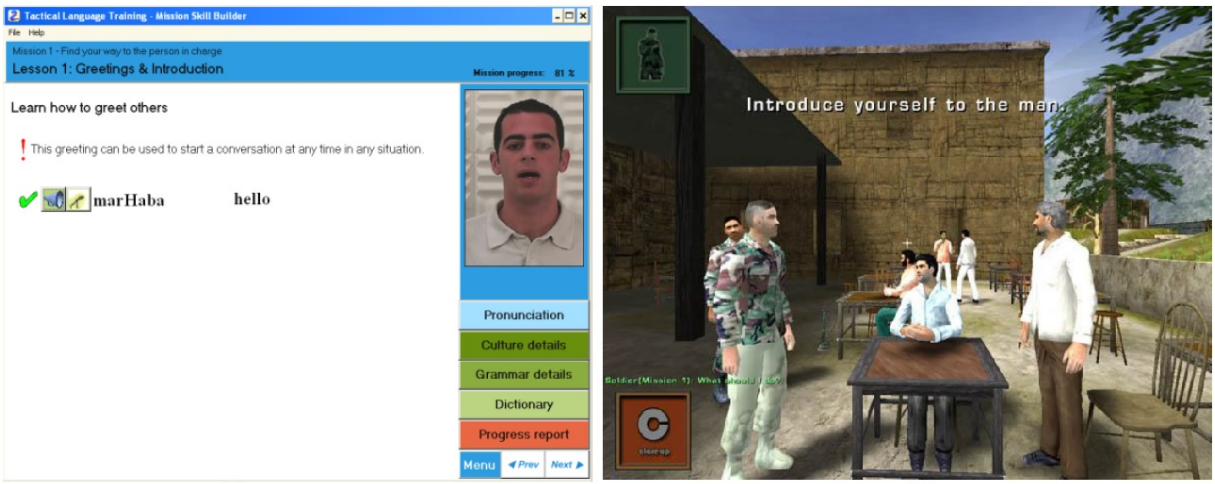
II.2.2.2 Un jeu sérieux pour l'apprentissage des langues

Le système de formation en langue et culture (TLCTS) [Johnson et al., 2004] est un environnement interactif pour l'apprentissage des langues et de la culture. TLCTS est conçu pour aider les gens à acquérir rapidement des compétences de communication orale. C'est un jeu sérieux, combinant des expériences de jeu interactives ainsi que des leçons interactives.

Architecture L'architecture du système TLCTS répond à plusieurs exigences internes. Un *modèle de l'apprenant* prend en charge les requêtes d'exécution et de mises à jour à la fois par *le constructeur des compétences* (Skill Builder) et de *l'environnement d'entrainement* (Practice Environment). Les apprenants doivent être en mesure de basculer facilement entre le constructeur des compétences et l'environnement d'entrainement. Le système prend en charge la création rapide de nouveaux contenus par des équipes d'experts et de développeurs du jeu. TLCTS est basé sur une architecture distribuée et logique, cette architecture est illustrée dans la figure figure II.6.

L'agent pédagogique suit la performance de l'apprenant, et utilise les données de cette performance à la fois pour suivre les progrès de l'apprenant dans la maîtrise des compétences ainsi que pour sélectionner la rétroaction adéquate à la situation. Le profil de compétences de l'apprenant est enregistré dans un *modèle de l'apprenant*.

Le Mission Skill Builder (MSB) (figure II.7 (a)) est un environnement d'apprentissage tous en un qui permet à l'apprenant d'acquérir des compétences sur le vocabulaire orienté mission ainsi que sur la formation de la prononciation et de la reconnaissance gestuelle. Dans cet environnement d'apprentissage, l'apprenant développe les compétences nécessaires pour accomplir des missions spécifiques.



(a) (b)

Figure II.7 – Capture d'écran du projet TLCTS [Johnson et al., 2004].
(a) : Le Mission Skill Builder.
(b) : Le Mission Practice Environment.

L'environnement d'entrainement aux missions (Mission Practice Environment, MPE, figure II.7 (b)) est responsable du rendu visuel et de la simulation 3D qui doit reproduire de façon réaliste et fidèle les situations sociales (basé sur le moteur de jeu Unreal Ungine [3]). Dans le MPE, l'apprenant peut interagir avec des personnages non-joueurs en parlant et en sélectionnant des gestes.

Représentation des connaissances La représentation des connaissances dans TLCTS a consisté à représenter la connaissance du domaine linguistique, comme la prononciation, le vocabulaire, un modèle de langue a été défini. Le modèle de langage comprend une reconnaissance de la parole. Un module de traitement de langage naturel permet d'annoter les phrases avec des informations structurelles et les faire correspondre aux explications grammaticales et au modèle d'erreur qui détecte et analyse les erreurs syntaxiques et phonologiques.

Le modèle de l'apprenant est implémenté comme un ensemble de règles d'inférence qui sont mises à jour dynamiquement via une base de données SQL. Le *modèle de l'apprenant* enregistre le nombre de tentatives réussies ou non pour chaque action au cours d'une session d'entrainement.

Évaluation des apprenants L'évaluation des apprenants se fait à l'aide du modèle d'erreur qui répertorie un ensemble d'erreurs linguistiques. Ainsi les erreurs enregistrées par le modèle de l'apprenant serviront à estimer la maîtrise de l'apprenant pour chaque élément de vocabulaire et de la compétence communicative, et de déterminer

3. http ://www.unrealengine.com/

quel type de rétroaction est plus appropriée pour une situation donnée. Un tuteur virtuel fournit alors une rétroaction personnalisée pour améliorer et accélérer le processus d'apprentissage.

Du fait que l'apprentissage se fait de manière individuelle, l'évaluation des apprenants se fait aussi de manière individuelle, durant la partie (évaluation formative) afin de servir de base pour le module pédagogique pour aider l'apprenant, et en fin de jeu (évaluation diagnostique) pour déterminer les missions maîtrisées et ceux qui ne le sont pas.

Conclusion TLCTS est présenté comme un jeu sérieux mais présente beaucoup de composants d'un système tutoriel intelligent (modèle du domaine et de l'apprenant, agent pédagogique). Néanmoins cet outil reste très proche d'un jeu sérieux du fait de sa conception basée sur un moteur de jeu vidéo (l'Unreal Engine). L'évaluation se fait de manière individuelle durant et après l'exercice et se base sur un modèle d'erreur qui répertorie des erreurs linguistiques. On peut noter que l'apprentissage se fait de manière individuelle et aussi l'absence d'apprentissage et d'évaluation collective.

II.2.2.3 FORMAT-STORE Application à un supermarché virtuel

Le projet FORMAT-STORE [Mathieu et al., 2011] est destiné à la formation d'étudiants en école de commerce. Il intervient en complément d'une formation traditionnelle regroupant sur une plate-forme pédagogique un ensemble de situations-problèmes rédigés par des professionnels de la vente, et relatives à la gestion de la relation clientèle, des stocks ou du magasin. Ces situations-problèmes se présentent en pratique sous la forme de dialogues interactifs entre un vendeur (l'apprenant) et un client, où chaque réplique du vendeur donne lieu à un score et un compte-rendu.

FORMAT-STORE a pour ambition de proposer une mise en contexte des situations-problèmes, permettant à l'apprenant d'appliquer dans un magasin virtuel les connaissances acquises de manière traditionnelle sur la plate-forme, et d'expérimenter diverses façons de traiter un cas. En outre, il permet en parallèle l'apprentissage de compétences liées à la gestion du temps ou la priorisation des tâches. La volonté est de proposer à l'apprenant une expérience immersive dans laquelle des clients virtuels aux profils différents simulent l'activité et les problèmes de clients réels. Par le biais de son avatar, l'apprenant assume le rôle d'un vendeur dans le magasin (voir figure II.8 (a)).

Architecture FORMAT-STORE est basé sur l'architecture agent IODA [Kubera et al., 2008]. IODA est aussi une méthodologie de conception orientée interactions (interactions entre les agents) dont l'originalité est d'accorder une tangibilité logicielle aux

interactions et de proposer une conception unifiée de la simulation multi-agents.

Le comportement d'un agent est défini par une disposition des blocs sémantiques appelés interactions. Une interaction est elle-même un ensemble de primitives impliquant simultanément un nombre fixe d'agents, qui décrit comment et sous quelles conditions les agents peuvent interagir les uns avec les autres ou avec l'environnement. Un agent possède un ensemble de primitives de perception utilisées pour obtenir des informations de l'état global de la simulation et un ensemble de primitives d'action utilisées pour changer cet état global (changement de l'environnement, son état ou l'état d'un autre agent). Ce sont les éléments atomiques d'interactions.

Représentation des connaissances La représentation des connaissances se fait via une matrice d'interaction entre les différents agents II.1. Une fois les familles d'agents identifiées, l'étape suivante dans la méthodologie IODA consiste en la description de leurs interactions. Ceci implique également leur allocation dans une matrice des interactions. IODA permet également, à travers une seconde matrice appelée la matrice de mise à jour, la prise en compte des changements d'états qui ne relèvent pas d'une interaction particulière (vieillissement d'un agent, usure, etc). La matrice des interactions présente l'avantage d'être lue très naturellement par un non informaticien. Malgré son apparente simplicité, elle décrit le comportement de chaque agent de manière exhaustive. En outre, exploitée par le mécanisme de sélection de l'action présenté dans la section suivante, elle est fonctionnellement équivalente à tout type de représentation du comportement. Les interactions dans FORMAT-STORE sont décrites dans le tableau II.1.

Source		Vendeur	Client	Porte	Affichage	Caisse	FileAttente	Article	Tache	Carton
Vendeur			Dialoguer(1,0)					Ôter(5,0) Réapprovisionner(5,0)	Nettoyer(3,0)	Ranger(3,0)
Client	Errer(0) AllerVers(1)	Repérer(10,9) Patienter(2,3)		Sortir(1,8)		Payer(5,6)	Entrer(2,4) SePlacer(1,5) Sortir(1,7)	Prendre(5,2)		
Porte	GénérerClient(1)		Informer(10,0)							
Affichage			Informer(10,0)							
Caisse			Informer(10,0) Encaisser(2,1)							
FileAttente										
Article	Périmer(3) GénérerTache(4) GénérerCarton(4)		Informer(10,0) Contrarier(1,1) AvertirRupture(1,2)							
Tache			Contrarier(1,0)							
Carton			Contrarier(1,0)							

Tableau II.1 – La matrice des interactions entre les différents types d'agents [Mathieu et al., 2011].

Évaluation des apprenants Dès son entrée dans le magasin, chaque client se voit attribuer une liste d'articles à acheter. De manière autonome, il se déplace dans le magasin à la recherche de ces articles, qu'il place ensuite dans son panier. Une ou plusieurs

situations-problèmes peuvent également être affectées à un client, ce qui se traduit par la nécessité de trouver le vendeur puis d'engager un dialogue. En situation conversationnelle, comme l'illustre la figure II.8 (b), l'apprenant doit sélectionner les réponses correctes aux questions des clients virtuels. À la fin du dialogue, et indépendamment des réponses du joueur, le client retourne à son activité. D'autres situations-problèmes peuvent ainsi être cumulées durant l'activité du client, ayant pour cause des perturbations introduites volontairement (une allée peut être encombrée) ou non (un article peut se trouver en rupture, leur quantité étant limitée).



Figure II.8 – Capture d'écran du projet FORMAT-STORE [Mathieu et al., 2011].
(a) : Évolution de l'apprenant dans un magasin à travers un avatar.
(b) : Les dialogues des situations-problèmes.

L'évaluation de l'apprenant se fait via ces réponses aux dialogues avec les clients ainsi qu'un score lié à la satisfaction du client à la sortie du magasin, ce score est basé sur différentes variables comme la disponibilité des produits recherchés, temps d'attente, etc. Dans FORMAT-STORE, l'apprentissage est individuel ainsi que l'évaluation des apprenants.

Conclusion Dans FORMAT-STORE, l'évaluation de l'apprenant prend en compte deux variables pour chaque client. Premièrement, l'apprenant est évalué immédiatement à l'issue de chaque dialogue avec un client, et en fonction de ses réponses. Deuxièmement, et indépendamment, le score du joueur prend aussi en compte le niveau de satisfaction de chaque client qui sort du magasin. La satisfaction d'un client décroît par exemple lorsqu'un article désiré est manquant ou une allée obstruée. L'évaluation se fait de manière individuelle et en temps réel, et utilise une formule de calcul assez simple. Dans [Mathieu et al., 2011], le travail proposé se concentre plus sur l'aspect adaptabilité des PNJs (personnages non joueurs) et de l'environnement basé sur l'architecture multi-agent IODA.



Figure II.9 – Exemple de vue 3D de l'environnement [Binsubaih et al., 2006]

II.2.2.4 Un jeu sérieux pour les accidents de la route

Binsubaih [Binsubaih et al., 2006] propose un jeu séreux pour la formation des policiers novices pour les interventions sur les incidents de la sécurité routière. Dans le domaine de la formation sur la sécurité routière, il y a trois domaines d'apprentissage : les connaissances, les compétences et les attitudes. Dans ce jeu sérieux, l'accent est mis sur les connaissances et les compétences. L'objectif pédagogique du jeu sérieux est de permettre aux apprenants novices d'effectuer les procédures suivantes :

- Rechercher et identifier les indices, en marquant leur position.
- Stationner le véhicule de patrouille à un endroit approprié pour aider à sécuriser la scène de l'accident et avertir les autres véhicules.
- Utilisez des cônes de signalisation pour sécuriser le lieu de l'accident.
- Photographier la scène de l'accident (véhicules, indices, scène).
- Prendre des mesures nécessaires pour permettre la reconstruction de la scène de l'accident.
- Dessiner la scène de l'accident.

Architecture

Le jeu est basé le moteur Torque, utilisé pour éditer et afficher l'interface du jeu. L'environnement du jeu est construit en utilisant un éditeur de mission fourni par le moteur de jeu. Le comportement est contrôlé à partir de l'espace de jeu qui reçoit les mises à jour et envoie les actions au moteur du jeu. L'interface graphique est décrite dans la figure II.9.

Représentation des connaissances L'architecture stocke un modèle du jeu en utilisant des ontologies. Cela rend plus facile la construction et la compréhension des événements. Le but est de permettre aux experts du domaine de construire et d'unifier les terminologies utilisées dans le domaine de la sécurité routière.

L'état du jeu réside à l'intérieur de la composante spatiale du jeu et contient des objets qui ont une représentation virtuelle dans le monde du jeu (c'est à dire les objets actifs). Ces objets sont appelés des objets "*mappés*" et ont besoin d'avoir le même ID unique utilisé dans le monde du jeu.

Les données enregistrées pour l'analyse des actions de l'apprenant sont enregistrées dans le temps et sont représentées par les actions suivantes : la navigation, la sélection d'objets, la photographie, les objets en mouvement, l'ajout d'objets, mesure, les interactions des ressources, et les questions. Ces actions sont utilisées après pour le débriefing et sont également stockées dans un profil enquêteur. Après certaines actions, une boîte de dialogue apparaît et demande à l'apprenant d'enregistrer la raison pour laquelle l'action a été prise. Ceci afin d'aider le formateur à comprendre le comportement de l'apprenant.

Évaluation des apprenants La performance de l'apprenant a été mesurée sur la base de la réussite des tâches suivantes (barème de notation établi par deux formateurs) :

- Assurer la scène de l'accident :
  - stationnement de la voiture de police (10,5 %),
  - mise en place des cônes (7 %),
- Photographier le lieu de l'accident (17,5 %).
- Prise de mesures appropriées (21 %).
- Placement des marqueurs à des indices importants (14 %).
- Dessiner la scène de l'accident (30 %).

Le scénario de l'accident implique une collision entre deux véhicules. L'un des conducteurs est légèrement blessé. L'apprenant a 30 minutes pour terminer son investigation. Avant que la session commence l'apprenant est informé qu'il recevra deux rappels : un bout de 15 minutes et un autre à 5 minutes de la fin, ces rappels lui spécifient que le dessin de l'accident doit être réalisé avant la fin des 30 minutes. Après la fin de la session, l'apprenant passe par l'auto-évaluation, ceci en faisant apparaître une liste d'actions à faire afin de réaliser les différentes tâches, et l'apprenant est invité à cocher les actions accomplis. Le système génère ensuite les résultats. L'apprenant a 10 minutes pour analyser les résultats avant de commencer une nouvelle session. Les participants qui ont eu un score inférieur à 70 % sont invités à recommencer la session de formation une fois de plus.

Une session de l'exercice est composée de deux parties : une épreuve écrite et une épreuve de dessin. L'épreuve écrite est composée d'une courte explication de la façon dont l'accident s'est produit. L'épreuve de dessin consiste à dessiner une scène 2D de l'accident, et une série de questions. Dans le test de dessin l'apprenant donne une description d'un accident différente de celui utilisé pour l'épreuve écrite, et est autorisé à dessiner la scène de l'accident en examinant l'environnement 3D.

Conclusion Ce jeu sérieux est basé principalement sur des techniques de jeu vidéo (moteur de jeu, mécanisme de jeu, éditeur, etc.), et la solution apportée au niveau de l'évaluation est assez simple et dépend beaucoup du domaine et du jeu en lui même. La solution retenue permet malgré tout de proposer un environnent virtuel complet pour la formation des policiers novices en sécurité routière.

II.2.2.5 Un jeu sérieux de formation des infirmières aux urgences

Dans [Vidani et Chittaro, 2009], Vidani propose un jeu sérieux pour la formation des infirmiers(ères) au traitement de cas d'urgence pour les personnes âgés. Ce jeu sérieux permet l'apprentissage de procédures de première urgence. L'originalité de ce travail est la modélisation des tâches avec la modélisation CCT (Concur Task Trees) [Paternò et al., 1997].

Architecture : Le jeu a été développé en utilisant NeoAxis, un moteur de jeu basé sur le moteur de rendu Ogre. Neoaxis fournit un ensemble d'outils pour l'édition des ressources (modèles 3D, cartes, interfaces graphiques,...). Le moteur fourni aussi des algorithmes basiques d'IA pour l'animation des personnages, la simulation physique, les cut-scènes ainsi qu'un gestionnaire de contrôleur (manette). Le moteur du jeu est développé en C# .

Représentation des connaissances : Le jeu sérieux utilise la représentation CTT pour représenter le domaine étudié. Les principales caractéristiques de CTT sont : une structure hiérarchique, utile pour décomposer le problème en petites parties ; une syntaxe graphique qui permet de représenter les tâches par un graphe ; et enfin, la notation concurrente, les tâches sont liées entre elles (tâches sœurs) dans l'arbre par des opérateurs temporels. Une tâche CTT [Paternò et al., 1997] est défini par :

- Nom : utilisé pour identifier la tâche.
- Type : il y a quatre types possibles : abstraite, utilisateur, application, interaction.
- Sous-tâche de : nom de la tâche mère.

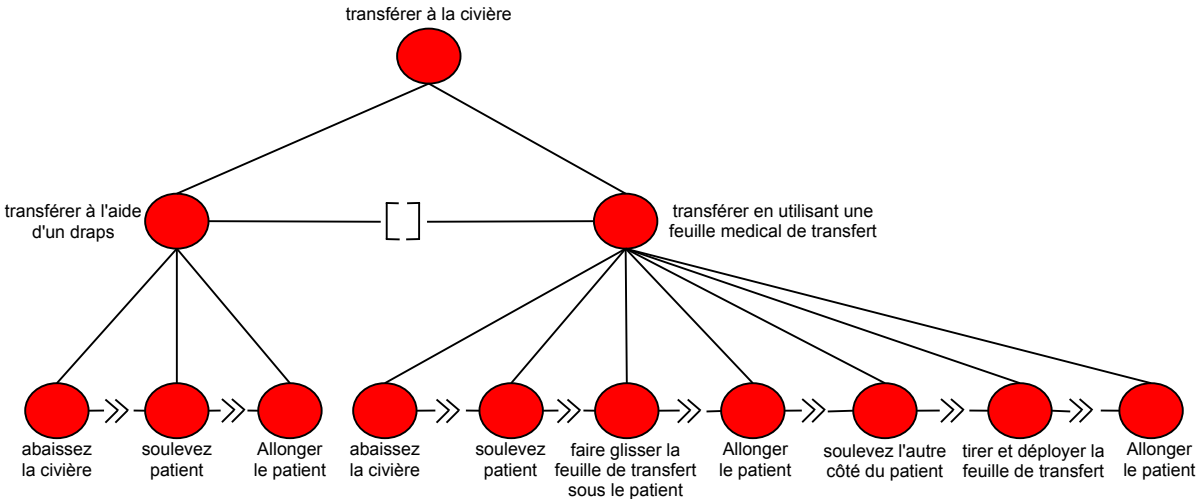


Figure II.10 – Extrait d'un arbre CTT [Vidani et Chittaro, 2009].

- Vecteur d'objets, chaque élément définit par : nom, type (interne, perceptible), liste d'objet en entrée, liste d'objet en sortie.
- Itératif : une valeur booléenne indiquant si la tâche est itérative.
- Première action : L'ensemble des actions initiales possibles.
- Dernière action : L'ensemble des actions finales possibles.

Dans [Vidani et Chittaro, 2009], Vidani défini une structure de données qui peut être construite automatiquement à partir d'un modèle CTT au format XML. La structure de données est par la suite mise en œuvre en C#, sous forme d'arbre récursif. Les feuilles représentent les actions que les joueurs peuvent effectuer dans le jeu, tandis que les nœuds internes ne servent qu'à regrouper les nœuds enfants. Cela signifie que les actions décrites dans le scénario du jeu seront représentées par les feuilles du modèle CTT, tandis que les nœuds internes seront utilisés par les concepteurs pour grouper les actions du joueur dans une structure logique. La figure II.10 montre un extrait d'un arbre CTT d'une procédure médicale.

Évaluation des apprenants :

Le jeu a trois objectifs principaux de formation :

- à travers une procédure d'urgence médicale organisée en tâches, l'objectif est de former l'apprenant aux tâches à effectuer, dans quel ordre, le timing, et les conditions dans lesquelles effectuer correctement la procédure ;
- montrer à l'apprenant comment les tâches sont effectuées, à travers des animations de personnages et d'objets dans le jeu ;
- permettre à l'apprenant de découvrir et de se familiariser avec les situations d'urgence.



Figure II.11 – Capture d'écran du jeu médical [Vidani et Chittaro, 2009]

Pour enseigner une procédure, le jeu se déroule comme suit. Pendant le jeu, les joueurs peuvent explorer l'environnement virtuel et choisir une tâche à accomplir parmi une liste de tâches possibles. La sélection d'une tâche par le joueur déclenche des événements spécifiques dans le jeu, par exemple l'exécution de certaines animations ou un effet visible sur le patient (par exemple, en pâlissant). La liste est remplie avec les tâches prises à partir du modèle CTT. Le modèle CTT est construit lors de la phase de conception, à partir de scénarios écrits en collaboration avec un expert du domaine.

La difficulté du jeu est de choisir la bonne tâche à effectuer au bon moment parmi les différentes tâches nécessaire pour réaliser la procédure. En utilisant la méthode décrite ci-dessus pour la simulation du comportement dynamique du modèle, le jeu ne permet pas au joueur d'exécuter une tâche qui ne répond pas aux contraintes imposées par les opérateurs temporels du modèle CTT. En ce sens, la sélection de tâche n'est pas une difficulté pour le joueur, car n'importe quel choix de tâche est correct.

Permettre de proposer des choix erronés rendrait le jeu plus difficile et offre d'autres possibilités d'apprentissage au joueur. Pour cette raison la possibilité de choisir des tâches qui ne satisfont pas les contraintes temporelles a été ajoutée. Le nombre de tâches proposées est déterminée selon le niveau de difficulté du jeu, identifiée par un nombre entier allant de 0 (le plus facile) à k (plus difficile), ce critère est définit lors de la conception et correspond au nombre maximum de niveaux de difficulté du jeu.

Conclusion : Dans ce jeu sérieux, les auteurs s'intéressent particulièrement à la modélisation du domaine avec la modélisation CCT. Mais en contrepartie délaisse l'aspect suivi et évaluation des apprenants. Néanmoins, l'ajout d'un niveau de difficulté donne

une dimension d'évaluation qui permet d'équilibrer l'apprentissage des apprenants.

II.2.2.6 Escape from Wilson Island (EFWI)

Le jeu sérieux Escape from Wilson Island [Wendel et al., 2012] s'inscrit dans l'apprentissage collectif assisté par ordinateur (Computer Supported Collaborative Learning, CSCL) [Stahl et al., 2006]. L'approche présentée dans [Wendel et al., 2012] propose la conception de scénarios d'apprentissage collaboratifs basées sur un jeu sérieux multijoueurs. Cette approche vise à combiner des concepts dans les domaines de l'apprentissage collaboratif (multi-joueurs) ainsi que la conception de jeu (Game Design). Cette approche prend en compte les exigences des jeux traditionnels solo (divertissement, la narration, l'immersion, graphisme, sons, etc), les défis de jeux multi-joueurs (jeu simultané, interaction) et la conception de jeux sérieux (contenus d'apprentissage, d'adaptation et de personnalisation). En outre, les exigences de l'apprentissage collaboratif sont considérés comme les objectifs du groupe, l'interdépendance positive, et la responsabilité individuelle.

Le jeu sérieux Escape From Wilson Island (EFWI) a été créé en utilisant le moteur de jeu Unity 3d. Le jeu peut être considéré comme un jeu 3D d'action aventure à la 1er/3ème personne (figure II.12). L'accent a été mis sur le "*gameplay*" collaboratif, y compris les compétences sociales comme le travail d'équipe, la coordination et la communication entre les joueurs. L'objectif des joueurs est de communiquer et de collaborer pour survivre et s'échapper d'une île déserte.



Figure II.12 – Capture d'écran du jeu EFWI [Wendel et al., 2012]

Architecture : Les auteurs ont essayé de développer un jeu sérieux permettant le travail coopératif tel que définit par Johnson [Johnson et Johnson, 1987] : interdépendance positive, responsabilité individuelle, interactions promotives, compétences

sociales et traitement de groupe. Cette approche se base sur les concepts suivants :

- **But commun/Succès :** Le but du jeu doit être conçu de telle manière que le succès est synonyme de réussite pour tous les joueurs. Le succès ou l'échec d'une partie ne peut dépendre que d'un seul joueur.
- **Ressources hétérogènes :** Chaque joueur doit avoir un ou plusieurs outils (ou capacités) uniques lui permettant d'effectuer des tâches uniques dans le jeu.
- **Des ressources personnelles qui s'épuisent (rechargeables) :** Afin de créer une certaine tension, les auteurs ont ajouté certaines ressources rechargeable (par exemple le niveau de santé ou un indice pour représenter la faim) qui s'épuisent lentement durant le jeu. En introduisant ce genre de ressource, les joueurs peuvent s'aider les uns les autres et cela peut créer une tension générale vers le succès, mais aussi une tension entre les joueurs et par conséquent, un lieu d'interaction avec une certaine dimension de prise de décisions.
- **Ressources collectables et négociables :** les auteurs ont aussi introduit des ressources dans le monde du jeu nécessaire pour gagner la partie. Ces ressources peuvent être échangeables afin de créer un espace pour la négociation ou encore la collaboration (par exemple, donner une ressource à un autre joueur pour le bien commun de l'équipe).
- **Tâches collaboratives :** Si toutes les tâches pouvaient être résolues par un seul joueur, il n'y aurait pas besoin de collaborer. Aussi il y a des tâches qui peuvent être résolues que si les joueurs agissent ensemble. Ces tâches peuvent inclure les ressources hétérogènes pour créer un besoin pour certains joueurs de participer aux tâches de l'équipe. Cela peut provoquer un besoin de discussion entre les joueurs lorsque le groupe dépend d'un individu.
- **Communication :** Il a été montré que la communication est essentielle pour l'apprentissage collaboratif. Ainsi le jeu doit offrir un moyen aux joueurs de communiquer (par exemple un système de "*chat*", communication vocale).
- **Système d'aide Ingame (durant la partie) :** Le jeu doit fournir une aide aux joueurs quand ils sont coincés. Le plus simple est une fenêtre qui s'affiche (popup) lorsque le joueur ne parvient pas à réaliser une tâche (ou qu'il met trop de temps à réaliser la tâche), le système d'aide peut aussi être déclenché par le joueur. Une manière plus immersive, (car il n'interrompt pas le flux du jeu) serait d'inclure l'aide dans le jeu lui-même, par exemple, avec un personnage du jeu (PNJ) qui fournir une aide en cas de besoin.
- **Tableau de bord :** Un tableau de bord doit montrer à la fois les efforts individuels et les efforts de l'équipe à la fin de la partie. Cela peut aider les joueurs à juger de la réussite globale (par exemple par comparaison avec d'autres équipes ou des tentatives précédentes) et la contribution de chaque joueur à la performance de l'équipe. Le score individuel peut fonctionner comme un facteur de motivation

pour les actions égoïstes qui rendent la collaboration pas si évidente que ça.
- **Système d'échange :** les joueurs peuvent échanger des objets entre eux. Cela permet de créer un espace pour prendre des décisions pour ou contre la collaboration.

Évaluation des apprenants : Dans EFWI, l'évaluation des apprenants ne se fait pas de manière automatique, néanmoins les auteurs ont pu mesurer le nombre de succès ou enregistrer les discussions des joueurs, et ont pu faire quelques observations intéressantes :
- Toutes les équipes ont pu réaliser au moins une partie, sinon la totalité des objectifs donnés durant les 30 min de la partie durant lesquels elles ont eu à collaborer.
- Les joueurs ont échangé sur les problèmes à résoudre dans le jeu, discutant ainsi de leur relation de travail, ils ont pu aider et promouvoir la réussite de chacun.

Bien que les améliorations des habiletés sociales n'ont pas pu être mesurées tout comme le travail d'équipe et la coordination ou la communication, il a été observé que l'ensemble de ces compétences ont été utilisées par les joueurs tout au long des parties. Cela indique qu'il est possible d'améliorer spécifiquement ces compétences dans la conception du jeu.

Conclusion : Dans [Wendel et al., 2012], Wendel présente une approche de jeux sérieux prenant en compte l'apprentissage collaboratif. Cette approche combine les techniques d'apprentissage collaboratif et celles des jeux sérieux ainsi que l'aspect multi-joueurs. Cette approche a été mis en œuvre dans le JS EFWI, un JS qui permet à un groupe de joueurs de collaborer et d'échanger entre eux afin de survivre et de s'échapper d'une île desserte. Malgré l'absence d'évaluation quantitative et qualitative des compétences individuelles (actions) et collectives (social), les observations et les résultats de l'expérience ont montré qu'il était possible de concevoir un jeu sérieux multi-joueurs qui répond aux exigences des principes de l'apprentissage collaboratif.

II.2.2.7 Autres travaux

Parmi les jeux sérieux les plus populaires au monde on peut citer America's Army (AA). AA est à l'origine de l'expansion des jeux sérieux (vers le début des années 2000). Le jeu est développé et édité par l'armée américaine pour attirer et recruter de nouveaux soldats. L'armée américaine avait une certaine exigence en ce qui concerne le niveau de détail dans la simulation : "*Que le jeu soit joué de manière fidèle, comme une représentation correct des services notamment en matière d'éthique, les codes de conduite, et s'étendant jusqu'à la représentation exacte de la hiérarchie, des missions, armes et équipements, uni-*

formes, les tactique d'intervention, etc.". AA est un jeu de tir à la première personne (First Person Shooter, FPS). La première version du jeu intitulée : Operations RECON (version 1.0), a été publiée en Juillet 2002. Depuis lors, l'armée américaine fournit des mises à jour régulières avec de nouveaux modules de formation, de nouvelles cartes, des corrections ainsi que des améliorations. En Décembre 2003, une version 2.0a a été publiée sous le nom de : "*forces spéciales*". Cette version a plus de trois millions de comptes enregistrés. Les données (connaissances) relatives au jeu ont été recueillies à travers plus de 400 heures de jeu.

II.2.3 Synthèse et conclusion

La plupart des jeux sérieux que nous avons étudié utilisent une architecture propre et se revendiquent être des jeux sérieux. Les jeux étudiés sont généralement mono-apprenant et les techniques d'évaluation qu'ils utilisent sont simples et individuelles.

Le point le plus important dans un jeu sérieux est l'aspect pédagogique que fournit ce dernier pour l'apprentissage. Dans ce cadre, la notion de jeu sérieux adaptatif est souvent utilisée. L'adaptation peut être définie comme une caractéristique exprimée au niveau du jeu sérieux, qui reflète sa capacité à se modifier structurellement en réaction à certains évènements bien identifiés. [Hocine et al., 2011a] fait un tour d'horizon des différentes approches de l'adaptation dans les jeux sérieux. Ces différentes approches ne prennent pas en compte le domaine étudié et ont une représentation simple de l'apprenant.

A l'opposé, les travaux issus du domaine de l'EIAH (Environnements informatiques pour l'apprentissage humain) ciblent spécifiquement des objectifs pédagogiques (apprentissage, évaluation...). Les systèmes tutoriels intelligents (STI) forment un courant particulier des EIAH qui a pour principe l'individualisation dans la formation. Un STI propose de représenter (explicitement ou non) les connaissances du domaine étudié avec le modèle du domaine [Anderson, 1988], ainsi que les connaissances des actions de l'apprenant tout au long du processus d'apprentissage avec le modèle de l'apprenant [VanLehn, 1988], Vanlehn parle de l'état mental d'un apprenant pendant la formation.

II.3 Les systèmes tutoriels intelligents

Dans cette section nous présentons plus en détail le concept des systèmes tutoriels intelligents (STI) ainsi que quelques travaux dans ce domaine.

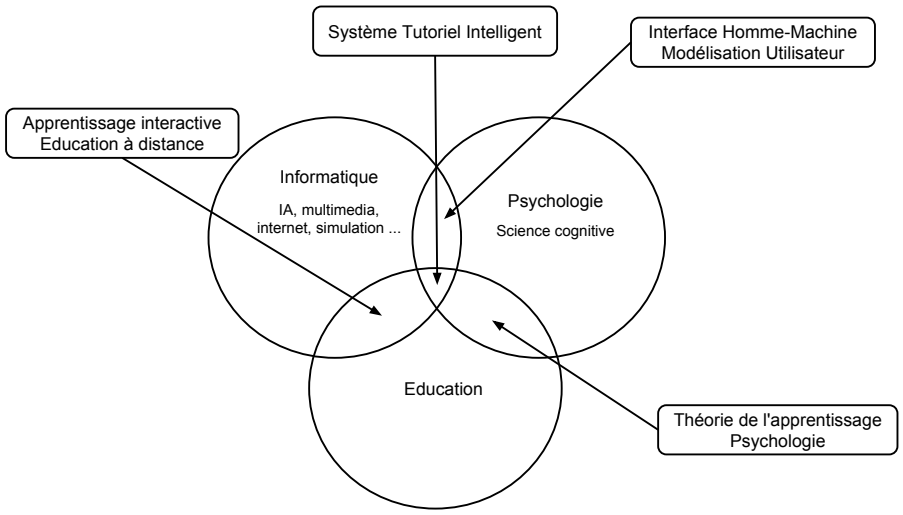


Figure II.13 – L'interaction des disciplines qui constituent un STI [Burns et Capps, 1989]

II.3.1 Définition et architecture

Les systèmes tutoriels intelligents sont nés de l'interaction de trois disciplines : l'informatique, la psychologie et l'éducation. Le domaine informatique fournit l'aspect multimédia qui se traduit par les différents outils d'immersion comme l'interface homme-machine, la 3D (monde virtuel), la simulation, etc. Le domaine de la psychologie apporte tous ce qui se rapporte aux sciences cognitives et à la compréhension des apprenants. Le domaine de l'éducation apporte la connaissance et les compétences à apprendre. Cette interaction est illustrée dans la figure II.13.

Burns [Burns et Capps, 1989] caractérise l'architecture d'un ITS comme composée de :(i) un module de l'expert qui contient les connaissances du domaine,(ii) un module de l'apprenant qui contient ce que l'apprenant sait sur le domaine, (iii) un module de l'instructeur qui a pour objectif d'identifier les lacunes de l'apprenant pour que le système adapte ses stratégies pour combler ces lacunes (ce module est aussi appelé module pédagogique). À ces modules se rajoutent un environnement pédagogique et une interface homme-machine, canal de communication du système tutoriel avec l'apprenant. La figure II.14 décrit l'architecture générale d'un STI.

Le module de l'expert

John Anderson [Anderson, 1988] a identifié le point le plus important dans la conception du module de l'expert dans la façon de modéliser (codifier) les connaissances. Le module de l'expert doit avoir des connaissances spécifiques et détaillées provenant de personnes qui ont des années d'expérience dans un domaine particulier. Il y a plusieurs manières de représenter les connaissances dans ce module. La première est

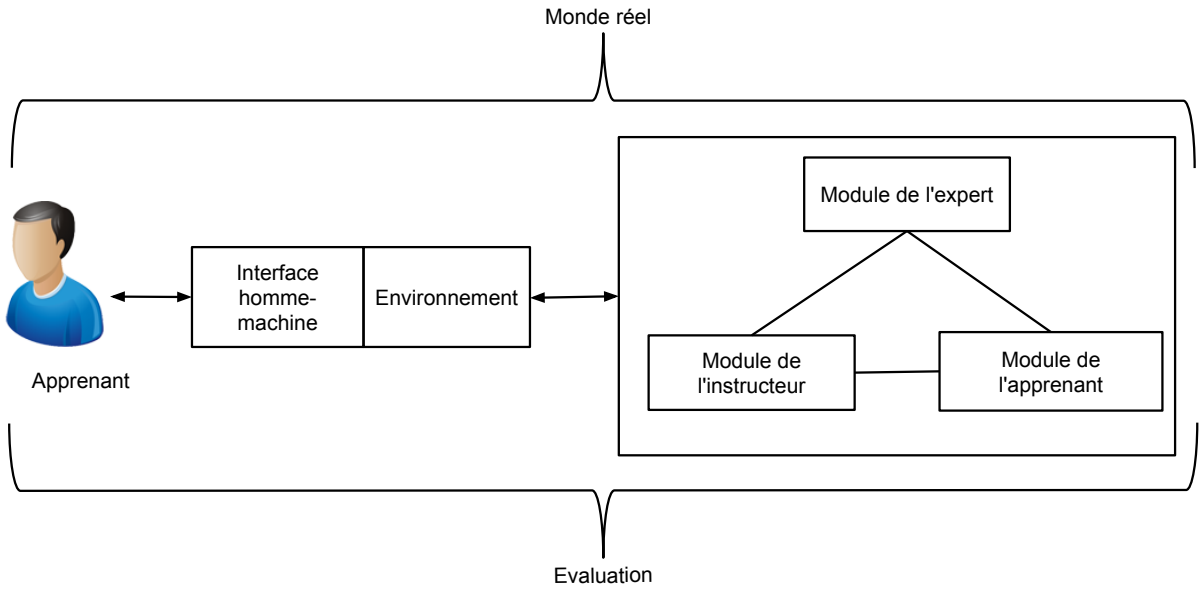


Figure II.14 – Architecture générale d'un STI [Burns et Capps, 1989]

de la représentée de façon simplifiée sans relayer l'intelligence humaine. La littérature fait souvent référence à des "boîtes noires". Ce type de représentation n'est pas très efficace dans le domaine des ITS mais peut suffire pour représenter des règles simples. La seconde façon est de la représenter de manière plus complexe avec une certaine sémantique par exemple avec les ontologies [Provitolo et al., 2009]. Les ontologies permettent de définir des concepts et des relations pour modéliser le modèle du domaine. On peut aussi avoir une représentation des connaissances sous forme de méta-modèle comme dans [Buche, 2005].

Le module de l'apprenant

Un ITS doit construire un modèle de compréhension de l'apprenant pour ensuite utiliser cette compréhension pour adapter les instructions aux besoins particuliers de l'apprenant. La structure de connaissances qui décrit l'état de l'apprenant est le "*modèle de l'apprenant*", et le processus de raisonnement en charge de son développement est appelé le "*diagnostique de l'apprenant*" [VanLehn, 1988]. Le module de l'apprenant utilise généralement le même type de connaissance que la représentation utilisée dans le module de l'expert. Cette réutilisation permet une superposition des connaissances, où les connaissances de l'apprenant sont représentées comme un sous-ensemble des connaissances de l'expert. Ainsi, les concepts manquants sont représentés, mais pas les erreurs. Les connaissances erronées doivent aussi être modélisées dans le module de l'apprenant. Afin de réaliser cette modélisation, le modèle de superposition est enrichi par une bibliothèque d'erreurs. Les erreurs peuvent être recueillies de manière empirique, ou bien générées automatiquement à partir de la procédure cible.

Le module de l'instructeur

Un ITS doit avoir trois caractéristiques de tutorat [Halff, 1988] : (a) le contrôle sur la représentation des connaissances de l'enseignement pour la sélection et l'ordonnancement des exercices ; (b) la capacité pour répondre aux questions des apprenants sur les objectifs pédagogiques et le contenu, et (c) des stratégies pour déterminer quand les apprenants ont besoin d'aide pour leur proposer des solutions. Le but de ce module est de circonscrire la nature de l'enseignement et de mettre en œuvre l'enseignement comme une solution au problème de la communication éducative (interaction entre l'instructeur et l'apprenant).

L'environnement pédagogique

Un environnement pédagogique est un composant important dans un ITS. Il représente les outils fournis par le système pour faciliter l'apprentissage. Parmi les problématiques se rapportant à l'environnement pédagogique, on trouve [Burton et Brown, 1979] : (a) le niveau d'abstraction, (b) la fidélité, (c) les séquences, et (d) l'aide de routine. L'abstraction signifie quelles sont les caractéristiques du monde réel qui seront représentées dans la conception de l'environnement. La fidélité se réfère à quel point la simulation de l'environnement correspond au monde réel. Il faut prendre en considération différents types de fidélité, par exemple, la fidélité physique, la fidélité d'affichage, une fidélité mécanique, etc. Les séquences représentent la structure mise en place par le concepteur pour l'apprentissage des compétences complexes. Un apprenant progresse à travers une séquence de micro jeux de plus en plus complexes, chacune offrant de nouveaux défis et de nouveaux objectifs à atteindre. Par le biais de l'aide de routine, le concepteur prend en compte des informations supplémentaires sur l'apprenant, ces informations peuvent être nécessaires pour le bon fonctionnement de l'ITS. Il y a différents degrés d'aide, par exemple, dire à l'apprenant quoi faire, ou bien un assistant fait la tâche à sa place.

L'interface homme machine

Le problème qui se pose lors de la conception de l'interface homme-machine est que l'apprenant doit utiliser la technologie afin d'apprendre, et ce n'est très probablement pas un utilisateur expert. Si l'interaction homme-machine est mal conçue, une session de formation sera probablement inefficace. Autrement dit, si l'apprenant est obligé de dépenser une énergie intellectuelle significative pour interagir avec l'ordinateur, l'apprenant aura moins d'énergie intellectuelle et émotionnelle pour l'apprentissage de la ou les compétences à apprendre [Miller, 1988]. On parle alors de surcharge cognitive.

II.3.2 Quelques travaux dans le domaine des STIs

Il existe beaucoup de travaux basés sur les STIs, ces travaux sont plus au moins originaux. Certains travaux ajoutent des modules additionnels au concept de base des STIs, comme dans [Buche, 2005], qui ajoute un module d'erreur ou bien dans [Amokrane, 2010], qui ajoute un module du domaine étudié (en l'occurrence, un module de risque).

II.3.2.1 Cadre d'analyse

Le cadre d'analyse est le même que nous avons utilisé pour l'étude des jeux sérieux et consiste à étudier différents STIs d'un point de vue architecture, représentation des connaissances et de l'évaluation des apprenants.

II.3.2.2 Help Agent for Learning (HAL)

Dans [Lourdeaux, 2001], Lourdeaux propose le concept de HAL (Help Agent for Learning), dans un STI pour la formation des conducteurs de TGV. HAL s'articule autour de la connaissance de l'apprenant (modèle de l'apprenant), de l'activité à apprendre (modèle de référence), des connaissances et des stratégies d'enseignement (modèle pédagogique) ainsi que la manière de les appliquer (module de diagnostic).

Architecture L'architecture de HAL reprend les modules de base d'un STI et est constituée des modèles (modules) suivants :

Modèle de l'apprenant : Le modèle de l'activité de l'apprenant dans HAL est basé sur la structure proposée par Self [Self, 1988]. Selon Self un modèle de l'apprenant doit répondre à quatre questions :

- Quel type d'apprenant est-il, i.e. quelles sont ses caractéristiques individuelles ?
- Qu'a-t-il fait, i.e. quel est l'historique de ses actions ?
- Que connaît-il, i.e. quelle est sa connaissance conceptuelle ?
- Que peut faire l'apprenant, i.e. par rapport à ses connaissances procédurales ?

Le modèle de l'apprenant est caractérisé par : le niveau de l'apprenant (débutant, intermédiaire, confirmé), l'expérience de l'apprenant (Nombre de fois où l'apprenant met en pratique cette procédure), le profil de l'apprenant (fonction), et les facultés de l'apprenant (Émotif et mémorisation visuelle ou sonore).

Modèle de reference : Ce modèle représente le domaine étudié et il est représenté par un ensemble de tâches (procédures) hiérarchiques organisées en plans. Une caracté-

ristique de base des plans de tâches est la linéarité. Un plan est linéaire si les tâches sont ordonnées dans le temps, i.e. les relations entre les tâches répondent à des contraintes d'ordre. Un critère de préférence relative (Cr) indique le meilleur raisonnement. Un plan est non-linéaire si l'ordre de certaines tâches n'est pas imposé. L'ordonnancement des tâches est justifié par des relations de causalité. La plupart des tâches sont linéaires et certaines ont des liens de causalité.

Modèle pédagogique : Pour structurer l'expertise pédagogique, le modèle pédagogique de HAL est basé sur deux types de paramètres : les paramètres correspondant au type d'intervention, et les paramètres d'intervention pédagogique.

Le type d'intervention correspond aux paramètres d'initialisation d'une session d'exercice. Ces paramètres peuvent être spécifiées, par le formateur, à l'initialisation du système ou pendant que l'apprenant exécute le scénario. Ces paramètres sont :

- Le mode de contrôle : Libre, semi-automatique.
- La méthode pédagogique : Méthode explicative, méthode active.
- La stratégie générale : Encourager et/ou perturber l'apprenant.
- Les niveaux cognitifs : Seuils de déclenchement pour les interventions de HAL.

Les paramètres d'intervention correspondent aux choix pédagogiques de HAL au cours de la session (en ligne), i.e. les solutions pédagogiques à apporter pour aider l'apprenant. Ils peuvent être spécifiés, par le formateur et/ou le programmeur, lors de la préparation "logicielle" du scénario. Ces paramètres sont implémentés pour chaque tâche, sous-tâche et chaque comportement prévu de l'apprenant (adéquat ou non). Il s'agit des stratégies pédagogiques et des assistances pédagogiques. Les stratégies pédagogiques peuvent être classifiées selon deux niveaux. Le premier niveau correspond au niveau stratégique. Il consiste à organiser les différentes sessions à long terme. Le second niveau correspond au niveau tactique. Il consiste à organiser les connaissances de l'apprenant au cours d'une session, i.e. au moment même de la résolution des problèmes. HAL propose alors des scénarios de base et des interventions pédagogiques en fonction des caractéristiques individuelles de l'apprenant. HAL se focalise ainsi essentiellement sur l'apport de stratégies pédagogiques tactiques.

Les stratégies pédagogiques tactiques, proposées par HAL peuvent être réparties en deux groupes. Les premières consistent à modifier le scénario et les secondes à guider l'apprenant. Les stratégies pédagogiques permettant de guider l'apprenant sont réparties selon deux méthodes pédagogiques : méthode active et la méthode explicative. Les différentes stratégies implémentées dans HAL sont résumées dans le tableau II.2.

Ces stratégies peuvent être représentées par différents types d'assistance pédagogiques et différents niveaux de réalisme (enrichissement, dégradation, rehaussement, etc).

Modification du scénario	Guidage de l'apprenant	
	Méthode active	Méthode explicative
Introduire des problème	Laisser faire l'apprenant	Montrer ou suggérer la connaissance
Ne pas générer de problèmes	Faire à la place de l'apprenant	Montrer l'écart entre ce que les apprenants doivent faire et ce qu'ils ont fait
Inhiber les problèmes	Dire ce qu'il faut faire	Expliquer
	Faire refaire aux apprenants	Monter la(les) conséquence(s) des erreurs

Tableau II.2 – Stratégies pédagogiques tactiques implémentées dans HAL.

Représentation des connaissances : L'historique des actions de l'apprenant est construit dynamiquement pendant la session. Un marquage des actions et des connaissances utilisées par l'apprenant, à partir du modèle de référence, permet à HAL de retranscrire la session et d'interpréter les événements jugés significatifs. Dans HAL Les connaissances conceptuelles sont représentées par le niveau K décrit par [Rasmussen, 1983] et les connaissances procédurales par le niveau R décrit aussi dans [Rasmussen, 1983]. La représentation des connaissances de l'apprenant correspond à l'écart entre les connaissances supposées de l'apprenant et les connaissances à acquérir. Deux types de représentation des connaissances de l'apprenant sont principalement utilisés dans les STI : le modèle d'expertise partielle et les modèles différentiels.

Pour le modèle de référence, HAL utilise une représentation de plans de tâches hiérarchiques. Chaque plan décrit un ensemble de buts à atteindre et un ensemble de tâches à réaliser pour atteindre ces buts. Les tâches peuvent être des tâches complexes ou des tâches élémentaires. Une tâche complexe est décrite par un sous-plan. Une tâche élémentaire décrit une action, comme prendre un objet. Les tâches de haut niveau sont décomposées en sous-tâches et récursivement jusqu'aux sous-tâches élémentaires. Les plans hiérarchiques sont faciles à gérer et facilement utilisables. Ils permettent aux formateurs et aux pédagogues de conceptualiser une tâche en un ensemble d'étapes et de sous-étapes hiérarchiques.

Évaluation des apprenants : Le modèle d'activité de l'apprenant de HAL utilise la méthode de type différentiel pour représenter les connaissances (conceptuelles et procédurales), un catalogue d'erreurs partiel a été ajouté pour compléter la base de connaissances. En effet, un modèle d'expertise partielle permet de conserver un modèle générique global avec un niveau d'abstraction élevé. Un catalogue d'erreurs génériques (e.g. saut d'étape, problèmes d'estimation des distances) permet de conserver cette généri-

cité. Par contre, dans HAL, il s'agit de réaliser des formations professionnelles sur des procédures très réglementées où les erreurs influent sur la sécurité des personnels et des voyageurs, ainsi, ces erreurs sont et doivent être maîtrisées.

Un critère de performance (Cp) permet de plus d'évaluer le degré de maîtrise de l'apprenant. Il existe un critère de performance pour chaque comportement attendu ou problématique de chaque tâche. Ce critère est mis à jour dynamiquement pendant la session.

Self [Self, 1988] a spécifié six fonctions principales permettant de caractériser les STI [Bruillard, 1997]. Ces fonctions dépendent des caractéristiques du modèle de l'apprenant et du modèle pédagogique. Ces fonctions sont les suivantes :

- Corrective : Corriger l'erreur (aider l'apprenant à éliminer les connaissances erronées).
- Élaboration : Compléter les connaissances manquantes.
- Stratégique : Contrôler la stratégie globale d'interaction.
- Diagnostic : Vérifier le modèle de l'apprenant en choisissant une action pour lever une ambiguïté.
- Prédictive : Simuler l'apprenant pour prévoir ses actions (limite l'espace de recherche).
- Évaluation : Évaluer les performances de l'apprenant.

Dans HAL, les connaissances sont représentées sous forme de plans qui permettent de contrôler le déroulement de la session, i.e. d'effectuer les choix des plans pédagogiques et leur éventuel changement en cours de route.

Conclusion : Les travaux de HAL se centrent prioritairement sur l'aspect pédagogique et le support à l'apprenant, l'évaluation se fait par le biais des stratégies d'interventions (quand et pourquoi intervenir). Néanmoins cette intervention est définie à l'avance par l'expert du domaine (concepteur du scénario) en se basant sur les tâches (procédures) à évaluer (évaluation statique).

II.3.2.3 PEGASE (PEdagogical Generic and Adaptive SystEm)

Buche [Buche, 2005] propose PEGASE, un STI qui s'articule autour des modules classiques d'un STI en y ajoutant un module d'erreur. Ce STI a été utilisé dans le projet GASPAR (Gestion de l'Activité aviation et des Sinistres sur Porte-avion par la Réalité Virtuelle) qui simule l'activité aviation sur un porte-avions. Cette application permet d'aider à la formation de la gestion de l'aviation et permet de visualiser les activités de décollages (catapultages) d'atterrissages (appontage) ainsi que l'observation et la création d'intervention sur les avions.

Architecture : L'architecture de PEGASE est de type multi-agents et ce compose des modules/modèles suivants :

- Le modèle du domaine : ExpertAgent
- Le modèle des erreurs : ErrorAgent
- Le modèle de l'apprenant : LearnerAgent
- Le modèle de l'interface : InterfaceAgent
- Le modèle du formateur : TeacherAgent
- Le modèle pédagogique : PedagogicalAgent

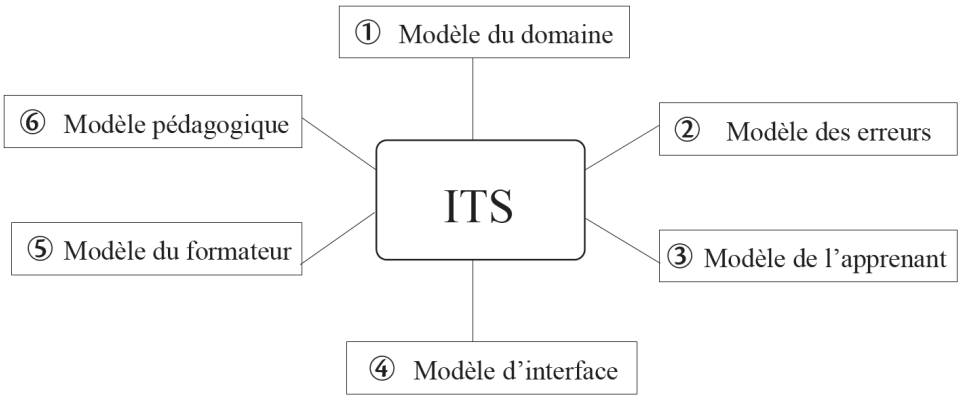


Figure II.15 – Les différents module de l'ITS proposé par Buche [Buche, 2005]

Chaque agent est en interaction avec un ou plusieurs agents, ce qui lui permet de mettre à jour ses connaissances et de prendre en compte l'analyse des autres agents. Chacun de ces agents se base sur une structure commune AgentPackage (l'entité la plus abstraite d'un agent).

Buche propose un processus pédagogique en cinq étapes :

Étape 1 Observer (InterfaceAgent) : Le modèle de l'interface analyse les activités de l'apprenant et identifie les éléments pertinents pour la formation ; ces éléments sont par la suite transmit au modèle de l'apprenant. Ces informations concernent : les actions de l'apprenant, les éléments observables par l'apprenant et les déplacements de l'apprenant.

Étape 2 Détecter et identifier une erreur (LearnerAgent, ExpertAgent et ErrorAgent) : Le système analyse les actions de l'apprenant et les compare aux actions à effectuer ; Cette comparaison permet de détecter d'éventuelles erreurs. Si une erreur est détectée un mécanisme d'identification d'erreur est mis en place.

Étape 3 Proposer une assistance pédagogique (PedagogicalAgent) : En utilisant le modèle de l'apprenant et le modèle du domaine, un mécanisme simulant un raisonnement pédagogique préconise une assistance pédagogique adapté à la situation. A noté que cette étape n'est pas conditionnelle et peut intervenir même si aucune erreur n'est détectée.

Étape 4 Choisir une assistance pédagogique (TeacherAgent) : Le formateur sélectionne une assistance pédagogique parmi celles préconisées.

Étape 5 Représenter l'assistance pédagogique (InterfaceAgent) : L'assistance pédagogique sélectionnée est représentée dans l'environnement virtuel.

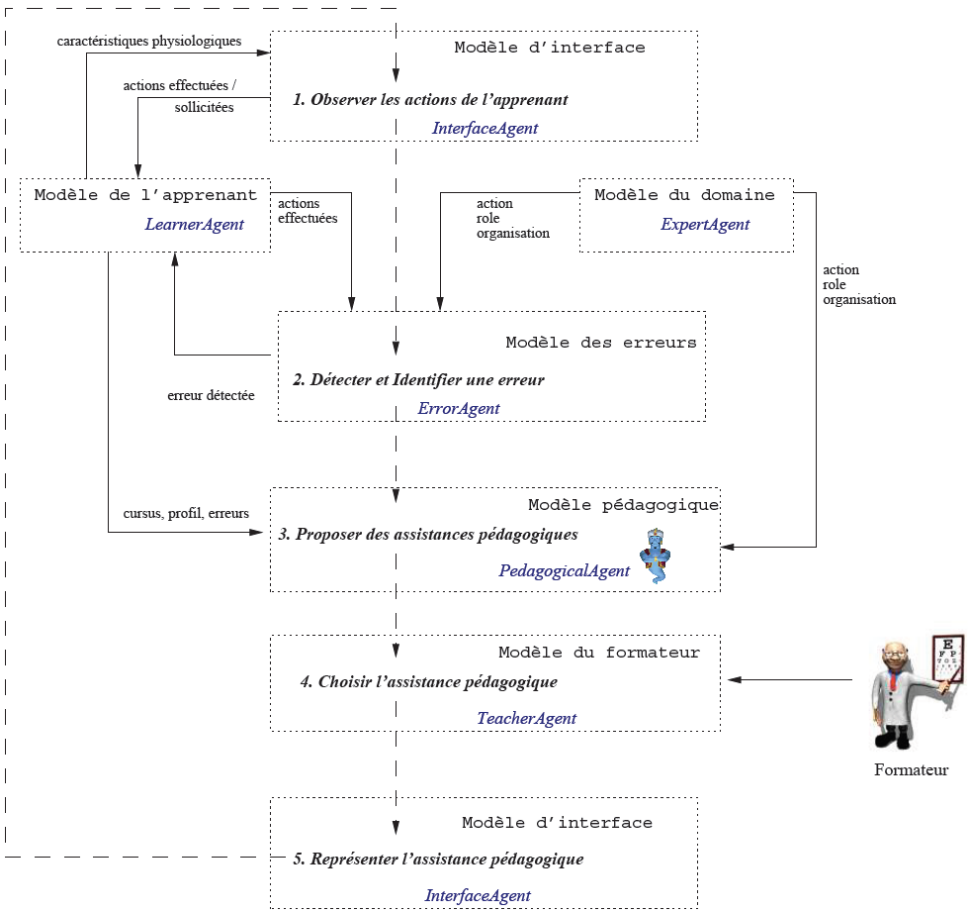


Figure II.16 – Processus pédagogique proposé par Buche [Buche, 2005]

Modèle du domaine : Le modèle du domaine représente la connaissance que l'apprenant doit acquérir. Dans le cadre d'étude de Buche, l'apprentissage porte sur la réalisation de procédures collaboratives ; Pour ce faire il a fallu spécifier les équipes, les procédures, les actions et les règles d'usage général.

Module des erreurs : Lorsque l'apprenant effectue des comportements qui sont en désaccord avec le travail à réaliser ou bien avec les règles d'usage général, le modèle des erreurs offre alors une caractérisation générique de ces erreurs permettant à l'agent pédagogique d'intervenir.

Modèle de l'apprenant : Le modèle de l'apprenant s'intéresse aux caractéristiques et aux activités de l'apprenant. Ce dernier doit représenter les informations caractérisant l'apprenant tant du point de vue de son profil que du point de vue de sa progression dans son apprentissage. Il correspond alors à la représentation de l'apprenant dans sa situation d'apprentissage.

Modèle d'interface : Le modèle d'interface représente et analyse les interactions entre l'apprenant et le système. Ce dernier se place comme médiateur entre le système et l'apprenant dans la situation d'apprentissage.

Modèle du formateur (une sorte de poste superviseur) Le modèle du formateur est la représentation du formateur sans la situation simulée. Il fixe les règles du jeu, quel est la ou les procédures à effectuer, et le ou les rôles à jouer par l'apprenant. Le modèle de l'apprenant offre également la possibilité d'intervenir par des moyens d'interaction privilégiés dans l'environnement au travers de l'agent formateur. Aussi il offre une vue d'ensemble sur la procédure à réaliser permettant le suivi de l'exercice courant. Enfin il récolte les propositions d'assistance de l'agent pédagogique.

Modèle pédagogique : Le modèle pédagogique est la représentation du pédagogue dans la situation d'apprentissage. Il simule un raisonnement pédagogique et influe par conséquence sur la pédagogie et l'apprentissage.

Représentation des connaissances : Pour chaque agent, Buche a définit une base de connaissance spécifique, ces connaissances sont cristallisées dans les différents métamodèle UML suivants :

Modèle du domaine : Buche définit deux types de connaissance à modéliser, il s'agit des éléments liés au travail à réaliser (procédural en équipe) et ceux liés aux règles d'usage général :

- Travail procédural en équipe : dans ce cas d'apprentissage, il faut représenter les connaissances qui portent sur les équipes, les procédures et les actions.
- Les règles d'usage général : ces règles sont indépendantes du travail procédural en équipe et ne sont pas soumises à l'ordonnancement de la procédure.

Modèle de l'apprenant : le modèle de l'apprenant représente les connaissances caractérisant l'apprenant. Buche propose une base de connaissance en deux parties, une partie épistémique et une partie non épistémique :

- Partie épistémique : Cette partie fournit les informations sur l'état des connaissances de l'apprenant pour la notion présente dans le domaine. Elle contient une représentation des actions et des erreurs effectuées par l'apprenant.
- Partie non épistémique : représente les trois concepts de l'ingénierie cognitive : un profil psychologique, des caractéristiques psychologiques, une image mémorielle (Reflète des éléments potentiellement observables).

Modèle d'erreur : le premier objectif du modèle d'erreurs est de caractériser les erreurs de manière générique, pour cela Buche distingue quatre types d'erreurs :

- Les règles d'usage (RuleError) : cette erreur précise les termes impliqués dans la mis en échec de la règle.
- Sur les actions (ActionError : pré-condition) : Elle détermine la/les pré-condition(s) d'une action qui sont en échec.
- Sur les procédures (ProceduralError : ordonnancement) : cette erreur intervient lorsque l'action sollicitée ne fait pas partie des actions à réaliser.
- Sur les rôles dans les procédures (TeamProceduralError) : il s'agit d'une spécialisation de ProceduralError, cette dernière intervient lorsque l'action sollicitée ne fait pas partie des actions à réaliser, et en plus, elle n'est pas de la responsabilité de l'apprenant.

Modèle pédagogique : L'agent pédagogique doit avoir une représentation des connaissances qui caractérise une situation pédagogique. Cette situation est composée d'un ensemble d'éléments qui déterminent les circonstances dans lesquelles l'apprenant évolue à un instant donnée. Ainsi, la base de connaissance comporte deux parties :

- La situation pédagogique : offre la possibilité de caractériser des contextes en filtrant l'ensemble des connaissances disponibles pour ne garder que les informations clefs.
- Les règles pédagogiques : les décisions pédagogiques sont prises en considérant des règles pédagogiques (formalisme à base de règles). Ces règles portent sur les éléments de la situation pédagogique.

Modèle du formateur : Avant de démarrer l'exercice, le formateur doit déterminer les consignes, les règles à respecter. La base de connaissances de l'agent formateur précise alors les informations suivantes : l'équipe dans laquelle les apprenants évoluent, la procédure à réaliser et enfin le(s) rôle(a) à jouer.

Modèle de l'interface : Le modèle d'interface contient une base de connaissance sur l'apprenant (une sous partie de la base de connaissance du modèle l'apprenant).

Cette connaissance va servir notamment à déterminer les éléments qu'il peut potentiellement observer dans l'environnement. De la même manière, le niveau de l'apprenant peut être utilisé pour gérer ses possibilités d'action.

Évaluation des apprenants : L'évaluation des apprenants se fait en temps réel via le le modèle d'erreurs. Les erreurs définies dans le modèle d'erreurs ne prennent pas en compte l'aspect collectif de l'apprentissage et ne permettent pas de donner une évaluation globale du groupe.

Conclusion : PEGASE propose un environnement d'apprentissage de tâches collaboratives dans un environnent virtuel. L'évaluation ce fait via le module d'erreur qui permet de comparer les actions de l'apprenant aux erreurs éventuelles décrites dans le module d'erreur. Néanmoins cette évaluation se fait de manière individuelle et ne permet pas une évaluation globale du groupe. Enfin, le module pédagogue, qui analyse d'éventuelles erreurs et propose des stratégies au formateur (nécessité d'avoir un formateur).

II.3.2.4 Helpful agent for safEty leaRning in virtuAl environment (HERA)

HERA [Amokrane, 2010] se base principalement sur deux modèles d'un STI classique : le modèle de domaine et le modèle pédagogique. Dans le but de gérer les erreurs de l'apprenant, HERA intègre un modèle d'erreurs comme dans Buche [Buche, 2005]. Pour permettre le rejeu, HERA utilise un modèle appelé "*trace de l'apprenant*". HERA est dédié à la formation à la gestion des risques, c'est pour cela qu'un cinquième modèle appelé "*modèle de risques*" a été ajouté.

Architecture : Contrairement à un STI classique où les notions de modèle et module sont confondues, dans HERA la différence est faite entre les données exploitées ou produites par le système (modèles), et les processus qui prennent en charge l'analyse, l'interprétation, la transmission et l'enregistrement de ces données.

Modèle du domaine : Les tâches de références sont modélisées par le langage HAWAI-DL. HAWAI-DL est un langage de description de la tâche qu'un apprenant doit faire. Il permet une description détaillée d'une tâche grâce à un ensemble d'attributs qui lui sont associés selon les besoins. Le modèle HAWAI-DL considère le monde sous trois points de vue différents (agent, travail, situation de travail) qui sont décrits par cinq concepts principaux : objet, rôle, agent, tâche, événement. Ces concepts et les relations entre eux constituent l'ontologie de HAWAI-DL, comme le montre la figure II.17.

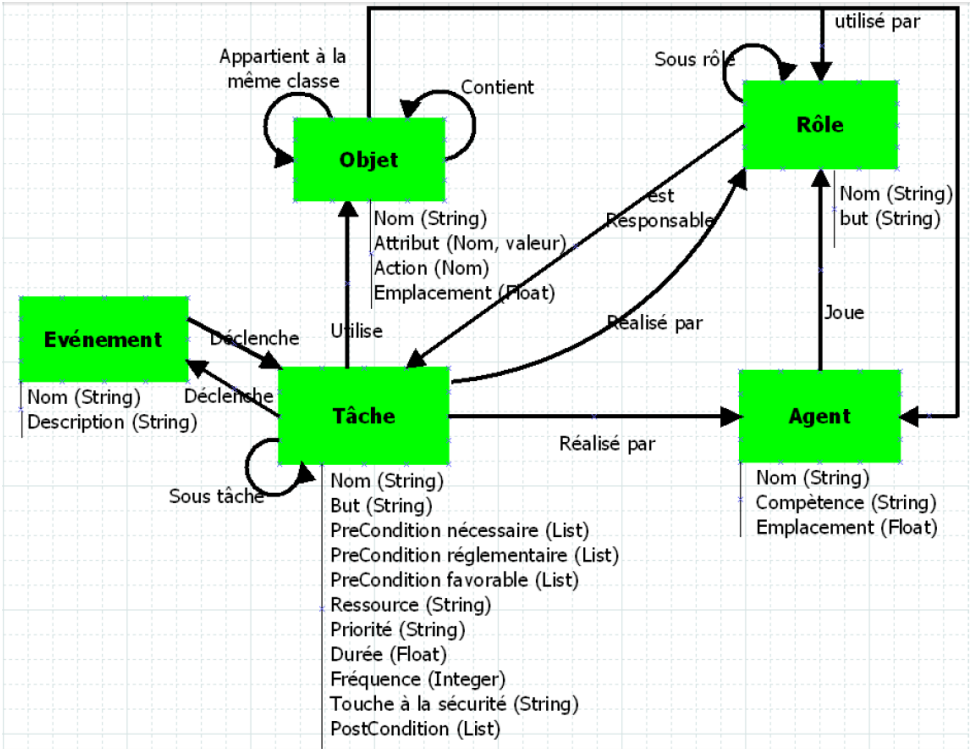


Figure II.17 – HAWAI-DL : Concepts et relations [Amokrane, 2010]

Modèle d'erreurs : Ce modèle contient une classification des erreurs qu'un apprenant peut commettre durant sa formation. Cette classification ne dépend pas d'un scénario donné, c'est une classification générique qui prend en compte tous les cas d'erreur dans les différents scénarios :

- Les erreurs classifiées par Hollnagel [Hollnagel, 1993] : elles concernent quelques phénotypes tels que : l'omission, la répétition, l'inversion, l'erreur d'objet cible, l'erreur d'action, etc.
- Les erreurs de condition liées au non respect de post-conditions ou de pré-conditions.
- Les erreurs de rôle : lorsqu'un apprenant réalise une tâche qui n'est pas dans son rôle.
- Les erreurs du point de vue subjectif : sont liées à l'aspect omniscient de l'apprenant, plus particulièrement sur son champ visuel. Par exemple, l'apprenant ne réagit pas à un phénomène (déclenchement d'un incendie) parce qu'il ne l'a pas vu.
- Les erreurs CLTU : lorsqu'un apprenant viole l'activité prescrite pour atteindre certains compromis : favoriser la production et le temps aux dépends de la sécurité (notre cas d'étude) ; favoriser la sécurité et le temps aux dépends de la quantité de production, etc. Toutes ces déviations sont tolérées par l'usage à un expert qui sait comment réagir par la suite en cas de risque ; mais pas à un apprenant.

Modèle pédagogique : HERA est doté d'un modèle pédagogique pour intervenir et pour assister l'apprenant, l'aider, lui montrer les conséquences de ses erreurs, etc. Le modèle pédagogique contient principalement un ensemble de règles qui se basent sur d'autres éléments du modèle tels que : le niveau de l'apprenant, le but pédagogique, et les situations d'apprentissage. Le modèle pédagogique est construit en utilisant le langage UML. Par ailleurs, le système raisonne en se basant sur des règles pédagogiques simples (Si ... Alors) afin de déterminer les rétroactions pertinentes.

Modèle de risque : HERA est un système de formation dédié aux sites classés à hauts risques (sites SEVESO), ainsi, une certaine analyse des risques a été nécessaire. Le modèle du risque contient principalement l'interprétation des études réalisées par l'INERIS (Institut National de l'Environnement Industriel et des Risques). Le modèle de risque est constitué de quatre concepts principaux :

- Le risque : représente tous les risques qu'un apprenant peut rencontrer durant sa formation ou durant son travail sur site réel.
- L'erreur commise par l'apprenant : Ce concept correspond à toutes les erreurs qui représentent les causes principales des risques produits dans l'environnement virtuel.
- La tâche en erreur : Pour permettre à l'apprenant de comprendre l'origine de son erreur ainsi que ses conséquences, le système montre la tâche en erreur ainsi que des explications sur les causes de son erreur.
- La condition d'environnement : ce concept représente des erreurs humaines qui n'engendrent pas de risque sans la présence d'une condition supplémentaire.
- La barrière de sécurité : c'est une tâche particulière qui permet de prévenir, d'éviter, ou de gérer un risque.

Modèle des traces de l'apprenant : La trace de l'apprenant est utilisée pour garder la trace de l'activité de l'apprenant résultant de l'analyse de ses actions observables et des effets dans l'environnement virtuel. Elle contient principalement les tâches réalisées par l'apprenant, ses erreurs commises, les causes et les conséquences de ces erreurs, ainsi que des explications. Toutes ces informations sont enregistrées (dans un fichier XML) pour permettre le retour sur ce qui a été fait en mode "*rejeu*". Ce modèle contient aussi une liste des objets observables par l'apprenant.

Module d l'interface : C'est un module intermédiaire entre les autres modules de HERA et n'importe quel module. Il a pour entrées tout ce qui se passe dans l'environnement virtuel, reçu sous forme de messages. Le module d'interface analyse ces messages et les adapte en un format compréhensible par le module de reconnaissance.

Module de reconnaissance : Ce module se base sur les techniques de "la reconnaissance de plan" afin de déterminer la tâche réalisée par l'apprenant, les erreurs commises et les risques engendrés par ces erreurs. HERA est basé sur "keyhole recognition" de El-Kechai [El-Kechaï et Després, 2006], qui est un système générique, et qui a été amélioré et adapté aux besoins de la gestion des risques. Dans ce module, chaque tâche est considérée comme une entité autonome. Le changement d'état de chaque tâche (en cours, faite, etc.) est évalué par la tâche elle-même à chaque événement dans l'environnement virtuel. L'état d'une tâche et de ses sous-tâches évolue au cours de la session selon les actions des opérateurs.

Module de l'apprenant : Ce module permet de déterminer la tâche réalisée par l'apprenant, parmi toutes les tâches considérées comme candidates fournies par le module de reconnaissance. Ce module est complémentaire au module de reconnaissance. HERA est basé sur une approche formelle (des heuristiques) proposée par El-Kechaï [El-Kechaï, 2007] pour résoudre le conflit entre les tâches candidates (identifier la tâche effective) grâce à une ou plusieurs heuristiques.

Module du risque : Ce module permet de prendre la décision du déclenchement ou de l'arrêt d'un risque. D'un côté, il reçoit le message d'erreur transmis par le module de l'apprenant, plus particulièrement les erreurs CLTU et les erreurs de point de vue subjectif. En consultant le modèle de risques, il vérifie si l'erreur commise engendre un risque et si ce dernier dépend d'une autre condition supplémentaire pour se déclencher. Dans ce cas, il vérifie si cette condition est satisfaite pour le déclencher. Ensuite, la probabilité du risque est calculée en utilisant un réseau Bayésien défini dans le modèle de risques. Enfin, le risque produit et sa probabilité sont envoyés au module pédagogique.

Module pédagogique : Ce module permet de déterminer en temps réel la rétroaction nécessaire (l'assistance ou le changement dans l'environnement virtuel). A la réception d'un message d'erreur du module de l'apprenant, et d'un message du risque engendré et sa gravité du module de risques, le module pédagogique décide s'il doit intervenir ou non et comment intervenir. Il consulte les règles décrites dans le modèle pédagogique, et selon le type d'erreur, la probabilité du risque, le niveau de l'apprenant et le but pédagogique, il détermine les rétroactions pertinentes.

Module de rejeu : Ce module vise à reproduire le parcours de l'apprenant durant une session de formation. Il prend en entrée la trace de l'apprenant et produit les messages d'action qui correspondent aux tâches réalisées par l'apprenant (correctes ou erronées), ou les messages de risque qui correspondent aux risques engendrés par ces

tâches avec les explications associées. Ces messages sont envoyés au fur et à mesure à l'environnement virtuel pour permettre à l'apprenant et au formateur de re-visualiser l'activité réalisée.

Représentation des connaissances La représentation des connaissances est essentiellement caractérisée par le modèle du domaine basé par le langage HAWAI-DL [Edward et al., 2008], et le modèle de l'apprenant basé sur une approche formelle (des heuristiques) proposée par El-Kechaï [El-Kechaï, 2007].

Évaluation des apprenants L'évaluation des apprenants se fait de manière individuelle en temps réel via les modules d'erreurs et le module du risque. Puisque l'apprentissage est individuelle, l'évaluation collective est absente.

Conclusion La solution apportée par HERA est très intéressante notamment avec son module du risque qui est lié au domaine étudié. Malheureusement HERA ne propose pas d'apprentissage collectif (malgré la présence de plusieurs acteurs avec différents rôles) et l'évaluation se fait de manière essentiellement individuelle en se basant sur une base d'erreur définie dans le modèle d'erreur.

II.3.3 Synthèse et conclusion

Contrairement aux jeux sérieux, la plupart des STIs ont sensiblement la même architecture de base (modèle du domaine, modèle de l'apprenant, modèle pédagogique et l'interface homme-machine), plus quelques modèles spécifiques comme le modèle d'erreur ou le modèle de risque. Le reproche que l'on peut faire aux STIs est l'individualisation de l'apprentissage et le manque d'aspect ludique (que l'on trouve dans les jeux sérieux) dans l'apprentissage assisté par ordinateur. Qui dit apprentissage individuel dit évaluation individuelle, or il serait très intéressent de voir l'apport des STIs dans le domaine de l'apprentissage de tâches collaboratives (par exemple dans la gestion de crise).

II.4 Les SMA dans les jeux sérieux

L'intelligence artificielle (IA) a toujours été présente dans les jeux vidéo, plus ou moins élaborée selon l'objectif du jeu vidéo [Bakkes et al., 2009]. Les jeux sérieux empruntent beaucoup aux jeux vidéo classiques, mais ils sont plus que des jeux, ils sont conçus dans l'objectif de l'acquisition d'une ou plusieurs connaissances/compétences.

Les outils et mécanismes empruntés aux jeux vidéo doivent être plus élaborés, la part et la complexité des techniques d'IA doit être accrue. L'IA est utilisée pour simuler des phénomènes naturels ou des comportements humains pour obtenir un monde virtuel réaliste pour la formation [Zyda, 2005]. L'IA peut aussi être utilisée pour simuler les personnages non joueurs (PNJ), et cela se fait en général via les systèmes multi-agents (SMA). Un agent est défini comme étant un programme informatique situé dans un certain environnement, et capable d'effectuer des actions d'une manière autonome dans cet environnement pour atteindre son objectif [Wooldridge et al., 1995]. Dans d'autres cas les agents sont utilisés pour l'intégration et l'architecture logicielle, grâce à leurs propriétés sociales (interactions entre les agents) et de leur modularité (architecture, organisation, comportement, etc) [Lourdeaux, 2001] [Buche, 2005]. Dans cette section nous verrons quelques travaux dans le domaine des jeux sérieux et des STIs qui utilisent les SMA pour la simulation de comportements (PNJ) et leur architecture logicielle à des fins d'intégration.

II.4.1 Simulation

Les sections II.2 et II.3 montrent plusieurs travaux qui utilise les SMA pour simuler l'environnement comme le projet FORMAT-STOR (section II.2.2.3) qui utilise la modélisation agent IODA pour simuler différents éléments de l'environnent (porte, article, etc) ainsi que pour simuler des comportements humains (clients). La simulation peut être utilisée pour simuler des catastrophes naturelles [Komatitsch et al., 2003], la propagation du feu [Ohgai et al., 2007] ou pour simuler le comportement humain comme dans le projet RoboCup Rescue. Ces comportements humains peuvent être complexes, comme la prise de décision ou de résolution de problèmes [Kitano et al., 1999], ou simples, comme la simulation de foule [Helbing et al., 2002] (où les comportements réactifs sont utilisés pour l'évacuation d'urgence). Dans HAL (section II.3.2.2) le concept d'agent de l'environnement est introduit, cet agent permet de simuler tous les changements dans l'environnent résultant des actions de l'apprenant.

Dans [Querrec, 2003], Querrec propose Mascaret un système multi-agents pédagogique pour environnement virtuel. Le modèle Mascaret est proposé pour organiser les interactions entre les agents en les dotant de capacités réactives, cognitives et sociales pour simuler l'environnement physique et social :

- L'environnement physique représente d'une manière réaliste les phénomènes naturels.
- L'environnement social est simulé par des agents d'exécution des tâches collaboratives et adaptatives.

Le modèle MASCARET a été appliqué au projet SécuRéVi, un outil pour la formation

des pompiers. Le modèle MASCARET permet de concevoir une organisation complexe (en définissant le rôle, l'organisation, les comportements,...).

Dans le domaine de la modélisation multi-agents organisationnelle, le modèle MOISE+ (Modèle d'Organisation pour les systèmes multi-agents) [Hübner et al., 2002] considère la structure organisationnelle et la dynamique d'un SMA (par exemple à des fins de simulation). Ce modèle ajoute une relation déontique explicite pour concevoir des agents artificiels capable de raisonner sur l'accomplissement de leurs obligations, en s'inspirant du modèle BDI (Belief, Desire, Intention).

Le modèle BDI est un standard du domaine SMA pour la modélisation du comportement, inspiré du processus de raisonnement humain [Rao et al., 1995]. Le modèle BDI est basé sur les notions de capacités, compétences, croyances, désir, intention-plan. Ainsi, un agent BDI vise à atteindre ses objectifs en exécutant des plans en fonction de ses connaissances (croyances). Ces concepts permettent la conception et la programmation des agents avec des comportements complexes. Les agents peuvent sont aussi utilisés pour des fins de modélisation et d'intégration comme nous le présentons dans la section suivante.

II.4.2 Architecture et intégration

Comme nous l'avons vu dans la section II.3.2.2, Lourdeaux utilise une architecture multi-agents pour décrire l'Agent Pédagogique Intelligent (HAL) [Lourdeaux, 2001]. HAL possède deux types d'agents. Les premiers gèrent l'exécution du scénario par l'apprenant (S-Agents). Et les seconds gèrent la communication avec l'environnement virtuel (E-Agents). Dans [Buche, 2005] (section II.3.2.3), Buche utilise un SMA pour décrire les différents modules du STI (ExpertAgent, ErrorAgent, LearnerAgent, etc). Certains jeux sérieux utilisent aussi les SMA pour leur techniques d'adaptation comme dans [Hocine et al., 2011b] et [Westra et al., 2009].

II.5 L'évaluation des apprenants

L'évaluation des apprenants est une tâche essentielle pour le moniteur et l'apprenant. Il existe plusieurs processus d'évaluation, et chacun est caractérisé par un objectif et le moment où il intervient. Ces processus ont été classés par Dintilhac comme suit [Dintilhac et Ignace, 2005] :

– **Évaluation pronostique :** cette évaluation survient avant l'apprentissage et a pour but de positionner l'apprenant dans un cycle d'apprentissage (généralement utiliser pour définir l'année d'étude d'un élève).

- **Évaluation diagnostique :** cette évaluation survient juste avant l'apprentissage et permet d'évaluer le niveau de compétences de l'apprenant.
- **Évaluation formative :** cette évaluation survient lors de l'apprentissage et permet d'informer et de réguler l'activité de l'apprenant.
- **Évaluation formatrice :** cette évaluation survient lors de l'apprentissage et permet à l'apprenant de réguler l'activité de son apprentissage.
- **Évaluation sommative :** cette évaluation survient après l'apprentissage et permet de certifier ou pas le résultat de l'apprentissage.

Les travaux présentés dans cette section traitent de l'évaluation durant (formatrice et diagnostique) et après (sommative) l'apprentissage. L'évaluation durant l'apprentissage vise à apporter un support en temps réel à l'apprenant alors que l'évaluation après l'apprentissage nous fournit un diagnostic sur les compétences apprises et permet un débriefing après la formation entre l'apprenant et le moniteur. C'est typiquement les caractéristiques des besoins de l'évaluation dans les jeux sérieux.

Nous avons analysé différent travaux de la littérature sur les jeux sérieux [Murray, 1999] [Ma et al., 2011] et nous avons retenu trois critères importants pour l'évaluation des apprenants :

- La représentation des connaissances
- Les entrées et le résultat de l'apprentissage
- La stratégie de rétroaction

L'évaluation a pour objectif de certifier l'apprentissage de l'apprenant. Cet apprentissage a pour but d'acquérir une certaine compétence ou connaissance, pour cela nous devons représenter ces derniers de manière informatique. La représentation et la manipulation des connaissances représente en elle même un domaine vaste de recherche, avec plusieurs paradigmes de modélisation en fonction de la nature des connaissances ainsi que des objectifs de manipulation [Anderson, 1988]. Dans notre contexte de recherche, les travaux de Tchétgani nous semblent particulièrement intéressants. Dans [Tchétagni et Nkambou, 2002], Tchétgani et al. présentent une représentation hiérarchique des connaissances. L'évaluation hiérarchique consiste à évaluer l'état des connaissances de l'apprenant à plusieurs niveaux de granularité. Cette représentation permet un diagnostic précis et flexible des compétences de l'apprenant. Nous pouvons aussi avoir d'autres types de représentation comme les ontologies. Dans [Provitolo et al., 2009], Provitolo et al. présentent une ontologie pour représenter le domaine des risques et des catastrophes naturelles. Les ontologies apportent une sémantique avec la définition de concept et de relation entre les concepts. Nous pouvons aussi avoir une représentation des connaissances sous forme de méta-modèle comme dans [Buche et al., 2010].

Pour évaluer les apprenants, nous devons récupérer les actions réalisées par ces derniers, dans les jeux sérieux ceci est fait généralement via l'environnement virtuel.

Le modèle Entrée-Environnement-Résultat (Input-Environment-Outcome, I-E-O) a été proposé par Alexander W. Astin [Astin et al., 2012] comme cadre d'orientation pour les évaluations de l'enseignement supérieur. Le principe de ce modèle est que l'évaluation n'est pas complète si l'évaluation n'inclut pas des informations sur les entrées de l'apprenant, de l'environnement éducatif, ainsi que le résultat de l'apprentissage. Dans le contexte des jeux sérieux, les entrées de l'apprenant sont représentées pas les actions des joueurs dans l'environnement 3D ainsi que les décisions prises via les interfaces spécifiques qui constituent l'environnement d'apprentissage, tandis que les résultats peuvent être calculés par simulation.

Dans [Schmidt et Bjork, 1992], Schmid et al. montrent que le retour immédiat fait à l'apprenant permet à de ce dernier de finir rapidement son exercice. En contrepartie, en laissant l'apprenant réfléchir un peu en retardant l'intervention, les apprenants présentent une meilleure rétention des compétences apprises dans le temps. Le mode de rétroaction (immédiat ou bien retardé avec un certain délais) peut être adaptée à la progression de l'apprenant durant la formation, en fonction de sa vitesse de réaction ou du niveau de connaissance initial et actuel.

II.6 Étude comparative

Le tableau II.3 compare différents travaux dans le domaine des jeux sérieux (JS) et des systèmes tutoriels intelligents (STI). Ces travaux touchent des domaines d'applications larges et variés. Au-delà du cadre d'analyse définit dans la section II.2.2.1, nous avons pris cinq critères de comparaison que nous jugeons importants pour notre problématique de recherche :

- L'environnement : un environnement 3D (voire de Réalité Virtuelle ou RV) conforte l'aspect immersif de l'apprenant, mais cela dépend aussi (du domaine) de l'application.
- L'aspect multi-apprenants : décrit la difficulté à modéliser un processus collaboratif entre les différents apprenants (gestion des communications et des interactions).
- L'évaluation : est nécessaire pour mesurer la performance de l'apprenant, mais cela est plus difficile si le jeu sérieux (ou le STI) est multi-apprenant (ajoute la notion d'évaluation collective, section I.3.2.1).
- Le rejeu : permet de reproduire le parcours de l'apprenant durant une session de jeu pour permettre à l'apprenant et au formateur de ré-visualiser l'activité réalisée pour faire le "*débriefing*".

Tableau II.3 – Tableau comparatif de différents travaux dans le domaine des jeux sérieux (JS) et des systèmes tutoriels intelligents (STI).

Projet	Domaine d'application	Type	Architecture	Multi-apprenants	PNJ	PNJ adaptative	Type d'évaluation	Type de connaissances
[Buche, 2005]	gestion des risques (Porte-avions)	STI	SMA	**oui**	oui	non	individuelle	méta modèle
[Lourdeaux, 2001]	ferroviaire (conducteur TGV)	STI	SMA	non	non	-	individuelle	modèle SKR
[Amokrane, 2010]	gestion des risques (sites SEVESO)	STI	SMA	non	non	-	individuelle	Langage HAWAI-DL
[Binsubaih et al., 2006]	sécurité routière (policier)	JS	Torque GUI	non	non	-	individuelle	ontologie
[Vidani et Chittaro, 2009]	médicale (Infirmière)	JS	Unreal	non (prévu)	non	-	individuelle	modèle CCT
[Mathieu et al., 2011]	Comercial (Supermarché)	JS	SG/SMA	non	**oui**	**oui**	individuelle	SMA
[Nieborg, 2004]	Militaire (armé américaine)	JS	Unreal	**oui**	**oui**	**oui**	-	?
[Johnson et al., 2004]	Militaire (apprentissage de langue)	JS	Unreal	non	**oui**	non	-	BDD SQL
[Wendel et al., 2012]	Travail collaboratif	JS	Unity 3D	**oui**	**oui**	non	-	?

Dans [Lourdeaux, 2001], Lourdeaux propose un outil à base d'ITS pour la formation des conducteur de TGV. Lourdeaux propose le concept de HAL (Help Agent for Learning). L'environnement de formation de HAL est basé sur la réalité virtuelle avec un écran géant et un casque de réalité virtuelle pour afficher l'environnement 3D. Par contre HAL est mono-apprenant et n'a pas de module d'évaluation ni de système de rejeu.

Dans le domaine de la gestion des risques, Buche propose PEGASE (PEdagogical Generic and Adaptive SystEm) un ITS pour la gestion des risques à bord des porte-avions, néanmoins l'outil est très restreint en terme de formation (uniquement les membres d'équipage) et n'a aucune solution pour l'évaluation et le rejeu.

Toujours dans la gestion des risques, Amokrane [Amokrane, 2010] propose un outil de formation pour les sites SEVESO. HERA (Helpful agent for safEty leaRning in virtuAl environment) est un STI qui traite l'évaluation des apprenants et possède un module de rejeu. Notons que HERA est mono-apprenant et ne traite pas du problème de l'évaluation collective. Dans le domaine des jeux sérieux, on peut citer les travaux de Binsubaih [Binsubaih et al., 2006] qui propose un jeu sérieux pour les accidents de circulation pour la formation de policiers. Ce jeu possède un mécanisme très sophistiqué pour l'évaluation, néanmoins il reste pauvre au niveau pédagogique et de l'accompagnement des apprenants.

Dans le domaine de la santé, Vidani [Vidani et Chittaro, 2009] propose un jeu sérieux pour la formation des infirmiers, ce jeu possède un module d'accompagnement

des joueurs mais reste assez limité car il ne donne pas la liberté aux apprenants dans le choix des actions à réaliser. Dans [Mathieu et al., 2011], Mathieu présente une architecture multi-agent pour un jeu sérieux pour l'apprentissage des étudiants de l'école de commerce. Cette approche permet une architecture modulaire et flexible et traite de la question de l'évaluation individuelle notamment avec la présence de PNJs adaptatifs. Néanmoins le jeu est mono-apprenant et ne traite pas l'évaluation collective.

Le jeu America's Army est un modèle dans le domaine des jeux sérieux, mais son utilisation reste plutôt dans le divertissement que l'apprentissage. Certes le jeu permet d'apprendre quelques notion concernant la stratégie d'intervention et le maniement des armes mais reste tout de même un outil pour s'initier au soldat. L'évaluation se fait de manière individuelle via un score comme le nombre de d'adversaire neutralisé ou par objectifs (désamorcer la bombe ou secourir les otages, etc). Dans [Johnson et al., 2004], Johnson présente un jeu sérieux pour l'apprentissage de langue. Le jeu est basé une architecture hybride JS/STI et se base sur un modèle d'erreur (basé sur des modèles linguistiques) pour détecter les erreurs de l'apprenant, néanmoins la correction ce fait en temps réel et ne fournit pas la performance globale de l'apprenant, de plus l'apprentissage est individuelle et ne traite pas de la problématique de l'évaluation collective.

Dans [Wendel et al., 2012], Wendel pointe du doigt le manque de jeux sérieux pour l'apprentissage collaboratif, et tente de proposer une approche de conception de jeux sérieux orienté apprentissage collaboratif. Pour illustrer cette approche Wendel présente le jeu Escape From Wilson Island qui met en œuvre un travail collaboratif dans le but d'une réussite globale, cette réussite globale consiste à collaborer pour survivre et s'échapper d'une île déserte. Malheureusement, et malgré l'aspect collaboratif, l'évaluation des apprenants se fait via des observations de session du jeu.

En comparant ces différents travaux nous avons remarqué que certains se concentrent plus sur l'aspect pédagogique [Buche, 2005], d'autres sur l'aspect immersif [Lourdeaux, 2001], et d'autres encore qui se concentrent sur la représentation des connaissances [Vidani et Chittaro, 2009] [Amokrane, 2010]. Nous avons aussi remarqué que certains travaux sont basés sur des architectures multi-agents [Buche, 2005] [Mathieu et al., 2011], d'autres sur des moteurs de jeux vidéo [Nieborg, 2004] ou encore sur des architectures hybrides [Wendel et al., 2012]. Nous avons enfin remarqué que l'évaluation des apprenants est très peu abordée notamment l'évaluation collective. L'absence d'évaluation collective est surtout dû au nombre faible jeux sérieux multi-joueurs traitant de l'apprentissage collaboratif comme signalé dans [Wendel et al., 2012].

II.7 Conclusion

Dans ce chapitre nous avons vu différentes philosophies concernant l'apprentissage des apprenants dans un environnement virtuel. Nous avons tout d'abord constaté différentes représentations du concept de JS avec différentes architectures et cadres conceptuels, nous avons aussi présenté quelques travaux dans ce domaine. Nous avons ensuite présenté une autre approche de l'apprentissage en environnement virtuel qui concerne les STIs. Nous avons présenté l'architecture des STIs ainsi que différents travaux dans ce domaine.

Notre objectif est de proposer une approche d'évaluation des apprenants dans le cadre d'un apprentissage de tâches collaboratives. Cette évaluation doit prendre en compte l'évaluation individuelle pour chaque individu afin de calculer la performance globale du groupe, cette évaluation peut être hétérogène et distribuée car pour une tâche collaborative, cela implique la participation de plusieurs apprenants avec des compétences hétérogènes mais dont la collaboration permet d'atteindre un objectif global commun. Cette approche doit être intégrée dans une architecture de type jeu sérieux/STI pour prendre en compte les différents aspects de l'apprentissage dans un environnement virtuel. Cette architecture doit être caractérisée tout d'abord par un aspect immersif qui permet de reproduire un monde virtuel, un aspect pédagogique (suivi et support de l'apprenant), un aspect représentation des connaissances qui permet de représenter le domaine étudié ainsi que l'apprenant (traces de l'apprenant), et en fin un aspect social qui permet de représenter l'aspect collaboratif.

Dans le chapitre suivant nous présentons un cadre méthodologique pour l'évaluation individuelle et collective des apprenants basé sur le concept "*d'espace d'évaluation*". Nous illustrons par la suite le concept d'espace d'évaluation dans le cadre d'un JS collaboratif pour la formation à la gestion de crise, en proposant trois espaces d'évaluation. Un espace physique pour évaluer les interactions des apprenants avec l'environnement, un espace comportemental pour évaluer les procédures de la gestion de crise, et enfin un espace social pour évaluer l'aspect collaboratif de la gestion de crise.

Dans le chapitre IV, nous proposons une architecture multi-agents pour la mise en œuvre du cadre méthodologique précédent basé sur le concept d'espace d'évaluation. Cette architecture permet de modéliser et d'intégrer les différents modules d'un STI combinés avec un JS. L'architecture est caractérisée par : un JS, un module d'évaluation, un module pédagogique, un module pour la représentation des connaissances (modèle du domaine, modèle de l'apprenant), et enfin un module pour la simulation des comportements (PNJ).

Deuxième partie

Contributions

CHAPITRE III

Un cadre méthodologique pour l'évaluation individuelle et collective dans les jeux sérieux collaboratifs

Sommaire

III.1 Introduction 66

III.2 Problématique de l'évaluation collective dans les jeux sérieux 66

III.3 Le concept d'espace d'évaluation 68

III.3.1 Définition du concept d'espace d'évaluation 68

III.3.2 Illustration du concept d'espace d'évaluation dans un jeu sérieux collaboratif pour la gestion de crise 72

III.3.2.1 L'évaluation des apprenants dans la formation à la gestion de crise 73

III.3.2.2 Définition des espaces d'évaluation pour la formation à la gestion de crise 74

III.4 Les espaces d'évaluation pour la formation à la gestion de crise 74

III.4.1 Espace physique 75

III.4.1.1 Les composants de l'espace physique 75

III.4.1.2 L'évaluation dans l'espace physique 76

III.4.2 Espace comportemental 76

III.4.2.1 Les composants de l'espace comportemental 77

III.4.2.2 L'évaluation dans l'espace comportemental 79

III.4.3 Espace social 82

III.4.3.1 Les composants de l'espace social 82

III.4.3.2 L'évaluation dans l'espace social 83

III.5 Conclusion 86

III.1 Introduction

Dans ce chapitre nous présentons notre cadre méthodologique pour l'évaluation individuelle et collective dans les jeux sérieux collaboratifs. Ce cadre méthodologique tente de répondre à la problématique d'évaluation d'un groupe d'apprenants en interaction dont le but est d'apprendre une tâche collaborative. Ces apprenants doivent communiquer et collaborer pour atteindre un but global commun. La difficulté de l'évaluation globale est qu'elle doit tenir compte de l'évaluation de chaque apprenant dont les compétences requises diffèrent d'un apprenant à un autre. Dans un premier temps nous présentons le concept d'espace d'évaluation, adapté à cette évaluation hétérogène et distribuée. Dans un deuxième temps nous illustrons le concept d'espace d'évaluation dans le domaine de la gestion de crise en présentant trois espaces d'évaluation : Physique, Comportemental, et Social.

III.2 Problématique de l'évaluation collective dans les jeux sérieux

Comme nous l'avons vu dans la section I.3.1 et I.3.2, les jeux sérieux ont des besoins bien spécifiques que nous avons résumés en : la représentation des connaissances, l'évaluation des apprenants, et le recueil des traces. Ces trois besoins sont intiment liés, car pour évaluer un apprenant il faut déjà représenter la ou les connaissances (compétences) que l'on veut enseigner. Essayons d'illustrer ces besoins dans un cas concret dans le domaine de la gestion de crise.

Comme nous l'avons illustré dans la section I.4.2, la formation à la gestion de crise est un processus collaboratif permettant à plusieurs personnes d'acquérir une compétence (différente ou identique), avec la particularité de ne pas cibler uniquement les spécialistes du domaine (les pompiers essentiellement) mais également des non professionnels du risque (Préfet, Maire, Directeur d'établissement, etc.). La gestion d'une crise majeure peut mobiliser plusieurs centaines d'intervenants, du Préfet dans son bureau au pompier sur le terrain. Ces intervenants sont amenés à communiquer et à collaborer pour rétablir la situation à un état normal. Afin de mieux comprendre l'aspect hétérogène des besoins d'évaluation, nous présentons un exemple de mise en situation.

Considérons le scénario suivant qui commence par un camion TDM (Transport des Matières Dangereuses) qui s'est renversé après un accident de la circulation. Le réservoir

est endommagé et de l'hydrocarbure se répand sur la chaussée. Un témoin de l'accident donne l'alerte en appelant le CODIS (Centre Départemental d'Incendie et de Secours), qui à son tour doit effectuer quatre missions pour répondre à l'alerte. Tout d'abord, le CODIS doit envoyer un pompier sur les lieux pour récupérer les informations sur le sinistre (*Envoyer pompier*). Une fois les informations sur le sinistre reçues (transmises par le pompier sur les lieux) et confirmant l'accident TDM, le CODIS doit donner ensuite des instructions à un officier (pompier) sur les mesures à prendre. Le CODIS doit ensuite remplir une fiche de renseignements sur le sinistre et l'envoyer par fax au Maire, au Préfet et au Sous-Préfet (l'ordre d'envoi n'est pas important). Enfin, la dernière mission est d'informer l'officier du lieu du PCO (Poste de Commandement Opérationnel), une fois ce dernier établi et communiqué par le Préfet.

Ce scénario implique plusieurs acteurs (CODIS, pompier, ...), chacun avec des procédures et des connaissances différentes à évaluer ainsi que l'orchestration des interactions qui doit être respectée. Comme ces informations diffèrent par leur nature, chacune nécessite une représentation et une méthode d'évaluation spécifique. Le diagramme d'activité UML de la figure III.1 représente, de manière simplifiée, les connaissances procédurales de l'acteur jouant le rôle du CODIS, sous la forme de mission à réaliser. Les interactions peuvent être évaluées par le temps écoulé entre la détection du sinistre et sa résolution ainsi que la zone impactée (par ex. zone détruite par l'incident). Chaque connaissance nécessaire à l'évaluation peut utiliser un modèle de représentation différent, ainsi qu'un calcul d'évaluation spécifique pour produire un score d'apprentissage pour valider ou non l'objectif d'apprentissage d'un scénario. Ces connaissances peuvent être produites individuellement ou collectivement (interactions entre acteurs et/ou joueurs simulés).

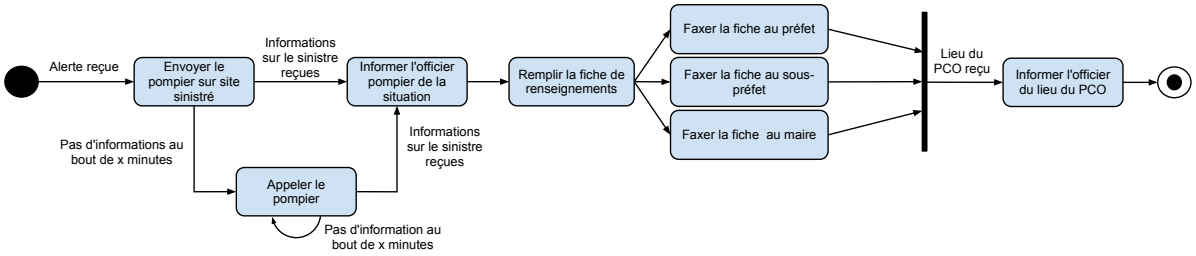


Figure III.1 – Diagramme d'activité UML de la mission du rôle CODIS

Ainsi, l'évaluation des apprenants nécessite la collecte d'informations provenant de différentes sources en fonction du rôle incarné par l'apprenant dans le scénario. Par exemple, pour évaluer la première action de l'acteur CODIS (téléphoner), nous avons besoin de données de l'environnement pédagogique (environnement 3D). Nous avons également besoin de l'historique des actions de l'apprenant pour ensuite les comparer avec la séquence d'actions prévue dans le modèle du domaine (défini par un expert du domaine, figure III.1). Les informations nécessaires pour l'évaluation varient dans leur

type ainsi que leur nature (selon le rôle joué par l'acteur), et devront être acquises auprès de sources de connaissances ou de données distribuées sur différents composants logiciels. Toutes ces informations sont nécessaires pour produire l'évaluation des apprenants, mais chacune répondant à un besoin précis de l'évaluation (correspondant à un objectif d'apprentissage), ou bien d'un certain point de vu (collaboration, procédure, etc).

Dans le cadre d'un Jeu Sérieux préexistant, auquel on souhaiterait ajouter des capacités d'évaluation, il faudra ajouter la possibilité de récupérer ces informations. Qu'une partie de ces informations soit déjà disponible ou non, l'optimisation des coûts requiert que ce besoin d'information remette en cause *à minima* l'architecture logicielle initiale. Cette considération sera abordée dans le chapitre suivant.

III.3 Le concept d'espace d'évaluation

Dans l'optique de l'évaluation individuelle et collective qui est naturellement distribuée (multi-acteurs) et de plus hétérogène (connaissances et compétences hétérogènes), nous proposons un cadre méthodologique basé sur le concept d'espace d'évaluation (ES : evaluation space). Un ES permet l'évaluation des apprenants selon un certain point de vue. Dans cette section nous présentons le concept d'ES et ces différents composants.

III.3.1 Définition du concept d'espace d'évaluation

Comme vu dans la section précédente, intégrer l'évaluation dans un jeu sérieux implique l'utilisation de connaissances, d'informations ou de données, produites ou transformées en permanence jusqu'à la fin du déroulement du scénario de jeu. Chaque élément d'information nécessite une manipulation spécifique (voire un raisonnement sur ces connaissances) afin d'en extraire une évaluation. Une façon naturelle de faire face à la complexité de la gestion de ces informations (au sens large) est de diviser et d'organiser ces informations en groupes homogènes dans lesquels des primitives dédiées peuvent être utilisées pour produire une évaluation. La notion d'espace d'évaluation s'inscrit dans cette démarche en englobant tous les éléments nécessaires pour produire des évaluations, en considérant un scénario de jeu à travers des vues différentes, chacune correspondant à un objectif d'évaluation particulier. Un espace d'évaluation se défini comme un quadruplet :

$$Space = \{Kw, I, M, AM\} \tag{III.1}$$

Avec :

- ***Kw*** [1] **= *un ensemble de modèles de représentation des connaissances :*** comme évoqué dans la section I.3.2.2, différents types de connaissances sont impliqués et chacun est basé sur un paradigme de modélisation spécifique (modélisation de données, base de règles, réseau Bayesien...). Pour assurer l'homogénéité de l'évaluation, un espace est associé à un ensemble de modèles de représentation de connaissances similaires.
- ***I = un ensemble d'indicateurs :*** un indicateur est une information associée à un phénomène défini pour en observer les variations périodiques. Ainsi, un indicateur représente une donnée quantitative qui caractérise une situation en évolution (une action ou les conséquences d'une action).
- ***M = un ensemble de métriques :*** les métriques représentent les méthodes de calcul et les unités de mesure utilisées pour exploiter les connaissances. Ainsi, une métrique permettra de quantifier un indicateur en vue d'une évaluation.
- ***AM*** [2] **= *un ensemble de modèles d'évaluation :*** Il y a différents modèles d'évaluation en fonction de l'espace et de sa représentation des connaissances. L'évaluation peut porter sur une action, sur une procédure ou bien globalement sur une session d'entrainement (cf. les différents types d'évaluation dans la section III.3.2.1). Le calcul d'indicateurs repose sur un modèle d'évaluation spécifique ainsi que sur sa métrique. Selon le modèle de représentation des connaissances, la production d'indicateurs peut requérir des méthodes issues de l'IA (heuristiques, appariement d'ontologies, inférences...).

Représentation des connaissances *KW* : La question des connaissances, informations, données (brutes ou agrégées) a déjà longuement été abordée, justifiant ainsi une place importante dans la caractérisation d'une évaluation et par conséquence dans la définition d'un Espace d'Évaluation. Il s'agit ici de formaliser la manière dont ces connaissances sont représentées afin de proposer un gain de généricité, cette question étant considérée parfois comme secondaire car traitée à un niveau logiciel (solution adhoc au cas par cas) d'autant que l'hétérogénéité des formes de connaissances est faible en général. Que cette question soit abordée en phase de conception du jeu sérieux ou a posteriori (amélioration d'un jeu sérieux existant), cet effort de formalisation permet tout à la fois d'aider le concepteur à appréhender la nature des informations nécessaires au jeu ainsi qu'une meilleure réutilisation de composants.

1. pour Knowledge models
2. pour Assessment Model

Les indicateurs *I* : L'usage d'indicateurs pour l'évaluation est récurrente dans les travaux sur les jeux sérieux [Thomas Benjamin et al., 2012] et les EIAH [Woolf, 2010]. Ils contribuent ou guident un processus d'évaluation renvoyé au moniteur du jeu ou aux joueurs eux-mêmes. Le calcul d'indicateur est un enjeu en soit, construit soit à partir d'un modèle théorique ou de dires d'experts pouvant nécessiter un travail de calibrage et de validation par expérimentation. Ce calcul peut aller de simples calculs arithmétiques à de véritables fonctions de préférences (cf. paragraphe suivant sur les modèles d'évaluation).

Dans le cadre d'évaluation dans un contexte de compétences ou connaissances homogènes, la combinaison d'indicateur (cumul ou moyenne de scores) ne pose pas de problème particulier. Lorsqu'on fait face à une plus grande hétérogénéité, comme dans le cas de la gestion de crise, l'évaluation des compétences des joueurs devra se baser sur des indicateurs de nature différente (unités différentes, qualitatif vs quantitatif,...) et de modes de calculs différents. Ce dernier point justifie la définition de modèles d'évaluation adaptés à l'objectif d'évaluation (voire de micro évaluation).

Ainsi, intégrer des couts financiers, le nombre de victimes touchées, la population déplacée, voire un impact environnemental peut interpeller. Cette situation est tout à fait comparable aux questions qu'ont eu à traiter les travaux dans le domaine de l'aide à la décision multi-critères [Maystre1994]. Aussi, nous sommes nous inspiré de ces travaux pour normaliser un indicateur comme produisant une valeur dans l'intervalle [0,1].

Métrique *M* : La notion de métrique est classiquement associée à la notion de distance, qui peut se reformuler en terme de proximité, comme par exemple lors de la comparaison de graphes, d'analyse de texte etc. En fonction de l'objectif d'évaluation, une même information nécessitera des mesures différentes. Il peut s'agir de comparer les comportements attendus et réels des acteurs (actions, décisions) ou de la manière d'analyser un graphe d'interaction. Ce même graphe d'interaction pourra être associé à des métriques différentes selon ce que l'on cherche : mesure binaire sur la présence ou non d'un élément (0 ou 1 selon qu'il existe ou non) ou mesure d'une intensité d'une interaction en fonction du nombre de messages échangés.

Les modèles d'évaluation *AM* : Ainsi, un modèle d'évaluation (AM) peut être vu comme une fonction utilitaire qui produit une évaluation (score) à partir d'indicateurs I quantifiés par des métriques (M) dans un sous-ensemble de connaissances exprimé selon un mode de représentation (Kw). En fonction des critères désirés, une même information peut produire des indicateurs différents selon la métrique employée. Par exemple, à partir d'un diagramme d'activité décrivant les actions requises d'un acteur,

on peut vouloir aussi bien comparer le nombre d'actions réalisées à celle attendues, que vérifier que l'ordre d'exécution des actions a bien été respecté. La métrique employée différera donc selon l'objectif d'évaluation. Un modèle d'évaluation, par le biais des scores qu'il permet de produire, intervient dans l'évaluation individuelle et collective des apprenants. Cette représentation est décrite ci-dessous :

$$AM : Kw \times M \longrightarrow I \tag{III.2}$$

Pour avoir une évaluation complète, le score d'évaluation doit être produit à partir de plusieurs indicateurs différents (chacun représentant un point de vue ou une propriété différente). Ainsi, pour chaque indicateur (i$\in$ I), un modèle d'évaluation (am $\in$ AM) est défini spécifiquement en fonction de l'objectif d'évaluation. Le calcul d'indicateur, nous l'avons vu, peut revêtir des formes différentes, et là encore, cette hétérogénéité nous renvoie à des questions anciennes déjà traitées par l'aide à la décision multi-critères au travers de fonctions de préférences. Ainsi, à l'image des critères généralisés de la méthode PROMETHEE [Brans et Mareschal, 2005], un modèle d'évaluation pourra se baser sur des fonctions non linéaires permettant de mieux tenir compte par exemple d'effet d'échelle et être plus proche d'une évaluation humaine.

Ces modèles d'évaluation (sorte de micro évaluation, par ailleurs individuelle ou collective) contribueront au résultat final de l'évaluation (AM) d'un Espace d'évaluation (figure III.2). Il pourra résulter de la moyenne des scores produits par chaque modèle d'évaluation (am). Le score d'évaluation (am(i)) produit est représenté par une valeur réelle entre 0 et 1 qui reflète la performance de l'apprenant (ou un ensemble d'apprenants) par rapport à cet indicateur. Un poids peut être ajouté au score d'évaluation (am(i)) pour signifier l'importance relative d'un indicateur (i) ou en d'autres termes le niveau de sa contribution à une évaluation. Cette représentation est décrite ci-dessous dans la figure III.2.

Comme nous l'avons vu dans la section III.2, la variété des compétences et des connaissances associées peut présenter une difficulté de conception (ou de reconception) d'un jeu sérieux à vocation d'apprentissage. En définissant ainsi les constituants d'un espace d'évaluation, nous cherchons à faciliter leur caractérisation indépendamment du domaine d'application auquel s'adresse un jeu sérieux, ainsi que des compétences devant être évaluées dans un jeu sérieux. À travers la définition d'un espace d'évaluation on est amené à formaliser les enjeux de formation (*learning outcomes*). Le concept d'espace d'évaluation peut ainsi faciliter la conception dudit jeu sérieux.

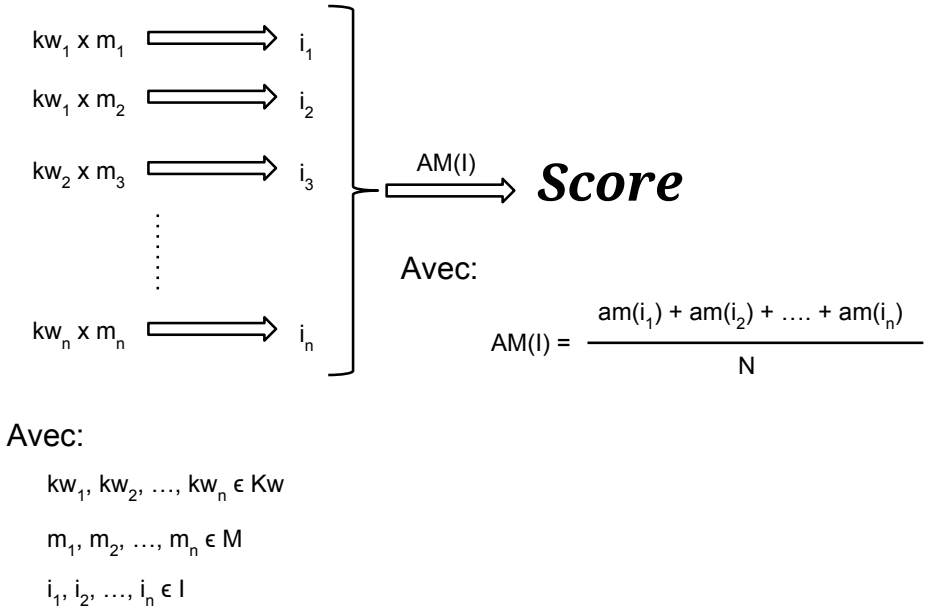


Figure III.2 – Représentation des différents composants de l'espace d'évaluation

III.3.2 Illustration du concept d'espace d'évaluation dans un jeu sérieux collaboratif pour la gestion de crise

L'ajout de capacités d'évaluation à un jeu sérieux pose des questions aussi bien en termes de représentation et manipulation de connaissances, d'acquisition de données, mais surtout de méthodes (au sens mathématique) d'évaluation. La gestion de crise présente des caractéristiques intéressantes de ce point de vue de par son caractère multi-acteurs et collaboratif.

La formation à la gestion de crise (via un jeu sérieux) est un processus comparatif permettant à plusieurs personnes d'acquérir une compétence (différente ou identique), avec la particularité de ne pas cibler uniquement les spécialistes du domaine mais également des non professionnels (du risque). La gestion d'une crise majeure peut mobiliser plusieurs centaines d'intervenants, du Préfet dans son bureau au pompier sur le terrain. Ces intervenants sont amenés à communiquer et à collaborer pour rétablir la situation à un état normal. Devant le nombre d'intervenants, le nombre d'absents à une session d'entrainement peut être non négligeable. De plus, l'objectif de formation d'une session d'entrainement peut cibler un sous ensemble des rôles impliqués dans la gestion d'une crise. Il faut alors pouvoir remplacer les apprenants (joueurs humains) par des joueurs virtuels (question qui sera abordée dans le prochain chapitre).

Le terme *apprentissage collaboratif*, tel qu'il apparaît dans la littérature, cible majoritairement l'apprentissage de compétences homogènes (à l'image d'une classe où les élèves apprennent la même leçon [Jermann et al., 2004] [Giordani et al., 2005], alors

que la gestion de crise est un processus collaboratif qui passe par la réalisation de plusieurs tâches différentes (dépendant du rôle de l'acteur ainsi que du sinistre). Ainsi, la formation à la gestion de crise vise un apprentissage de tâches collaboratives hétérogènes, réalisées par des acteurs différents ayant un objectif commun et présente donc des besoins spécifiques du fait de son caractère multi-acteurs.

Le multi-acteurs requiert deux types d'évaluation : une évaluation individuelle et une évaluation collective. La résolution d'une crise (incendie, pollution...) passe par le respect de toutes les procédures attribuées aux intervenants : dès lors l'évaluation individuelle peut avoir une influence sur l'évaluation collective, et réciproquement. Considérons, par exemple, un apprenant qui a bien effectué ses procédures, mais dont le but global n'a pas été atteint (e.g. pertes matérielles et humaines). Cet apprenant doit être évalué sur sa performance individuelle et collective pour déduire la raison de l'échec (manque de communication, procédure manquante d'un autre apprenant, ...).

III.3.2.1 L'évaluation des apprenants dans la formation à la gestion de crise

L'évaluation des joueurs d'un scénario de jeu, s'appuie sur des capacités d'évaluations mobilisées à des moments variés du jeu, à partir d'indicateurs situés dans des espaces différents. Ainsi, trois types d'évaluations intermédiaires ont été définis : durant la partie, individuelle et collective :

Évaluation durant la partie : Cette évaluation porte sur un diagnostic évoluant tout au long de la partie (scénario de jeu). L'évaluation en temps réel recouvre l'évaluation d'une action ou d'une mission (l'espace des comportements). Pour évaluer une action, le calcul d'un score se base sur les indicateurs correspondants. Par exemple, pour l'action "Fax", l'évaluation consiste à vérifier qu'aucun destinataire n'a été oublié. Pour évaluer une mission, différents indicateurs sont mobilisés (dépendant de la mission à réaliser : temps, préconditions, respect de l'ordre des actions, ...).

Évaluation finale individuelle : L'évaluation finale individuelle correspond à une évaluation sommative qui évalue et certifie l'apprentissage de l'apprenant à la fin du scénario de jeu. Cette évaluation compare l'état final du modèle de l'apprenant (actions de l'apprenant ainsi que ses connaissances) avec le modèle du domaine et établit un diagnostic sur les compétences acquises et les compétences qui restent à apprendre.

Évaluation collective : Utilisant l'espace social, un graphe d'interaction est construit, représentant les différentes communications et les interactions entre les différents acteurs (les apprenants ainsi que les acteurs simulés). Avec ce graphe, nous savons qui a

contacté qui, quand, et pour combien de temps, ainsi que les informations échangées. En combinant les informations provenant des différents espaces, nous avons la possibilité de déduire un lien de causalité entre les missions des différents acteurs (l'acteur A a échoué dans sa mission parce que l'acteur B n'a pas envoyé les bonnes informations) et ainsi obtenir une évaluation complète et précise de tous les acteurs.

L'évaluation globale qui ne peut être déterminée qu'à la fin d'une session de jeu, va intégrer l'évaluation à la fois individuelle et collective. La certification d'une compétence ou des connaissances de l'apprenant peut être obtenue par négociation comme dans [Chadli et al., 2010].

III.3.2.2 Définition des espaces d'évaluation pour la formation à la gestion de crise

La section III.2 montre le caractère hétérogène de l'évaluation dans un scénario de gestion de crise. Pour faire face à cette hétérogénéité, nous avons essayé d'analyser les différents besoins et caractéristiques de l'évaluation des apprenants pour la gestion de crise. La gestion de crise est un processus collaboratif, aussi il nous faut un espace pour évaluer l'aspect collaboratif de l'apprentissage, i.e les interactions entre les différents acteurs. La gestion de crise est aussi l'application de certaines procédures prédéfinies (section I.4) selon la fonction incarnée. Incarner un rôle permet à l'apprenant d'adopter un certain comportement pour réaliser son objectif. L'apprenant évolue dans un environnement virtuel et interagit avec ce dernier via son avatar. Cet environnement regroupe plusieurs informations nécessaires à l'évaluation comme la position des acteurs, les moyens, les différentes infrastructures de la ville comme les bâtiments (type, nombre d'habitants, etc.), routes, bouches d'égouts, etc.

Après avoir analysé les différentes caractéristiques de l'évaluation des apprenants pour la gestion de crise (interactions, comportements, environnement) nous avons défini trois différents espaces d'évaluation : Comportemental, Physique et Social. Ces différents espaces d'évaluation sont décrits dans la section suivante.

III.4 Les espaces d'évaluation pour la formation à la gestion de crise

La mise en œuvre des différentes évaluations, citées dans la section précédente, s'appuie sur le concept d'espace d'évaluation déjà proposé. Nous montrons plus en détail ici, comment ce concept est instancié sur un domaine d'application qui est la gestion de crise à travers trois espaces d'évaluation différents et complémentaires : physique,

comportemental et social.

III.4.1 Espace physique

L'espace physique représente le monde virtuel, avec les avatars des acteurs, les moyens mis à disposition des joueurs (voitures, téléphone, fax...) et l'environnement (bâtiments, routes, arbres...). Ces données peuvent évoluer dans le temps comme l'état d'un sinistre ou la position d'un acteur... Le monde virtuel est représenté par l'interface 3D du jeu sérieux. Le traitement et la manipulation de ces données sont effectués durant la partie par la simulation du jeu ainsi que par les différents mécanismes du moteur de jeu (interaction, animation, ...). L'évaluation consiste en l'agrégation de ces données via des expressions mathématiques. Par exemple, l'indicateur pertes matérielles et humaines va permettre l'évaluation de la performance globale des joueurs. Cet indicateur peut avoir comme métrique le montant en euros des pertes ainsi que le nombre de blessés et de morts.

III.4.1.1 Les composants de l'espace physique

Représentation des connaissances *KW* : L'espace physique représente l'environnement virtuel du jeu sérieux, les connaissances (Kw) modélisées dans cet espace sont relatives aux informations des modèles 3D (avatars, moyens, sinistres, végétation, arbres, etc.) ainsi que les métadonnées du système d'informations géographique (SIG) comme le type d'un bâtiment (commerce, résidence, école, etc.), le nombre d'habitants dans un bâtiment, etc. Ainsi ***Kw*** sera modélisé comme suit :

$$Kw = \{modèles\ 3D, SIG\} \tag{III.3}$$

Les indicateurs *I* : Les indicateurs (I) utilisés dans cet espace sont utilisés pour produire une évaluation à la fois individuelle et collective. Ainsi, par exemple, le déplacement d'un avatar d'un point A à un point B (évaluation individuelle) requiert la position de départ, la position d'arrivée, le temps écoulé. Par ailleurs, l'espace physique peut aussi fournir des indicateurs pour l'évaluation collective : par exemple les pertes humaines et matérielles. Cet indicateur peut donner une idée sur la performance globale du groupe. On a :

$$I = \{temps, positions, ressources, coût, etc.\} \tag{III.4}$$

Les métriques *M* : La métrique (M) va consister à quantifier (unité, fonction de calcul de distance, etc.) les indicateurs définis auparavant. Par exemple, la métrique pour un indicateur de type pertes matérielles peut être le coût en Euros. Ainsi les métriques auront la forme suivante :

$$M(I) = unité/fonction \tag{III.5}$$

par exemple :

$$M(temps) = seconde \tag{III.6}$$

$$M(position) = distance\,euclidienne \tag{III.7}$$

Les modèles d'évaluation *AM* : Le modèle d'évaluation représente la façon de calculer la performance (par exemple privilégier le temps de parcours plutôt que coût du trajet). Le modèle d'évaluation peut être représenté par une fonction objective qui inclut différents indicateurs (pertes humaines et matérielles, moyens mis en œuvre, etc.) pour calculer la performance (score) des apprenants.

$$Score = AM(I_M) \tag{III.8}$$

III.4.1.2 L'évaluation dans l'espace physique

Comme nous l'avons vu précédemment, l'espace physique concerne tout ce qui est relié à l'environnement virtuel d'apprentissage. Ainsi l'évaluation dans cet espace correspond aux actions qui auront des effets sur l'environnent (position, état d'un modèle, etc).

III.4.2 Espace comportemental

L'espace comportemental comprend les actions et les connaissances des acteurs, ainsi que les différentes informations sur les compétences et procédures à apprendre. Ces informations correspondent au "*modèle du domaine*" et au "*modèle de l'apprenant*" dans un STI. Le modèle du domaine est statique (il n'évolue pas au cours du jeu), composé par exemple d'une ontologie et de plans [Thomas Benjamin et al., 2012], et il est défini par un expert du domaine. Le modèle de l'apprenant est dynamique (il évolue durant l'expérience de jeu de l'apprenant), et il est alimenté par les actions réalisées par l'apprenant ainsi que par les connaissances acquises. Dans cet espace nous avons

à évaluer des actions ainsi que des missions, via des indicateurs dédiés. Par exemple pour l'action téléphoner, les indicateurs sont la durée de communication, mesurée en secondes (métrique), les informations échangées, ainsi que l'acteur appelé. L'évaluation se basera sur ces indicateurs pour attribuer un score à l'action entreprise.

III.4.2.1 Les composants de l'espace comportemental

Représentation des connaissances L'espace comportemental regroupe les actions et les connaissances des acteurs, ainsi que les différentes informations sur les compétences et procédures à apprendre (correspond aux modèles du domaine et de l'apprenant d' un STI). Les connaissances (Kw) qui interviennent dans cette espace sont modélisées sous forme de missions et d'actions. Chaque rôle (assigné à un acteur) se voit attribuer un ensemble de missions à réaliser pour un scénario donné. Une mission est constituée d'un ensemble de préconditions, d'un ensemble de missions qui la précédent, et d'un ensemble d'actions à réaliser pour atteindre l'objectif de la mission. Les actions sont déclinées en plusieurs types : téléphoner, envoyer un fax, parler, parler en utilisant un canal de la radio (pompier), se déplacer (à pied ou en véhicule), utiliser la main courante (portail web pour le suivi du sinistre). Ainsi ***Kw*** est représenté comme suit :

$$Kw = \{Modèle\ du\ domaine, Modèles\ de\ l'apprenant\} \tag{III.9}$$

avec :

$Modèle\ du\ domaine = \{Missions_{role\,1}, Missions_{role\,2}, ..., Missions_{role\ n}\}$,
avec $Mission = \{Préconditions, MissionPré, Actions\}$,
où $Actions = \{telephone, parler, fax, radio, main\ courante, deplacement\}$

et :

$$Modèle\ de\ l'apprenant_{role} = \{Actions\}$$

Les indicateurs Pour chaque action nous avons défini un indicateur spécifique (I) qui permet de calculer l'efficacité de l'action. Pour l'évaluation des missions, nous avons défini six indicateurs (I) généraux qui sont présentés dans le Tableau III.1. Ce tableau montre les différents indicateurs utilisés ainsi que leur mode de calcul (AM). Les indicateurs sont calculés de manière régulière (mise à jour dirigée par le temps ou des événements) et sont basés sur des variables qui peuvent elles-mêmes exiger un calcul spécifique.

D'autres indicateurs spécifiques peuvent être requis pour certaines missions, par exemple : la mission "*intervention sur le sinistre*" évaluera la performance selon les dégâts causés par le sinistre.

Tableau III.1 – les indicateurs pour l'évaluation d'une mission dans l'espace comportemental.

Indicateur	Objectif	Formule	Variable
Pré-conditions	Évalue le respect des pré-conditions de la mission	$S_{prec} = \frac{NbRespectedPrec}{NbPrec}$ (III.10)	*NbRespectedPrec* : pré-conditions respectées *NbPrec* : pré-conditions de la mission
Ordonnancement des actions	Évalue le respect de l'ordre des actions	$S_{order} = \frac{NbOrderedAction}{NbAction}$ (III.11)	*NbOrderedAction* : nombre d'actions réalisées dans l'ordre, *NbAction* : nombre total des actions de la mission
Nombre d'actions	Évalue le nombre d'action réalisées (tentées ou en plus)	$S_{order} = \frac{NbOrderedAction}{NbAction}$ (III.12)	*NbActionPerformed* : nombre d'actions effectuées par l'apprenant *NbAction* : nombre d'actions de la mission
Durée de la mission	Évalue la rapidité à réaliser la mission	$S_{Time} = \frac{1}{1+\exp^{-(\frac{plannedTime}{takenTime})}}$ (III.13)	*plannedTime* : temps estimé par l'expert, *takenTime* : temps mis par l'apprenant pour effectuer la mission.
Temps d'inactivité	Évalue le temps perdu par l'apprenant (durant les missions)	$S_{idleTime} = 2 * (1 - \frac{1}{1+\exp^{-idleTime}})$ (III.14)	*idleTime* : temps d'inactivité de l'apprenant.
Efficacité des actions	Évalue l'efficacité des actions effectuées par l'apprenant	$S_{Efficiency} = \frac{\sum_{i=0}^{nbAction} S_{action_i}}{NbAction}$ (III.15)	S_{action_i} : efficacité de l'action *i*, *NbAction* : nombre d'action de la mission

Les métriques Les métriques utilisées dans cet espace sont liées aux indicateurs, et sont principalement représentées par le temps ou des propriétés logiques comme des pré-conditions, ou de l'ordonnancement. Pour avoir un poids égal entre l'ensemble des indicateurs d'une mission, le score maximal pour chaque indicateur à été normalisé et aura comme valeur maximale la valeur 1.

Les modèles d'évaluation Le modèle d'évaluation AM qui permet de calculer le score de la mission est la moyenne des différents scores produits par les indicateurs définis dans le tableau III.1. Nous avons aussi défini pour chaque action un modèle d'évaluation spécifique. Prenons par exemple l'action téléphoner, nous avons comme in-

dicateur (I) : le temps de communication (la métrique sera le ratio du temps prévu par l'expert avec le temps effectivement réalisé), le destinataire, et les informations échangées (en cas de dialogue textuel avec un GA). Le calcul de la performance de l'action (AM) sera la moyenne de ces différents indicateurs.

III.4.2.2 L'évaluation dans l'espace comportemental

Pour avoir une idée plus précise de l'évaluation dans l'espace comportemental, nous allons prendre une séquence de jeu pour un acteur et essayer d'analyser l'influence des différents indicateurs. Reprenons l'exemple présenté dans la section III.2 avec l'acteur CODIS. Une fois que le CODIS reçoit les informations sur le sinistre (de la part du pompier sur les lieux), le CODIS a pour mission d'informer les autorités de l'accident TMD. Les actions à réaliser pour cette mission sont décrites dans le tableau III.2.

Tableau III.2 – Les actions à réaliser par le CODIS dans le cadre de la mission : *informer les autorités*.

Action	Action type	Indicateur	Score
Téléphone(durée ,dest, msg) = (77, officier, $\{TMD\}$)	Téléphone(45, officier, $\{TMD\}$)	ActionPhoneIndicator(stemps, cible, msg) = (45/77, 1, 1)	0.86
Fax(durée ,dest, nomFax) = (232, officier, ficheRenseignements)	Téléphone(120, officier, ficheRenseignements)	ActionFaxIndicator(stemps, cible, nomFax) = (120/232, 1, 1)	0.83
Fax(durée ,dest, nomFax) = (13, Maire, ficheRenseignements)	Téléphone(120, Maire, ficheRenseignements)	ActionFaxIndicator(stemps, cible, nomFax) = (1, 1, 1)	1
Fax(durée ,dest, nomFax) = (11, Préfet, ficheRenseignements)	Téléphone(120, Préfet, ficheRenseignements)	ActionFaxIndicator(stemps, cible, nomFax) = (1, 1, 1)	1
Fax(durée ,dest, nomFax) = (10, Sous-Préfet, ficheRenseignements)	Téléphone(120, Sous-Préfet, ficheRenseignements)	ActionFaxIndicator(stemps, cible, nomFax) = (1, 1, 1)	1

La première colonne du tableau représente l'action réalisée par l'apprenant avec ses propriétés, par exemple, pour l'action *téléphone*, on aura la durée de communication, le destinataire, et enfin les messages échangés. La deuxième colonne représente l'action type (issue du modèle du domaine) qui servira de base pour l'évaluation de l'action. La troisième colonne représente l'indicateur définit pour évaluer l'action, par exemple pour l'action *téléphone*. Comme nous l'avons mentionné auparavant, le score sera une moyenne entre un score lié au temps, un score lié à l'interlocuteur, et enfin un score lié aux messages échangées. Le score du temps est représenté par un ratio du temps prévu par l'expert sur le temps mis par l'apprenant, si le temps mis par l'apprenant est inférieur au temps prévu par l'expert, le score du temps sera égal à 1. Le score de l'interlocuteur est égal à 1 si l'apprenant a appelé la bonne personne, et 0 sinon. Le score des messages échangés est le ratio des messages envoyés prévus par l'expert sur les messages envoyées par l'apprenant (les messages sont liés à des phrases qui sont

échangées durant le dialogue, voir section V.4.2). La dernière colonne représente le score de l'évaluation de l'action. Dans les deux premières actions, l'apprenant a un peu tardé à réaliser l'action, ce qui réduit un peu son score. Le temps d'exécution des trois dernières actions est extrêmement faible par rapport aux prévisions de l'expert. Ceci est dû au fait que l'apprenant a retransmis le même fax déjà envoyé à l'officier (absence de phase d'édition).

Tableau III.3 – Les actions à réaliser par le CODIS dans le cadre de la mission : *informer les autorités*.

Indicateur	Résultat	Score
Pré-conditions	$Prec = \{TMD\}$	1
Ordonnancement	$Actions = \{telephone, fax, fax, fax, fax\}$	1
Nombre d'action	$NbAction = 5, NbActionExpert = 5$	1
Durée	$Temps = 589, TempsExpert = 525$	0.7
Temps d'inactivité	$TempsInactivité = 246$	0.03
Efficacité des actions	$ScoreAction = \{0.86, 0.83, 1, 1, 1\}$	0.93
Score mission		0.77

Intéressons-nous maintenant au score de la mission, le tableau III.3 décrit les différents indicateurs de la mission ainsi que le score.

Le premier indicateur correspond au respect des pré-conditions, l'apprenant (rôle CODIS) a déclenché la mission après avoir reçu les informations (TMD) de la part du pompier sur les lieux et ainsi satisfait la pré-condition *TMD* et le score étant égal à 1.

Le deuxième indicateur concerne l'ordonnancement des actions, l'apprenant à réaliser les actions dans le bon ordre, ainsi le score est égal à 1.

Le troisième indicateur concerne le nombre d'actions réalisées durant la mission (tentative ou actions inutiles), l'apprenant a réalisé le nombre d'actions prévu par l'expert, ainsi le score est égal à 1.

Le quatrième indicateur concerne le temps d'exécution de la mission (comptabilisé à partir de la première action de la mission jusqu'à la dernière). Le temps prévu par l'expert est de 525 secondes et l'apprenant a mis 589 secondes pour finir la mission, ce qui donne un score de 0.7. Le score du temps d'exécution est relatif au temps d'exécution prévu par l'expert un delta de 5 minutes (300 secondes) ne signifie pas que le score soit mauvais car si le temps prévu par l'expert est de 60 minutes (score de 0.85). Alors que si le delta est court, par exemple 30 secondes et que le prévu par l'expert est de 30 secondes cela donne un mauvais score (0.5). La figure III.3 montre la courbe de variation du score du temps d'exécution par rapport au delta (différence entre le temps prévu par l'expert et le temps mis par l'apprenant) pour une mission de 300 secondes contre une de 30 secondes.

Le cinquième indicateur concerne le temps d'inactivité de l'apprenant. Ce temps est calculé en additionnant le temps d'inactivité de l'apprenant entre chaque action de la

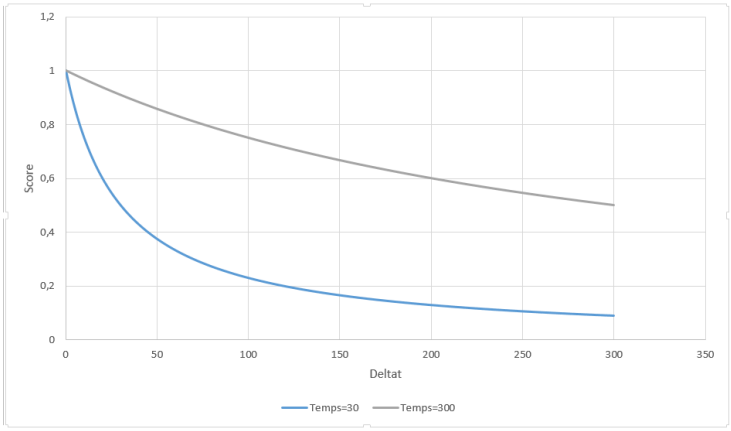


Figure III.3 – Évolution du score du temps d'exécution selon le delta pour une durée de mission de 30s et 300s

mission. On remarque que le temps d'inactivité de l'apprenant est élevé car entre le cumul du temps d'exécution des cinq actions (343s) et le temps d'exécution de la mission (du début de la première action jusqu'à la fin de la dernière action, 589), nous remarquons une différence de 246 secondes, ainsi le score d'inactivité est de 0.03. Le temps d'inactivité a une grande influence sur le score, car en gestion de crise, l'apprenant ne peut pas perdre de temps entre l'exécution de chaque action d'une même mission. La figure III.4 donne un aperçu du score d'inactivité par rapport au temps d'inactivité.

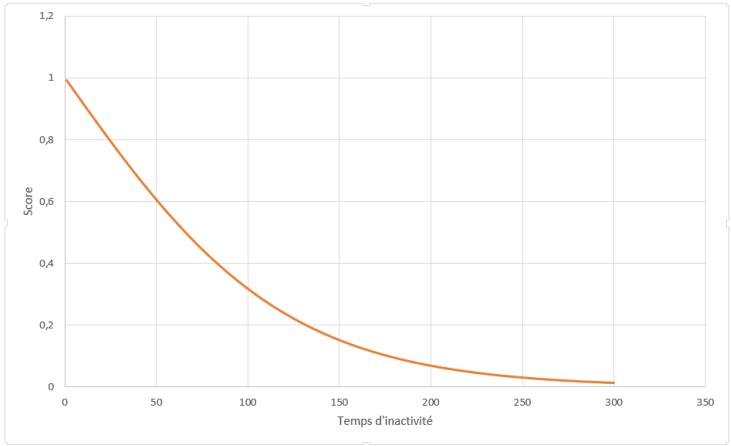


Figure III.4 – Évolution du score du temps d'inactivité selon le temps d'inactivité

Enfin le sixième et dernier indicateur concerne l'efficacité des actions, cet indicateur est calculé par la moyenne des scores des actions présentées dans le tableau III.2. Ainsi le score de l'efficacité des actions est de 0.93. Le score global de la mission est la moyenne des six indicateurs et est égal à 0.77. Après (ou durant la partie) l'apprenant peut avoir des indications sur sa performance, ainsi il peut remarquer que son score d'inactivité est faible et qu'il n'était pas assez réactif pour enchainer les actions de cette mission.

III.4.3 Espace social

L'espace social représente l'interaction (communication simple, coordination / coopération, ...) entre les différents acteurs intégrant ainsi la dimension collective de l'apprentissage. L'espace social est représenté par un graphe qui reprend chaque interaction entre les acteurs et permet un calcul d'une force d'interaction entre chaque acteur ainsi que le degré d'intégration des acteurs du réseau. L'évaluation sera basée sur ces mesures. On retrouve ici les considérations de [Mathieu et al., 2011] sur la représentation et l'exploitation des interactions pour l'évaluation des apprenants. Les indicateurs représentés dans cet espace sont le couplage du réseau et la force d'interaction entre les acteurs et sont calculés comme dans [Miller, 1978]. Les échanges (entre les acteurs) courts et fréquents produisent un indice de couplage fort, les échanges longs et rares traduisent un couplage faible. Cet indicateur va déterminer si les acteurs ont bien communiqué avec les bonnes personnes.

III.4.3.1 Les composants de l'espace social

Représentation des connaissances L'espace social représente l'interaction sociale entre les différents acteurs intégrant ainsi la dimension collective de l'apprentissage. Cet espace nous permet d'évaluer l'aspect collectif du groupe. Les connaissances (Kw) modélisées dans cet espace sont représentées par un ensemble d'interactions, où chaque interaction est caractérisée par :

- l'acteur initiateur de l'interaction
- l'acteur cible
- le type d'interaction (appel téléphonique ou radio, échange de fax, etc.)
- l'heure de l'interaction
- la durée de l'interaction

$$Kw = \{SocialInteraction\} \tag{III.16}$$

avec :

$$SocialInteraction = \{Acteur_{origine}, Acteur_{cible}, Type, date, durée\} \tag{III.17}$$

Les indicateurs Les indicateurs dans l'espace social sont modélisés par le couplage d'un réseau (d'acteur). Les interactions entre les acteurs vont nous permettre de calculer le couplage du réseau à l'image du réseau d'interaction dans [Kay et al., 2006], mais en tenant compte de considérations spécifiques au domaine de la gestion des risques et plus

particulièrement de l'étude des interactions entre pompiers lors d'une campagne [Baumard et Vidal, 2009]. Le couplage entre deux entités A et B (qui fait office d'indicateur) fait référence à la fiabilité avec laquelle on peut prédire le comportement de B, connaissant le comportement de A.

Les métriques Les métriques utilisées pour calculer le couplage du réseau sont modélisées par la durée de l'interaction, le nombre d'interaction, et enfin la durée entre chaque interaction.

Les modèles d'évaluation [Miller, 1978] [Weick, 1982] [Baumard et Vidal, 2009]. Baumard et Vidal [Baumard et Vidal, 2009] proposent de mesurer le couplage au niveau de deux individus i et j, par la formule suivante :

$$C_{i,j} = \frac{f_{i,j}}{d_{i,j}} \tag{III.18}$$

avec :

- $f_{i,j}$, la fréquence des interactions entre i et j est ainsi :

$$f_{i,j} = \frac{n_{i,j}}{T} \tag{III.19}$$

  (i.e. nombre d'interactions entre i et j divisé par la durée totale considérée).
- $d_{i,j}$, la moyenne arithmétique des durées des interactions $d'_{i,j}$, pondérée par les temps entre chaque interaction $\Delta_{i,j}$:

$$d_{i,j} = \frac{\sum d'_{i,j} \times \Delta_{i,j}}{T'},\ Avec\, T' = \sum \Delta_{i,j} \tag{III.20}$$

Ainsi Baumard et Vidal définissent le couplage opérationnel total par la moyenne harmonique des couplages de chaque paire, pondérée par leur intensité (nombre d'interactions) sur la période considérée.

III.4.3.2 L'évaluation dans l'espace social

A partir du couplage entre acteurs, nous proposons de comparer le couplage de l'exercice et le couplage prévu par l'expert qui est calculé à partir du modèle du domaine. Pour cela nous simulons le scénario d'exercice tel que défini par l'expert et nous extrayons les différentes interactions entre les acteurs (en respectant l'ordre des missions, les pré-conditions, etc). À partir de ces interactions "*SocialInteraction*", nous calculons le graphe référence qui sera utilisé pour évaluer l'aspect collaboratif de l'exercice. Cette évaluation consiste à calculer la différence entre le couplage référence (entre deux

acteurs) et le couplage de l'exercice. Ce procédé d'extraction est illustré dans la figure III.5.

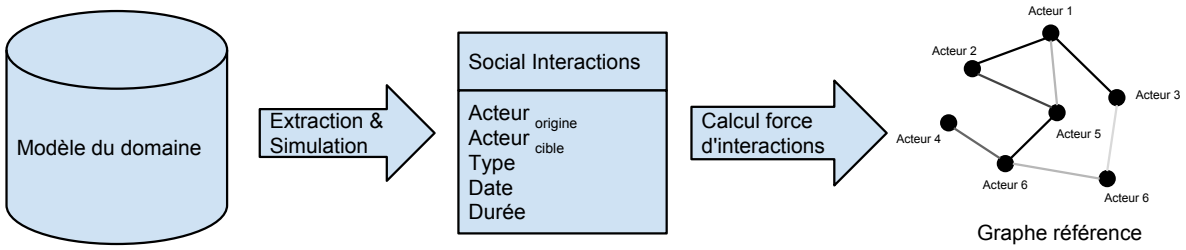


Figure III.5 – Extraction du graphe référence à partir du modèle du domaine.

Ainsi, si le couplage est faible entre A et B et que le couplage calculé via le modèle du domaine est fort, cela signifie qu'il y a eu un manque de communication entre A et B durant l'exercice. Le couplage global quand à lui, permet de suivre dans le temps l'évolution du couplage du réseau et ainsi désigner les moments clés de l'exercice.

Le Tableau III.4 illustre les données utilisées pour la construction du graphe (a) de la Figure III.6. Ce tableau affiche le couplage entre chaque paire d'acteurs ayant participé au scénario. Les valeurs du Tableau III.4 sont normalisées pour simplifier l'exploitation et la construction du graphe (couplage maximal égal à 1).

Tableau III.4 – Exemple de calcul de la force d'interaction durant l'exercice.

	P1	P2	P3	O	C	Ma	Pr	S-p	D	Me	Po	T
Pompier1	-	0	0	0.269	0.602	0	0	0	0	0	0	0.269
Pompier2	0	-	0	0.269	0	0	0	0	0	0	0	0
Pompier3	0	0	-	0	0	0	0	0	0	0	0	0
Officier	0.269	0.269	0	-	0.237	0	0.269	0.244	0	0	0	0
Codis	0.602	0	0	0.237	-	0.073	0.483	0.357	0	0	0	0
Maire	0	0	0	0	0.073	-	0.134	0.141	0.384	0	0	0
Préfet	0	0	0	0.269	0.483	0.134	-	1	0	0.384	0	0
Sous-Préfet	0	0	0	0.244	0.357	0.141	1	-	0	0	0	0
Directeur	0	0	0	0	0	0.384	0	0	-	0	0	0
Média	0	0	0	0	0	0	0.384	0	0	-	0	0
Policier	0	0	0	0	0	0	0	0	0	0	-	0
Témoin	0.269	0	0	0	0	0	0	0	0	0	0	-

La Figure III.6 montre un exemple de couplage pour un exercice de gestion de crise (la couleur d'un arc représente l'intensité du couplage entre deux acteurs). Le graphe (a) représente la force d'interaction du groupe durant l'exercice (une couleur noire signifie un couplage maximal). Ici, le couplage maximal est obtenu par la relation entre le Préfet et le Sous-Préfet. Le graphe (b) représente le graphe référence pour l'exercice (construit à partir du modèle du domaine défini par l'expert). Le graphe (c) représente la différence entre le graphe de l'exercice et le graphe référence. La couleur rouge représente une différence négative, c'est-à-dire que le couplage lors de l'exercice entre

les deux acteurs est inférieur au couplage référence (manque de communication), par exemple entre le Préfet et le Maire. La couleur bleue représente une différence positive c'est-à-dire que le couplage lors de l'exercice entre les deux acteurs est supérieur au couplage référence (surplus de communication), par exemple entre le CODIS et le pompier1.

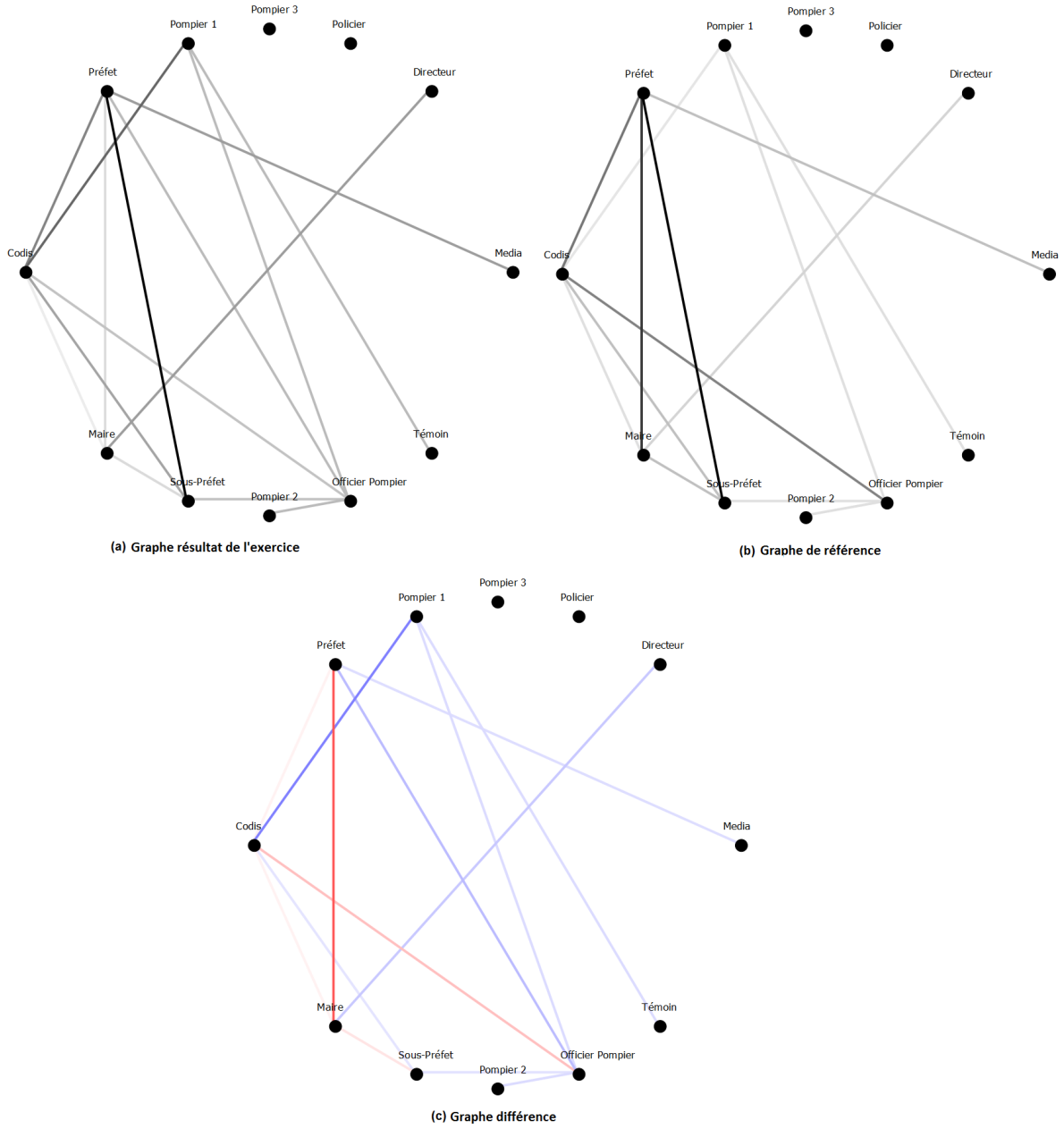


Figure III.6 – Exemple de calculs de couplages pour un exercice de gestion de crise

Ainsi grâce au calcul du couplage du réseau, nous pouvons répondre à l'évaluation individuelle d'un acteur dans ses relations aux autres (en comparant les interactions entre les acteurs) et à l'évaluation collective (grâce au couplage global du réseau) en comparant le couplage résultant de l'exercice avec celui prévu par l'expert.

Dans l'espace social, l'indicateur principal est représenté par la différence de couplage entre le graphe de l'exercice et celui de référence (graphe différence). Ainsi cet indicateur joue un rôle dans l'évaluation globale du groupe (plus la différence sera grande plus le score de l'évaluation sera mauvais). Le calcul du score global en utilisant la différence du couplage représente une évaluation automatique, mais le graphe d'interaction peut aussi servir pour une évaluation dirigée et permettre à un moniteur humain qui suit l'exercice de repérer les situations à problème (tels les outils de visualisation [Kay et al., 2006] évoqués en section I.3.2.1). En effet le graphe différence peut pointer un souci de communication entre deux acteurs, et permettre au moniteur de donner des indications en temps réel afin de résoudre le problème.

Les traces recueillies dans l'espace social sont représentées par les interactions entre les différents apprenants. Ces traces sont quantitatives et peuvent juste faire ressortir une différence avec le modèle référence, néanmoins, ces traces peuvent être vérifiables qualitativement en se basant sur le modèle du domaine (en se basant sur les missions des acteurs ou d'autres indicateurs), car il est très difficile d'interpréter la discussion entre deux acteurs humains (coût du traitement, complexification de la création de scénario etc.).

III.5 Conclusion

Dans ce chapitre nous avons présenté un cadre méthodologique pour l'évaluation individuelle et collective pour les jeux sérieux collaboratifs. Ce cadre méthodologique est caractérisé par le concept d'espace d'évaluation. Un espace est composé par une représentation des connaissances, d'un ensemble d'indicateurs, de métriques de modèles d'évaluations. Nous avons ensuite illustré ce concept dans un domaine d'application qui est la formation à la gestion de crise. Nous avons présenté trois points de vue caractérisés chacun par un espace : un espace comportemental qui représente les actions des différents acteurs, un espace physique qui représente le monde virtuel, et enfin un espace social qui reflète les différents interactions entre les acteurs (évaluation collective). Nous avons aussi détaillé les différents indicateurs, métriques, et modèles d'évaluations pour chaque espace, et comment à partir d'un ensemble d'évaluations individuelles hétérogènes et distribuées arriver un une évaluation collective du groupe.

Dans le chapitre suivant nous allons voir comment mettre en œuvre ce cadre méthodologique d'évaluation (qui représente l'évaluation individuelle et collective) et comment l'intégrer dans l'architecture d'un jeu sérieux pour avoir un outil d'apprentissage optimal qui combine à la fois les avantages des jeux sérieux (immersion et mécanisme de jeu), de l'apprentissage collaboratif (multi-joueurs), et le support et le suivi de l'apprenant (pédagogie et représentation des connaissances).

CHAPITRE IV

Une architecture multi-agents pour l'évaluation individuelle et collective dans les jeux sérieux

Sommaire

IV.1 Introduction 88
IV.2 Pourquoi les SMA ? 88
IV.3 Évolution d'un jeu sérieux vers un Système Tutoriel Intelligent 89
IV.3.1 Rôle du jeu sérieux 90
IV.3.2 Présentation générale de l'architecture multi-agents 90
IV.4 Module d'évaluation 92
IV.4.1 Agent Source de Données 93
IV.4.2 Agent Indicateur 94
IV.4.3 Agent évaluateur 95
IV.5 Module de simulation des comportements 96
IV.5.1 Modélisation des agents de jeu 97
IV.5.2 Modélisation des actions 98
IV.5.3 Modélisation des dialogues 100
IV.6 Agent de l'apprenant 100
IV.7 Agent pédagogique 101
IV.8 Représentation des connaissances 102
IV.8.1 Modèle du domaine 102
IV.8.2 Modèle de l'apprenant 103
IV.9 Interaction entre les différents agents 104
IV.10 Conclusion 106

IV.1 Introduction

Dans ce chapitre nous présentons comment faire évoluer l'architecture d'un jeu sérieux pour tenir compte de l'apprentissage collaboratif (notamment l'évaluation collective) ainsi que du support et l'évaluation des apprenants tout en gardant son aspect immersif et ludique. Plus précisément, il s'agit de montrer comment opérationnaliser le cadre méthodologique proposé dans le chapitre précédent dans un JS collaboratif. Cette évolution se fait via une architecture multi-agent intégrant différents aspects de l'apprentissage collaboratif et des STIs.

IV.2 Pourquoi les SMA ?

Les systèmes multi-agents (SMA) ont déjà été utilisés dans les STI et les jeux sérieux ainsi que nous l'avons vu dans la section II.4 et ont montré leur intérêt par rapport à d'autres systèmes (comme les systèmes experts).

Les SMA présentent de nombreux avantages aussi bien au niveau conception qu'au niveau architectural, comme, par exemple :

- Permettre aux agents d'avoir un comportement indépendant de l'expertise de la tâche à enseigner grâce à des mécanismes de prise de décisions (évaluation des apprenants).
- Permettre au formateur et au concepteur du scénario de décrire aisément des plans en assemblant des sous-plans modulaires (simulation de comportements).
- Décrire des architectures distribuées. Ainsi le concepteur peut décrire facilement pour chaque tâche les comportements associés (collecte de données, évaluation, support pédagogique, etc).

Ainsi avec une approche de conception orienté agent, nous pouvons définir une architecture modulaire et distribuée pour une plateforme d'apprentissage collaboratif qui prend en compte l'évaluation individuelle et collective. Les agents peuvent intervenir au niveau de l'évaluation individualisée des apprenants, grâce à des mécanismes de prise de décision (est ce que l'action effectuée est valide ? l'apprenant a t'il choisi le bon objectif ? etc.) [Lourdeaux, 2001], ainsi que dans l'évaluation collective, par exemple par processus de négociation (ou collaboration) afin de valider ou pas l'exercice [Chadli et al., 2010]. Les SMA peuvent aussi intervenir dans la simulation des PNJ (Personnages Non Joués), au travers de modèles de comportements simples ou plus complexes, lorsqu'ils requièrent des prises de décision tenant compte d'un contexte, avec par exemple un raisonnement orienté but [Rao et al., 1995]. Enfin les SMA peuvent intervenir pour la description (voire la mise en oeuvre) de l'architecture distribuée grâce à leur aptitude sociale (communication entre agents) et leur autonomie [Serment et al., 2008].

Dans la section IV.3.2 nous présentons une architecture multi-agents d'un environnement virtuel pour l'apprentissage collaboratif. Nous montrons comment les agents permettent d'améliorer l'apprentissage grâce à l'intégration du concept d'espace d'évaluation présenté dans le chapitre III.

IV.3 Évolution d'un jeu sérieux vers un Système Tutoriel Intelligent

Dans le chapitre III, nous avons présenté les besoins en termes d'informations (au sens large) nécessaires pour produire une évaluation des apprenants. Afin de faire évoluer un jeu sérieux en un véritable environnement de formation (*Training System*), l'architecture logicielle doit s'adapter aux différents composants logiciels (environnement 3D, simulation multi-agents, simulation des interactions humaines, base de données, etc.) tout en respectant l'hétérogénéité des connaissances nécessaires à l'évaluation des apprenants.

Comme nous l'avons vu dans la section I.3, les jeux sérieux montrent certains besoins dans le domaine du suivi et l'évaluation des apprenants, des besoins auxquels peuvent répondre en partie les différents modules d'un STI notamment pour la représentation de l'apprenant (avec le modèle de l'apprenant) ainsi que le domaine étudié (avec le modèle du domaine), sans oublier notre problématique principale qui est l'évaluation des apprenants, individuelle et collective. L'ajout d'un modèle du domaine, d'un modèle de l'apprenant ainsi qu'un modèle pédagogique à un jeu sérieux permet de renforcer l'aspect pédagogique (suivi et soutien de l'apprenant) tout en conservant l'aspect ludique et immersif d'un jeu sérieux (mécanisme de jeu, environnement 3D, réalité virtuelle...). De cette façon, le jeu sérieux joue le rôle d'environnement pédagogique et d'interface utilisateur d'un STI.

Une évaluation globale et collective peut être effectuée par un processus collaboratif (ou processus de négociation) entre les différents agents évaluateurs. Le processus d'évaluation est basé sur une transformation graduelle des données de jeu jusqu'à la production d'une évaluation des apprenants. Chaque étape est assurée par un agent dédié, de manière asynchrone (i.e. au fur et à mesure du déroulement du jeu). Il s'agit de conserver le caractère hétérogène de ces informations (connaissances, procédures, données brutes) en proposant des méthodes d'évaluation adaptées (techniques relevant de l'IA ou simple agrégation mathématique) ainsi que la variété de leur source. La spécialisation des agents ainsi que leurs interactions à un niveau de langage indépendant des composants logiciels "bas niveau", permet d'assurer une certaine généricité et ainsi de faciliter l'évolution du système en cas de modification de processus d'évaluation ou

d'ajout de nouveaux indicateurs et/ou sources d'information [Serment et al., 2008]. Il s'agit également ici de proposer une approche générique permettant à tout jeu sérieux existant d'évoluer vers un STI.

IV.3.1 Rôle du jeu sérieux

Le jeu sérieux s'occupe de l'interface utilisateur, du rendu 3D du monde virtuel (un ensemble de modèles 3D qui reconstitue une ville ou une commune, voitures, arbres, bâtiments, etc), de différents modules de simulation (propagation de feu, inondation, etc), ainsi que les modèles de données (Acteur, moyens et sinistres). Le jeu sérieux peut être basé sur une architecture client/serveur, c'est-à-dire un serveur de jeu (chargeant un certain scénario) auquel se connecte les joueurs (apprenants) à travers des clients via réseau ou internet pour gérer l'aspect collaboratif.

Par analogie avec les ITS, l'interface utilisateur et l'environnement pédagogique est représenté par l'interface 3D. L'environnement 3D joue un rôle très important car la 3D apporte un aspect immersif aux joueurs qui leur permet d'adopter leur rôle et de faire abstraction de l'outil utilisé (ordinateur, simulateur, ...). Élément central de l'environnement de formation, l'évolution vers un STI se construira ainsi par l'ajout de modules autour du jeu sérieux en vue de lui ajouter les capacités de simulation et d'évaluation qui peuvent lui manquer.

IV.3.2 Présentation générale de l'architecture multi-agents

L'aspect distribué et hétérogène du problème nous a conduit à opter pour une architecture multi-agents pour la mise en œuvre de la solution. Les SMA permettent une architecture logicielle décentralisée et distribuée basée sur des agents possédant des capacités sociales (communication et interactions entre les agents) [Wooldridge et al., 1995]. Ils peuvent également intervenir à un niveau d'intégration logicielle [Fipa, 2001] [Serment et al., 2008]. Notre objectif est de combiner les différentes technologies des jeux sérieux (environnement 3D, mécanisme de jeu, animation, ...) et un STI (représentation des connaissances, pédagogie, ...) pour obtenir un outil d'apprentissage optimal. Les agents composant le SMA ont pour mission de collecter les données liées aux apprenants, de traiter et d'évaluer ces données et de fournir un soutien aux apprenants.

La figure IV.1 décrit le fonctionnement général du système. Tout d'abord nous avons le jeu sérieux avec son modèle 3D, son interface utilisateur, un module de simulation, et enfin ses modèles de données (acteurs, moyens, sinistres dans notre cas d'application). Ces derniers peuvent être manipulés par le SMA en charge de simuler les comporte-

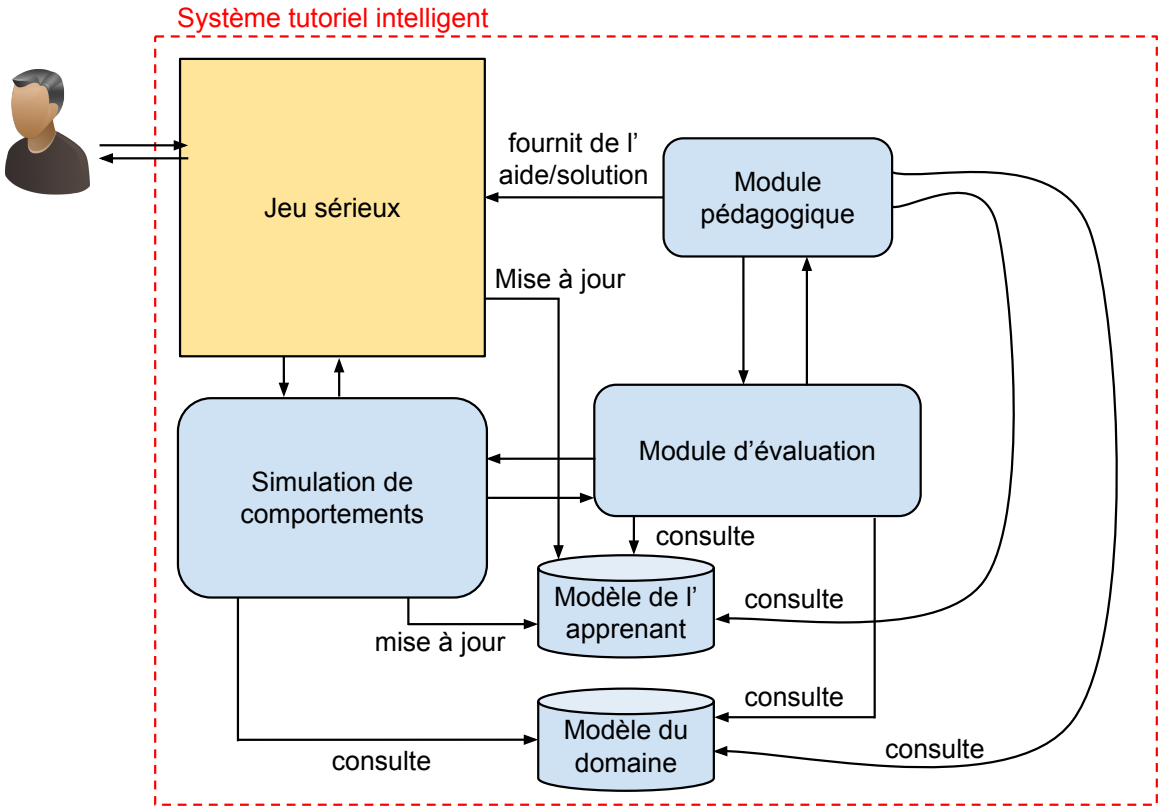


Figure IV.1 – Architecture générale.

ments humains (déplacement d'un acteur, intervention sur sinistre ...). L'objectif ici est de faire l'hypothèse la plus pessimiste, c'est à dire celle impliquant l'existence antérieure d'un Jeu Sérieux (JS) auquel on souhaite ajouter *a posteriori* des capacités de simulations et d'évaluation des apprenants. Cette hypothèse provient de notre contexte applicatif, mais il a semblé intéressant de la généraliser afin de pouvoir proposer une architecture logicielle générique qui sépare les considérations techniques des besoins de simulation et d'évaluation. En utilisant des agents soit comme médiateur entre le JS (ou ses composants internes) et les nouvelles fonctions proposées, soit comme moteur de ces fonctions (évaluation et simulation), l'architecture gagne en généricité et évolutivité mais permet également de mettre en œuvre des processus distribués et coopératifs.

Ainsi, pour chaque composant du système un agent source de données a pour mission de collecter les informations nécessaires pour le processus de monitoring. L'agent indicateur est chargé de calculer ou de sélectionner les indicateurs essentiels pour l'évaluation des apprenants, à partir des informations fournies par les agents sources de données. Le caractère technique de l'obtention de ces données est abstrait grâce aux agents sources de données. Les agents évaluateurs utilisent les indicateurs fournis par les agents indicateurs pour évaluer l'apprenant. Le résultat de l'évaluation est utilisé par la suite par l'agent pédagogique et peut servir de base pour un débriefing post-jeu. Les agents évaluateurs ont à leur disposition une partie du modèle du domaine

(partie correspondante à un rôle ou compétence à évaluer). A partir du modèle du domaine (partiel ou complet) et du modèle de l'apprenant, les agents évaluateurs peuvent comparer les actions des apprenants aux actions qui doivent être faites. Les agents évaluateurs transmettent par la suite le résultat de l'évaluation à l'agent pédagogique pour qu'il adopte la stratégie adéquate pour venir en aide à l'apprenant. Nous détaillerons chaque composant dans les sections suivantes.

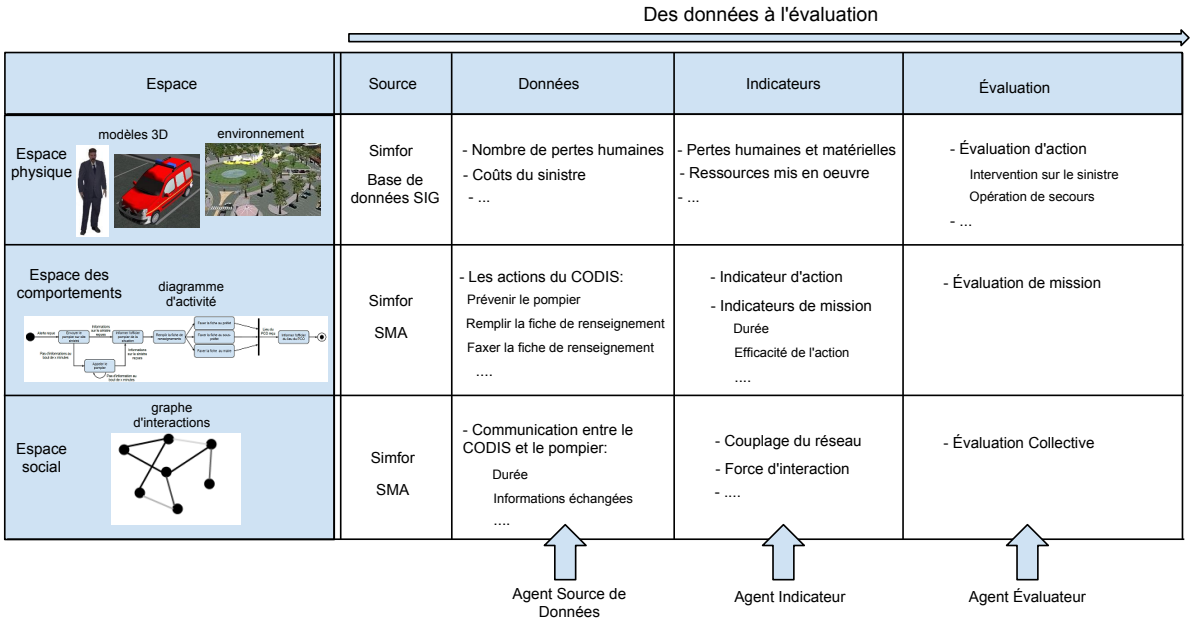


Figure IV.2 – Des données à l'évaluation

Pour mieux comprendre comment les agents distribuent le processus d'évaluation présenté dans chapitre III, la Figure IV.2 résume le processus de transformation graduelle des données de jeu vers l'évaluation des apprenants. Ces informations sont transformées et agrégées progressivement mais chaque étape est assurée par un agent dédié, chacun mettant en oeuvre des mécanismes spécifiques adaptés à la nature de l'information manipulée. Cette figure illustre le caractère hétérogène de ces informations ainsi que leur source, et comment les agents arrivent à combler le vide entre les connaissances et les traces au niveau logiciel.

Les sections suivantes détaillent chaque module notamment en montrant le rôle des agents ces modules.

IV.4 Module d'évaluation

Le module d'évaluation vise à fournir, à l'agent pédagogique, une évaluation des apprenants en temps réel. Ce module se compose d'agents source de données, d'agents in-

dicateurs, et enfin d'agents évaluateurs. La figure IV.3 décrit le fonctionnement général du module d'évaluation. Les sections suivantes détaillent la composition et le fonctionnement de ces différents agents.

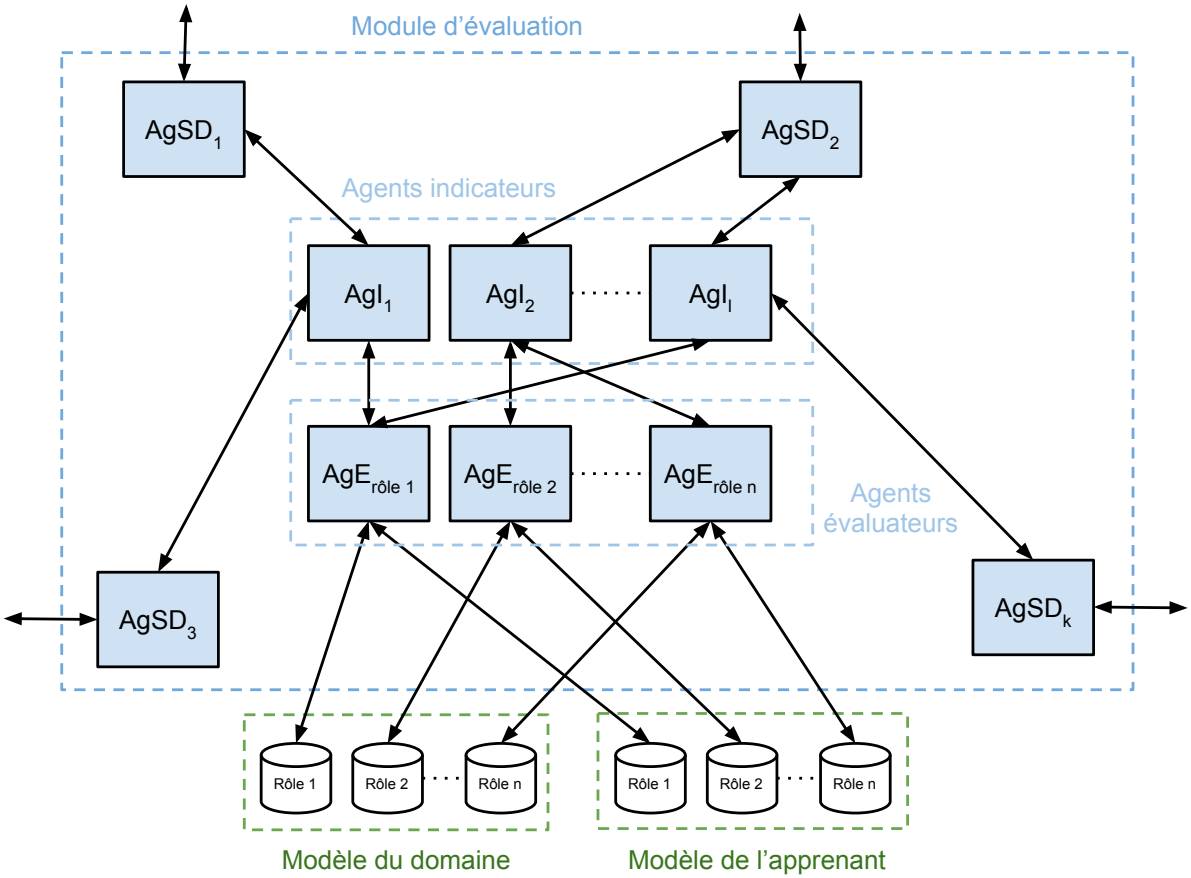


Figure IV.3 – Les composants du module d'évaluation.

IV.4.1 Agent Source de Données

Pour chaque composante génératrice d'information (un système de gestion de base de données par exemple), un *agent source de données* (AgSD) est associé, dont la mission est de collecter les informations nécessaires pour le processus de suivi et d'évaluation des apprenants. Ces informations peuvent être liées au jeu sérieux, consistant en des données de simulation comme le temps ou l'évolution d'un sinistre. Ces informations peuvent aussi être liées à la base de données géographiques décrivant l'emplacement des bouches à incendie, le nombre d'habitants dans un bâtiment etc. Ces informations peuvent enfin être liées aux comportements des agents (MAS responsables de la simulation des PNJ). Par exemple, l'état d'un agent (déplacement, communication, etc.) ou de l'action en cours d'exécution. L'agent source de données doit récupérer les données provenant des différents *Espace d'évaluation* (figure IV.2).

Comportements

L'AgSD reçoit en permanence des données à partir des composants auxquels il est lié (le jeu sérieux, base de données SIG, simulation de comportement). Selon les données reçues, ces dernières sont mises à jour si elles existent déjà, comme la position d'un acteur, ou bien ajoutées comme nouvelle entrée (par exemple, une action effectuée par un acteur). Le AgSD a les comportements suivants :

- *Diffuser des données* : Un agent ou un service peut s'abonner à l'AgSD pour suivre une certaine donnée. Lorsque les données en question sont mises à jour, l'AgSD diffuse l'information aux abonnés concernés.
- *Répondre à une requête* : L'AgSD peut recevoir une requête pour des données spécifiques, si L'AgSD possède les données en question, l'AgSD envoie les informations à l'agent ou au service demandeur.

Connaissances

L'AgSD manipule les connaissances suivantes :

- *Une liste de données* : Les données sont représentées par une structure de données en fonction du composant qui génère la donnée. Nous avons défini trois types de données : les données relatives aux acteurs, les données relatives aux actions et aux connaissances, et enfin les données d'information générale (données de simulation).
- *Liste d'abonnés* : Cette liste contient l'identifiant des agents (ou service) qui souscrivent à l'AgSD pour suivre certaines données.

IV.4.2 Agent Indicateur

L'agent Indicateur (AgI) doit calculer ou sélectionner l'information pertinente pour l'évaluation des apprenants.

Comportements

L'AgI possède les comportements suivants :

- *Calcul d'indicateurs* : selon le type d'évaluation (en fonction de la requête de l'agent évaluateur) et des indicateurs disponibles, l'agent indicateur sélectionne les indicateurs appropriés, et les envoie à l'agent évaluateur. Comme spécifié dans l'Espace d'évaluation, chaque indicateur requiert des modes de calculs spécifiques : ainsi chaque agent peut présenter un comportement interne différent.

– *Demande de données* : en fonction de l'indicateur retenu, l'AgI envoi une requête à l'AgSD pour récupérer les données nécessaires au calcul de l'indicateur.

Connaissances

L'AgI manipule les connaissances suivantes :
– *Indicateur* : un indicateur est utilisé pour calculer une évaluation. Cette évaluation peut porter sur une action, une mission ou d'une évaluation collective. Les indicateurs et leur pertinence sont définis par l'expert du domaine. Les indicateurs sont étendus avec des attributs spécifiques pour décrire les actions spécialisées ou de missions spécifiques.
– *Liste de données* : cette liste de données (la même structure utilisée par l'agent source de données) est utilisée par l'agent indicateur pour générer des indicateurs.

IV.4.3 Agent évaluateur

L'agent évaluateur (AgE) a pour mission de fournir une évaluation individuelle en temps réel ainsi qu'une évaluation finale à la fin de la partie. L'évaluation collective est calculée par un agent évaluateur global qui agit à titre d'arbitre, en utilisant la connaissance globale et partagée de tous les agents évaluateurs.

Comportements

L'AgE possède les comportements suivants :
– *Évaluation temps réel* : l'AgE tient à jour l'évaluation de l'apprenant tout au long du scénario de jeu, cette évaluation va permettre à l'agent pédagogique de proposer de l'aide et un support en temps réel à l'apprenant.
– *Évaluation finale* : l'évaluation finale de l'apprenant synthétise la valeur finale de l'évaluation temps réel combinée avec des informations sur le résultat final de l'exercice (mesure des dégâts par ex.). Cette évaluation représente une évaluation sommative qui certifie (ou non) l'apprentissage de l'apprenant (compétences apprises et compétences qui restent à apprendre).
– Demande d'indicateur : en fonction du type d'évaluation (action, mission, évaluation en temps réel, finale ...), l'agent évaluateur envoie une requête à l'agent indicateur pour obtenir les indicateurs nécessaires pour l'évaluation.
– *Répondre à une demande d'évaluation* : lorsque l'AgE reçoit une demande d'évaluation, l'AgE calcule l'évaluation (selon la requête, évaluation temps réel, final, ...) et envoie le résultat au demandeur.

Connaissances

L'AgE manipule les connaissances suivantes :

- *Le modèle du domaine* : le modèle du domaine représente les différents concepts du domaine étudié (voir figure IV.5). Celui-ci est segmenté en plusieurs parties où chaque partie représente un rôle ou les compétences à apprendre. Le modèle du domaine peut être codé sous la forme d'une ontologie.
- Le modèle de l'apprenant : pour chaque apprenant ou AgJ (acteur simulé), un modèle de l'apprenant est associé. Le modèle de l'apprenant représente l'état mental d'un acteur à l'instant *t*. Comme le modèle du domaine, le modèle de l'apprenant peut également être codé sous forme d'une ontologie.
- *Les indicateurs* : les indicateurs (la même structure utilisée par l'AgI) sont utilisés par L'AgE pour calculer l'évaluation.
- *L'évaluation* : l'évaluation à l'instant *t* est représentée par le score global ainsi que les six autres scores utilisés pour calculer le score global. Cette approche multi-critères permet d'englober les différentes facettes de l'évaluation des compétences.

IV.5 Module de simulation des comportements

Le module de simulation de comportements simule les comportements humains pour remplacer les joueurs absents par des joueurs "artificiels" (que nous appelons agent de jeu) communément appelés Personnages Non Joueurs (PNJ) dans le jargon des jeux vidéo. Pour simuler les PNJs, nous avons opté pour une architecture BDI (Belief, Desire, Intention) [Rao et al., 1995] permettant la simulation de comportement décisionnels complexes. Notre objectif est de fournir une architecture d'agent facilitant la conception des PNJs pour un scénario spécifique par l'expert du domaine. Ces PNJs peuvent avoir :

- des comportements nominaux (effectuer les comportements attendus),
- ou des comportements intentionnellement erronés afin de vérifier que le ou joueurs détectent et prennent en compte les erreurs d'un PNJ.

Les PNJ doivent aussi adapter leurs comportements en fonction :

- *des actions et interactions* (compétences sociales) des autres acteurs (apprenants),
- *des événements liés à l'environnement* (monde virtuel) qui se produisent au cours de la partie.

IV.5.1 Modélisation des agents de jeu

Le modèle proposé est composé d'un ensemble d'agents, d'actions et de faits. Chaque scénario de jeu est associé à plusieurs modèles d'agents qui reflètent les comportements des PNJs. Un modèle d'agent est représenté comme suit :

$$Modèle(rôle) = \{Buts, Plans, Faits, Dialogues\} \quad \text{(IV.1)}$$

Un agent du jeu (AgJ), représenté par un agent BDI, joue un rôle dans un scénario, et en tant que tel tente de réaliser les objectifs (qui sont soit activés soit évalués comme accessibles dans le contexte). Pour cela, il adopte des plans établis en fonction de ses connaissances (définies dans une liste de faits déclaratifs).

Chaque plan est composé d'actions, dirigées soit vers l'environnement soit vers d'autres agents/acteurs (provoquant un autre type d'effets sur le scénario). Les interactions entre un AgJ et un acteur humain (mais aussi les autres AgJ) sont représentées par des dialogues adaptatifs basés sur un échange de phrases. Ces phrases correspondent à des propositions faites à l'apprenant, ce dernier doit choisir la réponse appropriée. De son coté, le AgJ choisit une réponse selon sa base de fait ainsi que les pré-conditions liées à chaque réponse. La section IV.5.3 détaille la notion de dialogue pour les AgJ. La figure IV.4 synthétise la structure générale du modèle BDI avec un méta-modèle UML.

Sur le plan de l'architecture logicielle, un moteur d'agent est défini pour animer les agents (i.e. exécuter le cycle de vie des agents). Un AgJ commence à mettre à jour sa base de faits au travers :

- de sa perception de l'environnement (événements),
- des messages des autres agents (ou joueurs).

L'agent sélectionne :

- En premier lieu le but approprié en fonction de la situation (rôle, base de fait, etc.).
- En second lieu l'intention (le plan) qui permettra d'atteindre ce but.

puis il exécute le plan.

Pour sélectionner un but, l'AgJ commence par évaluer la faisabilité des buts (plans pour chaque plan). Une fois les buts réalisables connus, l'AgJ sélectionne le but de plus forte priorité. S'il y a plusieurs buts de même priorité, l'AgJ sélectionne l'un d'entre eux au hasard. Ce processus permet à un agent d'adapter son comportement à l'état de l'environnement au sens large (ie informations sur le monde virtuel et les autres acteurs incarnés par des agents ou non). L'objectif de ce modèle est de fournir un comportement adaptatif des PNJs mais implique aussi de fournir au concepteur (de scénario) un outil d'édition graphique convivial et efficace pour faciliter la conception de scénario.

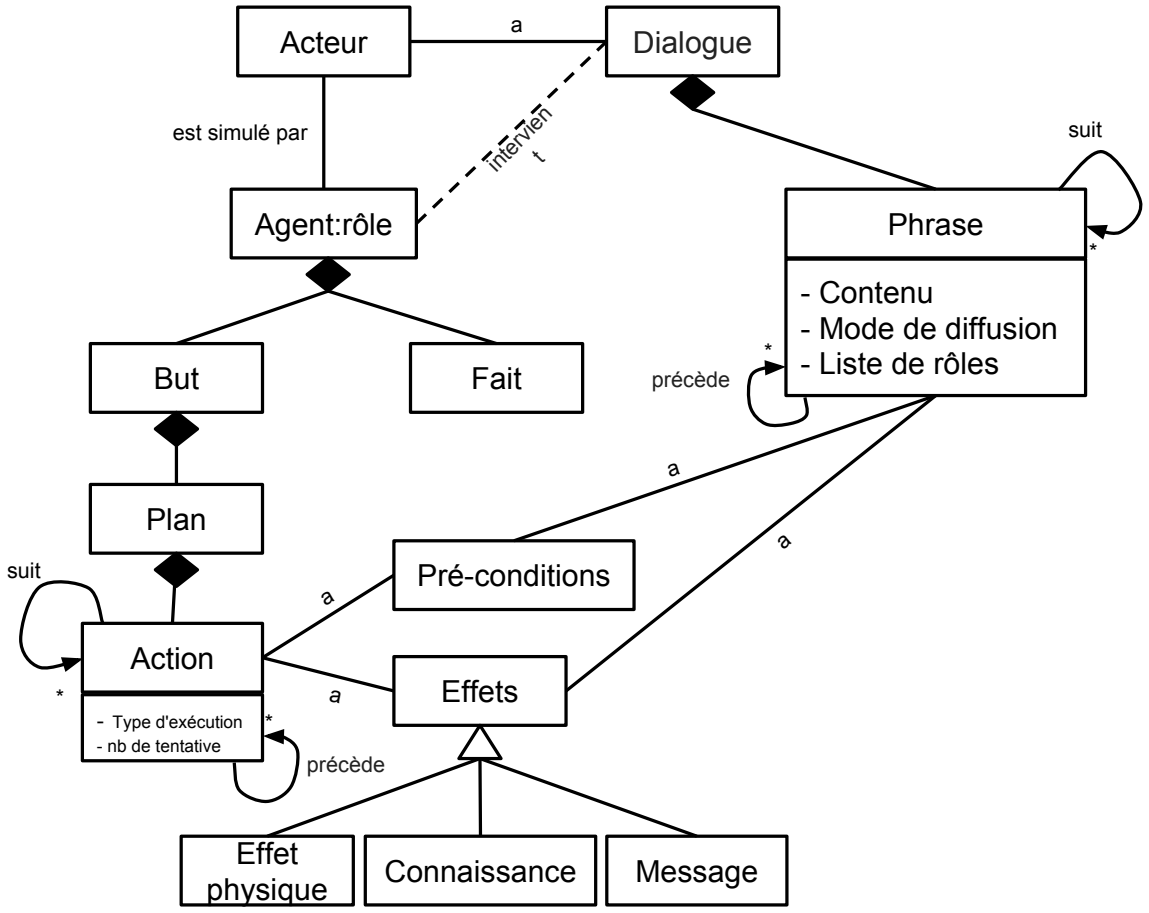


Figure IV.4 – Représentation UML du modèle BDI.

IV.5.2 Modélisation des actions

Une action caractérise ce qu'un acteur peut faire lors d'une situation de gestion de crise. Ces actions servent ainsi de modèle pour concevoir les comportements des agents ainsi que les données d'évaluation lorsque l'on compare ce qui est fait à ce qui aurait dû être fait.

En ce qui concerne l'agent de jeu, une action est caractérisée par :
- *des pré-conditions* (exprimées par un ensemble de faits censés être présents dans la base de faits de l'agent),
- *des effets*.

Une action peut être exécutée de plusieurs façons :
- une seule fois, lors de son déclenchement,
- cycliquement (action répétitive comme "vérifier la progression du feu"),
- de manière programmée (s'exécute à l'instant t prédéfini) .

Pour chaque action, le concepteur peut spécifier un nombre de tentatives au-delà

duquel, en l'absence de succès, l'agent passe à l'action suivante. Les actions influencent l'environnement à travers trois types d'effets :

- *des effets Physiques* (PE) : un tel effet influence l'environnement 3D du jeu sérieux (comme le ferait un acteur humain au travers de son interface utilisateur) ;
- *des effets Connaissances* (KE) : les conséquences sont la modification des faits qui peut annuler ou valider un but, annuler ou valider des actions préalables ;
- *des effets Messages* (ME) : un tel effet reflète la nature sociale de l'agent et des acteurs dans le jeu sérieux.

Tableau IV.1 – Les différentes actions ainsi que leurs effets définis dans le cadre de la formation à la gestion de crise.

Action	Type d'effet	Description
Téléphoner	KE, ME	Cette action permet de joindre un autre joueur par téléphone. Si l'interlocuteur est un joueur humain, la communication se fait par voix oral via VOIP (Voix sur IP). Si l'interlocuteur est un AgJ, la communication se fait par un dialogue textuel via un échange de phrase (voir section IV.5.3).
Fax	KE, ME	Cette action permet d'envoi un fax à un ou plusieurs destinataires. Les fax sont représentés par des documents HTML pré-formatés correspondants aux documents officiels de la gestion de crise (à compléter avec les bonnes informations).
Radio	KE, ME	Cette action permet de joindre un(ou plusieurs) joueur(s) par radio. Le joueur sélectionne un canal puis appui sur le bouton parler pour s'adresser à l'ensemble des joueurs qui sont sur le même canal.
Parler	KE, ME	Cette action permet de parler avec les joueurs qui sont à proximité (défini avec un certain périmètre).
Se déplacer	PE	Cette action permet aux joueurs de se déplacer dans l'environnement virtuel. Le joueur peut se déplacer avec la souris ou bien de manière automatique en sélectionnant une adresse dans son carnet d'adresses (à pied, en utilisant un véhicule...).
Main courante	KE	La main courante simule un portail web que les différents intervenants (de la gestion de crise) utilisent pour relater les informations sur le sinistre. Le joueur écrit l'information sur la main courante qui pourra être consultée par l'ensemble des joueurs qui ont accès à l'outil *"main courante"*.

Le tableau IV.1 illustre les différentes actions courantes dans la gestion de crise ainsi que leurs effets dans le jeu sérieux. Certaines actions ne peuvent être menées que par les AgJ (PNJ). Ces actions contribuent à enrichir la simulation et de la rendre plus réaliste. Par exemple, un assistant du maire (PNJ) qui doit préparer une salle pour une conférence de presse. Cette action prendra un certain temps au cours duquel le maire (humain) ne peut pas commencer sa conférence de presse. Ce genre d'actions peut aussi déclencher un changement physique dans l'environnement 3D, par exemple activer une sirène d'urgence, mettre en place un tapis de mousse (action pompier). Ces actions virtuelles sont définies par le concepteur du scénario détaillant pour chacune :

- le temps d'exécution,
- les pré-conditions,
- les effets.

IV.5.3 Modélisation des dialogues

Il y a plusieurs façons d'interagir dans un jeu sérieux pour la formation à la gestion de crise (téléphone, fax, radio ...), mais si l'un des interlocuteurs est un AgJ, la communication se fait sous la forme de dialogue textuel. L'interaction automatisée partielle peut entraîner une interaction fixe qui manque de souplesse pour tenir compte des différentes situations auxquelles font face les différents acteurs (ou AgJ). Afin d'éviter cela, nous avons défini un processus de dialogue basé sur un ensemble de phrases potentielles en fonction du contexte relatif à la perception des agents de leur environnement. Un dialogue est conçu comme un arbre où chaque nœud est représenté par une phrase. Une phrase est caractérisée par la liste des propriétés décrites dans le tableau IV.2.

Tableau IV.2 – Structure d'une phrase dans un arbre de dialogue.

Attributs de la phrase	Description
Id	identifie la phrase dans l'arbre de dialogue
Parents	liste de (possibles) phrase parentes (phrases précédente)
Enfants	Liste de phrases (possibles) suivantes
Contenu	représente le texte affiché dans la boite de dialogue
Pré-conditions	contient les pré-conditions (ensemble de faits) requis pour que le AgJ puisse répondre cette phrase.
Informations transmises	représente un ensemble de faits transmis quand la phrase est choisie (par un AgJ ou un joueur humain)
Mode d'affichage	précise si la phrase va être affichée a : i) l'ensemble des rôles du scénario, ii) seulement les rôles dans la liste rôles ou bien iii) tous les rôles exceptés les rôles dans la liste de rôle
Liste de rôle	liste des rôles qui peuvent interagir avec cette phrase, utilisé pour l'affichage (Mode d'affichage)

IV.6 Agent de l'apprenant

Pour chaque apprenant ou agent, un modèle de l'apprenant est associé. Ce modèle représente l'état mental des acteurs à l'instant t. L'agent de l'apprenant (AgA) recueille les actions et des connaissances apprenants, et les stocke dans le modèle de l'apprenant. Le modèle de l'apprenant et l'AgA représentent le module de l'apprenant d'un STI.

Comportements

L'AgA possède les comportement suivants :

– *S'abonner à l'AgSD* : l'AgA s'abonne à l'AgSD pour recevoir les nouvelles actions et connaissances de l'apprenant.

– *Mise à jour du modèle de l'apprenant* : quand l'AgA reçoit des données de l'AgSD (nouvelles actions ou connaissances), l'AgA met à jour le modèle de l'apprenant correspondant.

Connaissances

L'AgA utilise les connaissances suivantes :
– *Le modèle de l'apprenant* : la structure de données du modèle de l'apprenant est la même que celle du modèle du domaine, afin de pouvoir utiliser la superposition des connaissances (*overlay model*) par l'agent d'évaluateur.
– Données brutes : l'AgA reçoit les actions et les connaissances de l'apprenant par L'AgSD.

IV.7 Agent pédagogique

L'agent pédagogique (AgP) joue le rôle d'un tuteur virtuel qui suit l'apprenant dans sa formation. Le AgP fournit un support et de l'aide à l'apprenant ce qui permet d'optimiser l'apprentissage dans l'environnement virtuel. Le AgP analyse la situation (résultats de l'évaluation, modèle du domaine, modèle de l'apprenant) et sélectionne la stratégie appropriée (propose une action à exécuter, affiche l'aide, correction, ...).

Comportements

L'AgP possède les comportement suivants :
– *Demande d'évaluation* : lorsqu'il y a un nouvel événement (action ou modification de l'environnement), L'AgP demande une évaluation en temps réel à l'AgE.
– *Sélection d'une stratégie de support* : une fois l'évaluation reçue, l'AgP doit sélectionner une stratégie. Cette stratégie dépend du résultat de l'évaluation ainsi que du niveau de l'apprenant et du niveau de difficulté. Nous avons défini quatre types de stratégie :
  – *Laisser faire l'apprenant* : les études menées par [Schmidt et Bjork, 1992] ont montré que les apprenants ayant reçu une assistance non immédiate retiennent plus longtemps les compétences apprises. Si l'apprenant n'est pas novice, l'agent pédagogique laisse l'apprenant trouver la solution par lui-même.
  – *Donner un indice* : si l'apprenant est novice, l'agent pédagogique commence par donner des indices sur la procédure à suivre.

– *Proposer une action* : si l'apprenant est bloqué (délai dépassé) l'agent pédagogique propose l'action à réaliser (dans le cas du CODIS "Téléphoner aux pompiers").
– *Faire l'action à la place de l'apprenant* : si l'apprenant ne sait pas comme faire, l'agent pédagogique fait l'action à la place de l'apprenant tout en lui expliquant ce qu'il fallait faire.

Connaissances

L'AgP utilise les connaissances suivantes :
– *Le modèle de l'apprenant* : pour fournir un support, l'AgP a besoin d'accéder au modèle de l'apprenant. Le modèle de l'apprenant fournit l'historique des actions ainsi que les connaissances de l'apprenant.
– *Le modèle du domaine* : le modèle du domaine fournit à l'AgP les procédures à apprendre de l'apprenant, ce qui permet à l'AgP de déterminer la prochaine action à réaliser par l'apprenant.
– *Résultat d'évaluation* : le résultat de l'évaluation permet d'identifier les lacunes des apprenants à travers les six scores (indicateurs).

IV.8 Représentation des connaissances

Comme nous l'avons vu dans les sections I.3.2.2 et I.3.2.3, la présentation des connaissances touche principalement les connaissances liées au domaine étudié et à la modélisation de l'apprenant, représentées respectivement dans notre architecture par le modèle du domaine et le modèle de l'apprenant.

IV.8.1 Modèle du domaine

Le modèle du domaine représente les différents concepts du domaine étudié en l'occurrence ici, la gestion de crise. Le modèle du domaine est segmenté en plusieurs parties représentant un rôle ou une compétence à apprendre. Le modèle de domaine est représenté par une ontologie qui décrit les différents concepts de gestion crise. Actuellement, l'ontologie est limitée aux actions de base de la gestion de crise, mais peut être étendue pour répondre à toute demande d'un expert du domaine. L'ontologie a été modélisée par l'outil de création d'ontologie "*Protégé*" [Noy et al., 2003], cette ontologie est décrite par la figure IV.5. Le modèle du domaine représente le module expert d'un ITS.

Comme nous l'avons vu dans la section II.3, le modèle du domaine vise essentielle-

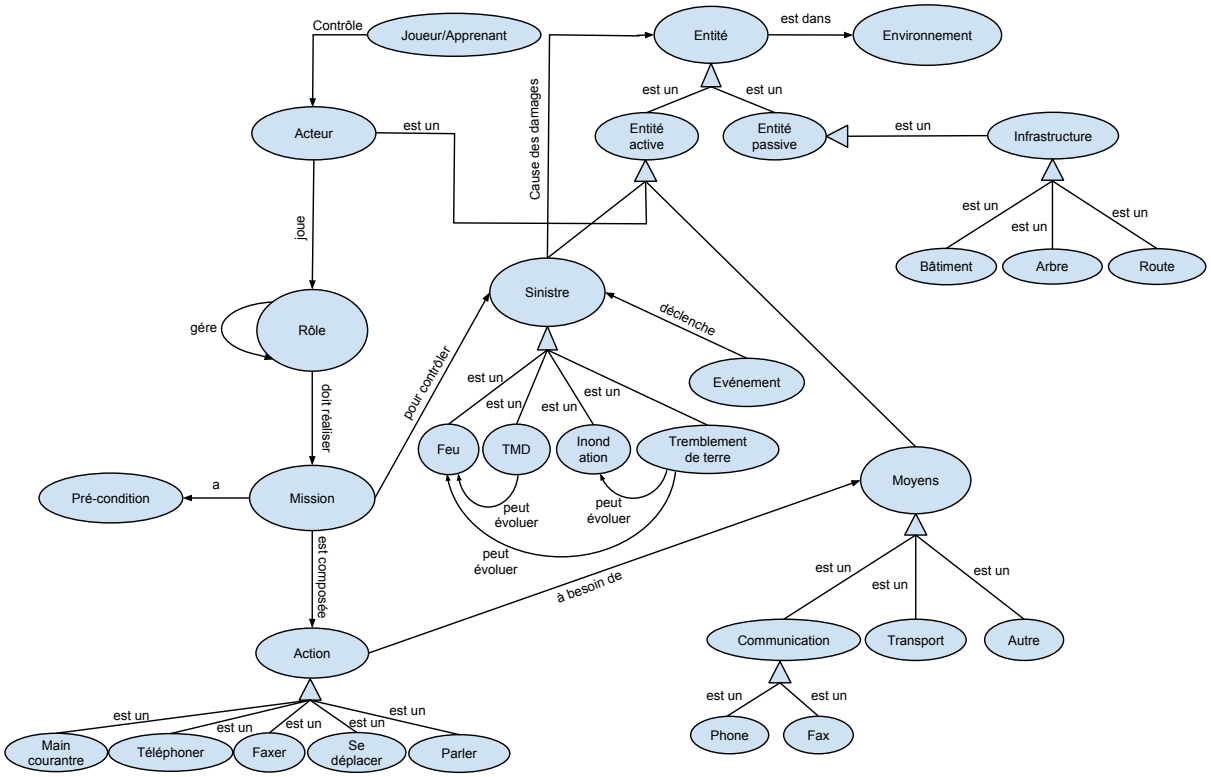


Figure IV.5 – Ontologie utilisée pour représenter le modèle du domaine pour la formation à la gestion de crise.

ment à représenter le domaine étudié (la gestion de crise). Les procédures de la gestion de crise sont représentées par l'espace comportemental dont la représentation des connaissances est constituée d'un ensemble de missions à réaliser pour chaque rôle du scénario, la figure IV.6 décrit la structure du modèle du domaine.

IV.8.2 Modèle de l'apprenant

Pour chaque apprenant, un modèle de l'apprenant est associé. Ce modèle représente l'état mental de l'apprenant à l'instant t. Le modèle de l'apprenant ainsi que l'agent de l'apprenant représentent le module de l'apprenant d'un STI. La structure de données du modèle de l'apprenant possède la même structure que celle du modèle du domaine, représenté par une ontologie, afin de pouvoir utiliser la superposition des connaissances [VanLehn, 1988] (*overlay method*), où la connaissance de l'apprenant est représentée comme un sous-ensemble du modèle du domaine (la superposition peut être utilisée par l'agent évaluateur afin de comparer ce qui a été fait avec ce qui doit être fait).

Comme pour le modèle du domaine, le modèle de l'apprenant est représenté dans l'espace comportemental et est constitué des actions réalisées par l'apprenant. La figure IV.7 illustre la structure du modèle de l'apprenant.

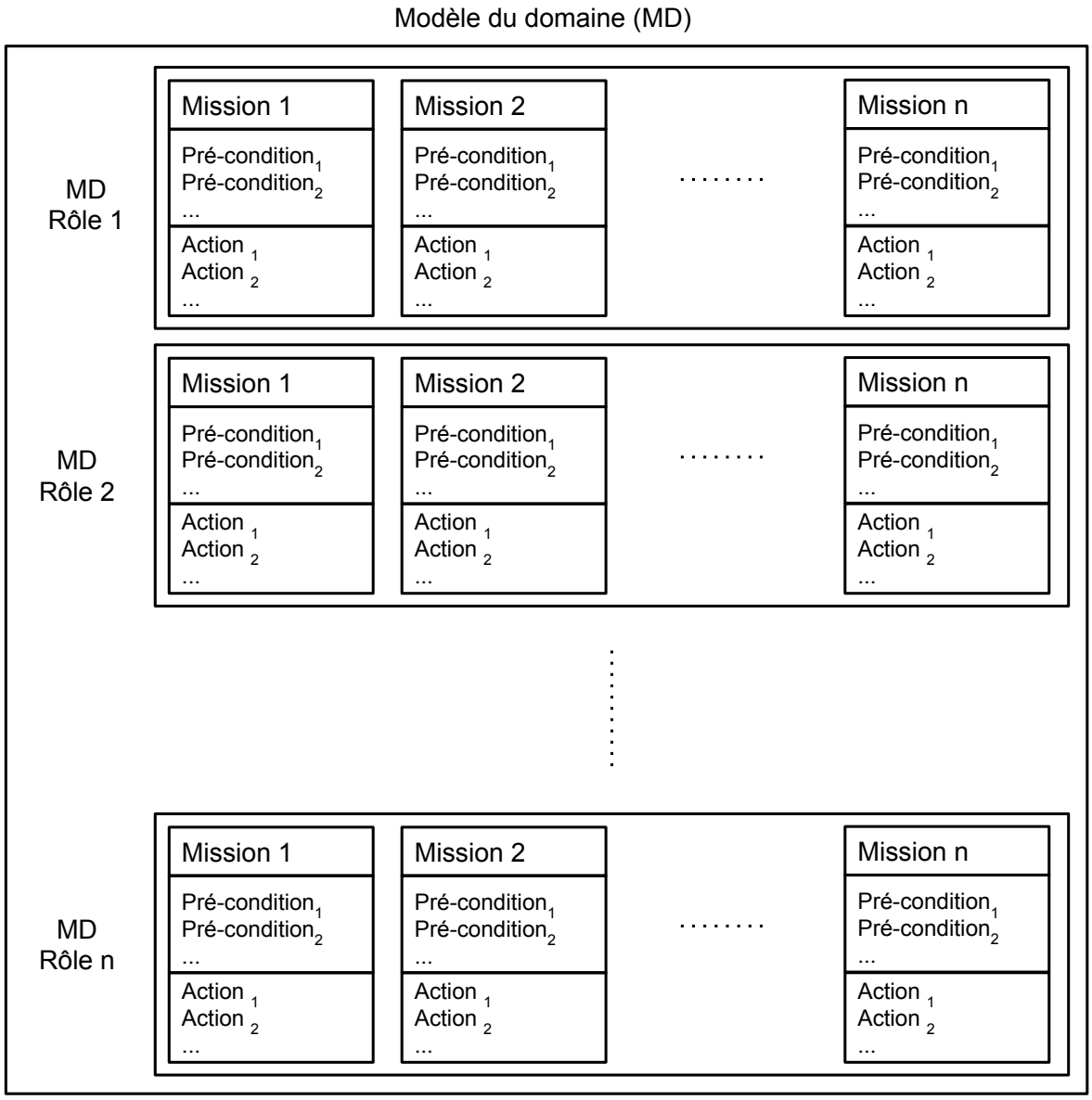


Figure IV.6 – Structure du modèle du domaine.

IV.9 Interaction entre les différents agents

La figure IV.8 décrit, avec un diagramme de séquence UML, l'interaction des différents agents au cours du processus d'évaluation, et notamment lors de la phase de collecte de données. Ce diagramme présente le déroulement du suivi de deux apprenants : le CODIS et le Sous-préfet. L'Agent Apprenant (AgA) a pour mission de mettre à jour le modèle d'un apprenant (un agent apprenant est associé à chaque acteur), pour cela l'AgA s'abonne à un Agent Source de Données (AgSD) pour obtenir les actions et les informations/connaissances de l'apprenant. Lorsque l'apprenant effectue une action, l'AgSD envoie les données de l'action à l'Agent Apprenant (id de l'acteur, cible, ...), signalé par les flèches 1 et 2 dans la figure IV.8. Une fois les données reçues, l'Agent

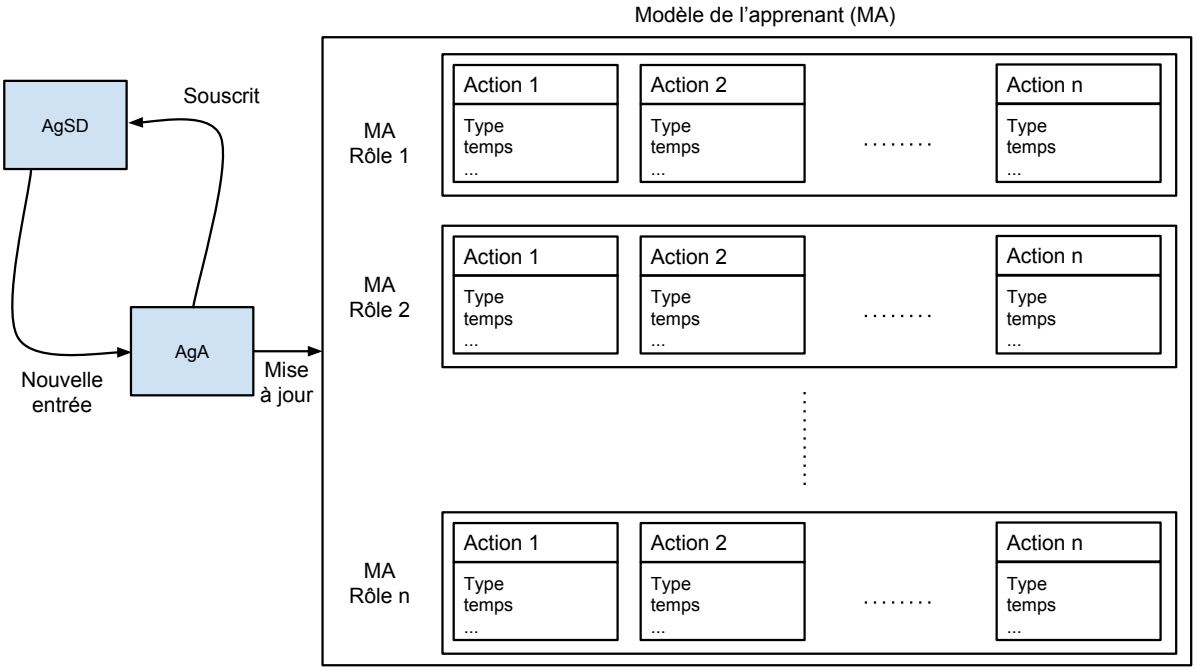


Figure IV.7 – Structure du modèle de l'apprenant.

Apprenant met à jour le modèle de l'apprenant et informe l'Agent Pédagogique (AgP) d'un nouvel événement (nouvelle action effectuée, flèches 3 et 4).

Une fois l'événement transmis, l'AgP procède à l'accompagnement de l'apprenant (en le guidant dans l'exécution de ses actions par exemple). Tout d'abord, l'AgP demande une évaluation de l'apprenant (flèches 5 et 6, pour chaque apprenant, un agent pédagogique est associé). L'Agent évaluateur (AgE) récupère les informations pertinentes du modèle de l'apprenant (niveau, rôle, procédure en cours de réalisation, ...), il fait ensuite une requête à l'Agent Indicateur (AgI) pour obtenir la valeur des indicateurs concernant l'évaluation. Dans le cas du sous-préfet, l'agent évaluateur procède à l'évaluation d'une action de déplacement et effectue une requête à l'Agent Indicateur (AgI) pour récupérer l'indicateur adéquat pour l'action de déplacement. Pour l'action de déplacement, L'AgI fait une requête auprès de l'AgSD relié au jeu sérieux (flèche 8) pour récupérer le temps de parcours ainsi qu'à l'AgSD relié à la base de données géographique (flèche 9) pour localiser l'adresse de destination de l'apprenant.

Une fois l'indicateur constitué (ActionMoveIndicateur), l'AgI envoie ce dernier pour que l'AgE calcule l'évaluation. Du côté de l'agent évaluateur en charge du Codis, ce dernier doit procéder à l'évaluation d'une mission. Pour cela l'AgE fait une requête à l'AgI de missions (flèche 7.2). L'AgI doit interroger le modèle de l'apprenant ainsi que le modèle du domaine pour constituer toutes les données nécessaires pour l'évaluation (actions, temps, etc.). Une fois l'indicateur calculé, ce dernier est envoyé à l'AgE pour produire l'évaluation demandée (flèche 11.2).

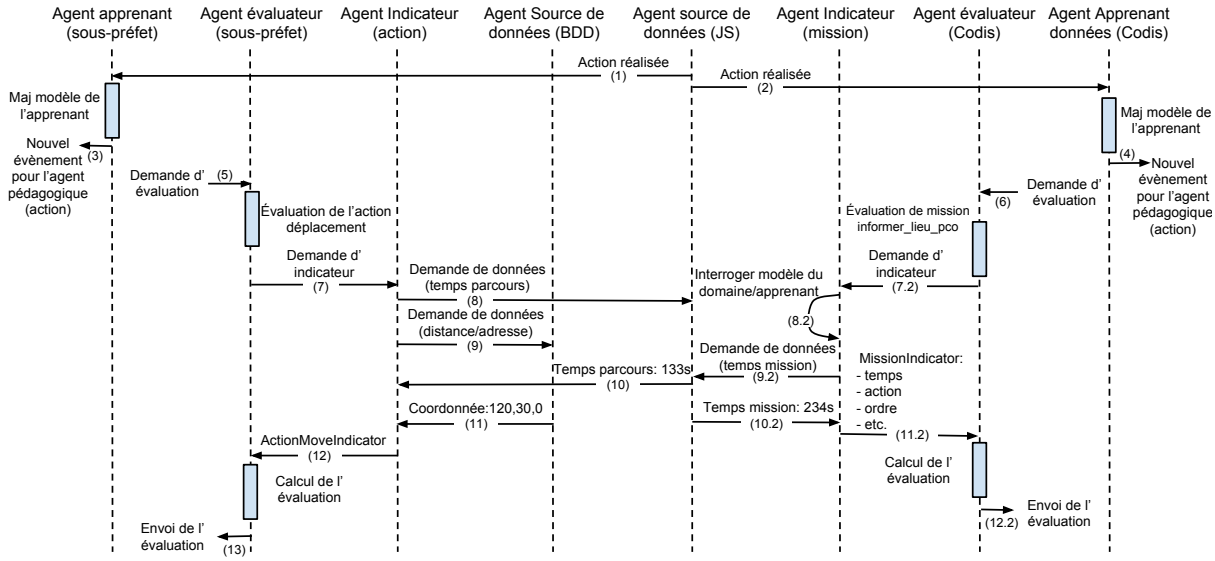


Figure IV.8 – Diagramme de séquence UML décrivant les interactions entre les différents agents

IV.10 Conclusion

Dans ce chapitre nous avons présenté l'architecture multi-agents pour la mise en œuvre de l'évaluation individuelle et collective des apprenants dans les jeux sérieux, et comment nous avons intégré les différents modules d'un STI avec notre cadre méthodologique : la notion d'espace d'évaluation. Les SMA permettent d'exploiter des propriétés distribuées et sociales afin de répondre à la problématique de la distribution et de l'hétérogénéité des sources d'informations.

Dans le chapitre suivant nous présentons l'implémentation de l'architecture logicielle multi-agents pour l'évaluation individuelle et collective des apprenants et son intégration dans un jeu sérieux pour la gestion de crise, à savoir le projet SIMFOR.

Troisième partie

Applications

CHAPITRE V

Implémentation

Sommaire

V.1 Introduction 109
V.2 Présentation du projet SIMFOR 110
V.3 Architecture de SIMFOR 110
V.4 Simulation de comportement 113
V.4.1 Éditeur d'agent 114
V.4.2 Éditeur de dialogue 115
V.5 Architecture 116
V.5.1 Serveur SIMFOR 116
V.5.2 Client SIMFOR 118
V.6 Suivi et interaction des agents 121
V.7 Évaluation 122
V.7.1 Éditeur du modèle du domaine 123
V.7.2 Moniteur d'évaluation 124
V.8 Les agents 126
V.9 Conclusion 131

V.1 Introduction

Dans ce chapitre nous présentons l'implémentation de l'architecture logicielle présentée dans le chapitre précédent et son intégration. Nous présentons tous d'abord le projet SIMFOR, un jeu sérieux pour la formation à la gestion de crise. Nous présentons ensuite les différentes applications d'édition (édition des AgJ, édition des dialogues, etc.) et de suivi (suivi des apprenants, suivi des AgJ) que nous avons développé dans le

cadre de notre approche. Enfin, nous présentons les différents agents intervenant dans la simulation de comportements et dans l'évaluation des apprenants (AgJ, AgSD, AgI, etc.)

V.2 Présentation du projet SIMFOR

Le projet SIMFOR (Figure V.1) est un jeu sérieux développé par la société SII [1] en partenariat avec la société Pixxim [2] en réponse à l'appel à projet "*Serious Gaming*" lancé par le secrétaire d'état Français chargé de la Prospective et du Développement de l'économie numérique. SIMFOR permet une approche ludique de l'apprentissage de la gestion de crise sous la forme d'un jeu sérieux. SIMFOR permet aux apprenants de se former à la gestion de crise en intégrant l'aspect multi-intervenant. L'objectif de SIMFOR est de proposer aux utilisateurs un contexte de gestion de crise en temps réel proche de la réalité en termes d'environnement, de scénarios auto-évolutifs et d'acteurs. Reposant initialement sur une évaluation humaine des compétences et des PNJ (Personnages Non Joueurs) aux comportements simplifiés (comportements linéaires et non adaptatifs), nous visons à faire évoluer SIMFOR en développant :

- la simulation de comportement humain des acteurs non joués ;
- l'accompagnement et l'évaluation des apprenants dans leur formation.

V.3 Architecture de SIMFOR

Le cœur de SIMFOR est basé sur trois composantes principales qui sont les *acteurs*, les *moyens* et les *sinistres*. Les *acteurs* représentent les avatars des apprenants dans le jeu, les *moyens* représentent les outils/objets mis à la disposition des joueurs dans le jeu (téléphone, fax, voiture, ...) et enfin les *sinistres* représentent le risque majeur que les joueurs devront gérer pendant la partie. Toutes ces entités doivent être spécifiées dans un fichier de configuration du scénario. Un scénario correspond à la mise en scène d'une situation de crise due à la découverte d'un risque majeur, avec la mobilisation et l'intervention d'acteurs et de moyens dans un ou plusieurs environnements dont l'objectif est de maîtriser le sinistre et de retourner à la situation normale. Les entités qui le composent, évoluent dans le temps par le biais d'actions et de comportements.

Le projet SIMFOR est développé en C++ Qt [3] en se basant sur le moteur de jeu Dela3D [4] (voir annexe A). La figure V.2 présente l'architecture générale de SIMFOR.

1. www.groupe-sii.com
2. www.pixxim.fr
3. http ://qt-project.org/
4. http ://www.delta3d.org/



Figure V.1 – Capture d'écran du projet SIMFOR

L'éditeur et le "*Game Player*" (environnement de jeu) utilisent les modules suivants :

- **Le module gestion des entités :** ce module permet à l'éditeur de gérer les entités de la base de données BDD entités qui contient l'ensemble des objets du monde virtuel SIMFOR susceptible d'évoluer soit suite à un événement de type crise soit en réaction à des actions d'acteurs (PNJ ou joueurs).
- **Le module gestion des scénarios :** ce module permet à l'éditeur de gérer la liste des scénarios de SIMFOR.
- **Le module création de scénarios :** ce module permet à l'éditeur de définir les différentes interactions entre les entités à l'aide des actions et des comportements, de définir les objectifs des acteurs joués pour mettre en évidence les objectifs en terme de jeu et de formation.
- **Le module animateur de scénarios :** le module Animateur de scénarios permet à l'éditeur et au *Game Player* la lecture et l'interprétation des données qui font vivre le scénario dans le temps.
- **Le module configurateur de scène 3D :** le module Configurateur de scène 3D permet, via l'éditeur de scénarios, la configuration des environnements intérieurs et extérieurs constituant le monde virtuel de SIMFOR.
- **Les modules d'intelligence artificielle :** ce module permet l'animation des personnages non joueurs PNJ. Module tres simplifié initialement, nous l'avons fait fortement évoluer à l'aide d'agents dits "délibératifs". Nous reviendrons plus en détails sur ce module dans la section simulation de comportements (section IV.5).

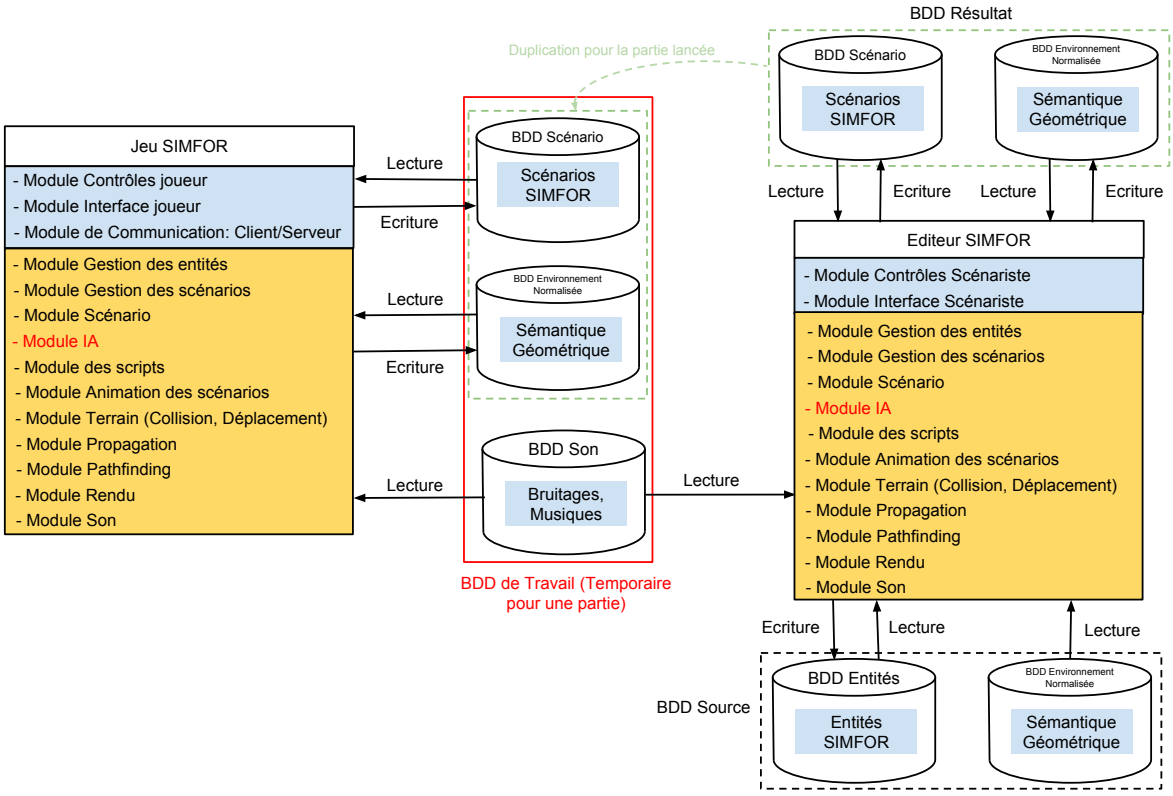


Figure V.2 – Architecture générale du projet SIMFOR.

- **Le module de simulation des sinistres :** ce module permet de gérer la diffusion du sinistre, suivant son type, dans l'environnement 3D.
- **Le module animation des figurants :** dans SIMFOR, de nombreuses entités (acteurs, moyens de transport, piétons, etc.) se déplacent dans un environnement 3D en évolution. Ce module permet de gérer leur déplacement dans un environnement donné, et de les animer en fonction de leur état.
- **Le module import de données géographiques :** ce module permet à l'éditeur d'importer des données SIG 3D pour les stocker en base de données relationnelle. Ces données seront ainsi lisibles par l'éditeur pour configurer la scène et le scénario.
- **Le module export de données géographiques :** ce module permet à l'éditeur d'exporter les données présentes dans la BDD environnement.
- **Le module stockage de données géographiques :** ce module permet d'architecturer et d'optimiser la BDD Environnement relationnelle, dans laquelle sont stockés les environnements de SIMFOR.
- **Le module requêtes et mises à jour de base de données 3D :** ce module permet d'accéder aux données de la BDD Environnement et de les mettre à jour si nécessaire.
- **Le module physique :** ce module permet de gérer les données physiques dans

l'environnement, et leur influence.

- **Le module rendu :** ce module permet la visualisation d'environnement et objet 3D. La principale fonction de ce visualisateur est d'afficher les différents éléments 3D composants le lieu (environnements géographiques et/ou intérieurs) où se déroule le scénario.
- **Le module sons :** ce module est responsable de l'ambiance sonore du jeu. Au cours du jeu, il n'y a pas de musique de fond, seuls les bruitages instaurent une ambiance sonore. Ces bruitages sont en rapport direct avec les actions des personnages de la scène (discutions entre joueurs, explosion, sirène, etc.).

Ces modules sont basés en grande partie sur moteur de jeu Delta3D, qui permet une parfaite homogénéité et transparence dans la conception de SIMFOR.

L'ajout de capacités d'évaluation (individuelle et collective) passe par l'architecture présentée dans le chapitre précédent, avec un *module d'évaluation* (AgSD, AgI, AgE), un *module de simulation de comportement*, un *module pédagogique*, les *modèles de représentation des connaissances* (modèle de l'apprenant, modèle du domaine),et enfin *le jeu sérieux* (SIMFOR dans notre cas). Cette architecture est décrite dans la figure V.3.

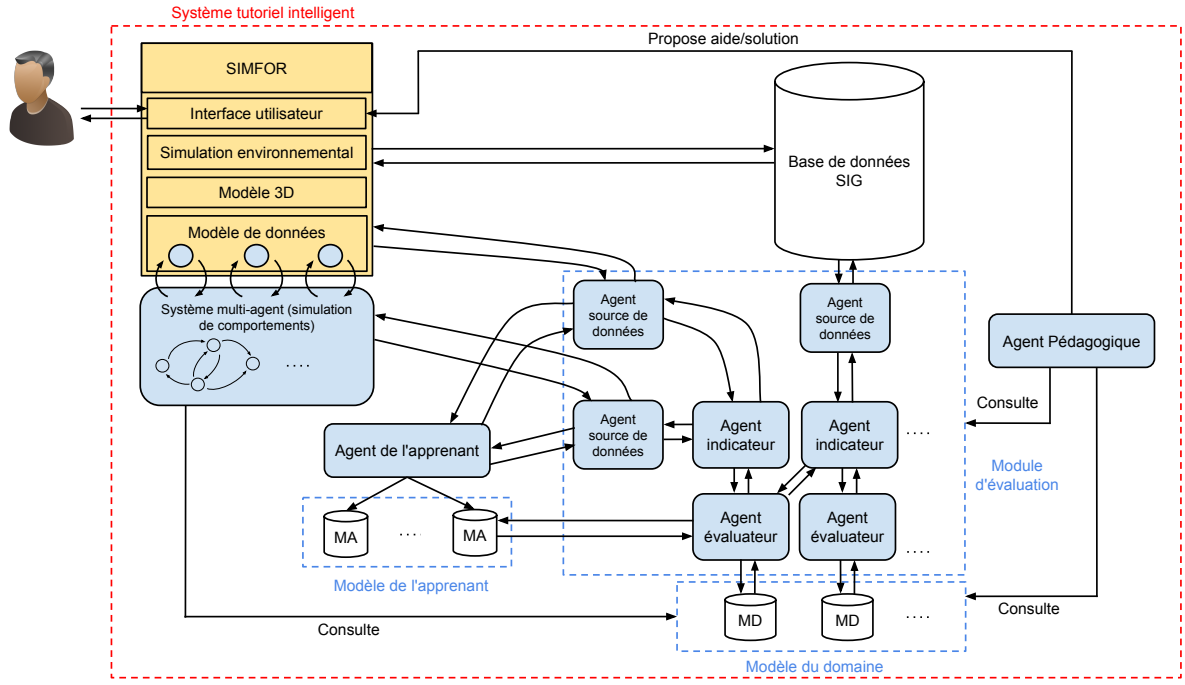


Figure V.3 – Architecture générale du système.

V.4 Simulation de comportement

Dans cette section, nous présentons les applications liées à l'édition des AgJ ainsi qu'aux dialogues. Ces applications sont représentées par l'éditeur d'agent et l'éditeur de

dialogue.

V.4.1 Éditeur d'agent

L'éditeur d'agent permet au concepteur de mettre en place les comportements des PNJ. Le scénario de SIMFOR spécifie les acteurs, les moyens, les sinistres ainsi que les événements associés (un sinistre de type incendie par exemple). Pour chacun des acteurs SIMFOR (jouant un rôle), le concepteur doit associer soit un PNJ (AgJ), soit un participant humain. A noter qu'en cours de partie, il est possible d'intégrer un apprenant (l'AgJ correspondant est désactivé), ou bien de remplacer un apprenant qui quitte la partie par un AgJ (en activant l'AgJ correspondant).

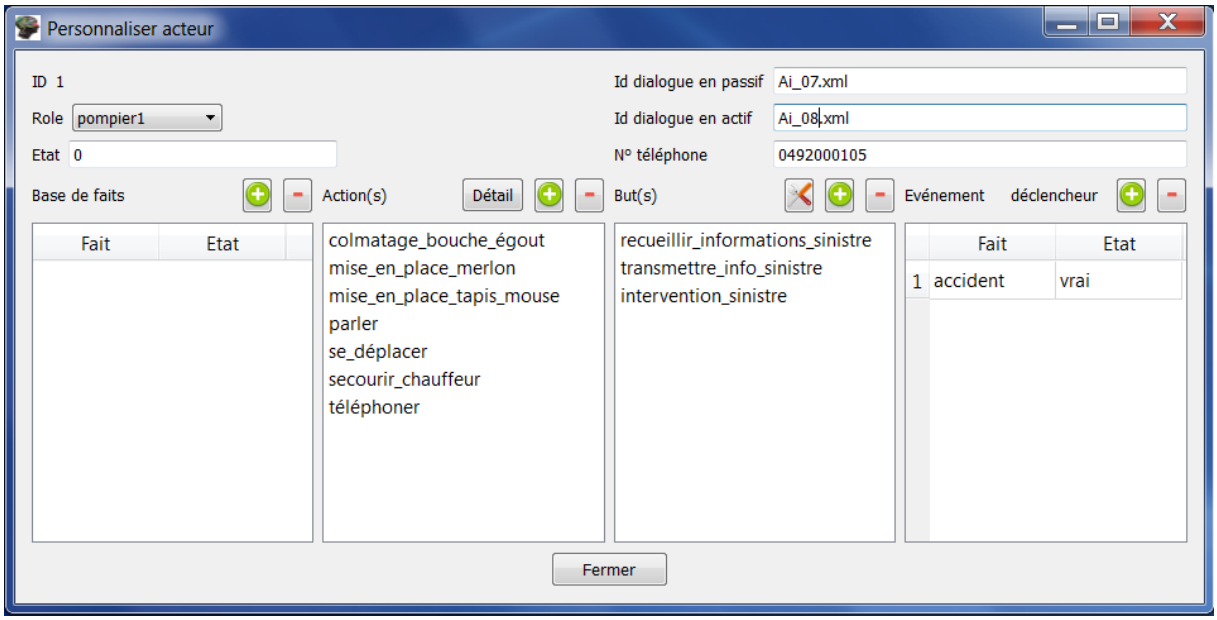


Figure V.4 – Capture d'écran de l'édition d'agent (AgJ).

Pour l'éditeur d'agent (figure V.4), le concepteur doit spécifier les propriétés suivantes (de l'AgJ) :

- Les *faits* présents dans la base de faits de l'agent, qui constitue les connaissances initiales d'un AgJ au début de la partie.
- Les *actions*, c'est à dire celles qui peuvent être réalisées par l'AgJ (ces actions serviront à définir des plans)
- Les *événements déclencheurs*, qui représentent une liste de faits qui peuvent déclencher certains buts
- Les *buts* de l'agent représentent les objectifs à atteindre. Chaque but est défini par :
  - un nom,
  - une priorité,

– des pré-conditions ,
– des pré-requis (but qui doit être réalisés précédemment),
– un ensemble de plans (permettant d'atteindre cet objectif), où chaque plan est représenté par un ensemble ordonné d'actions.

V.4.2 Éditeur de dialogue

Dans un scénario de gestion de crise, les différents intervenants doivent collaborer pour rétablir une situation nominale. Pour ce faire, il existe plusieurs processus d'interaction entre les acteurs pendant la partie. Conformément aux concepts de dialogue et de phrase définis précédemment (section IV.5.3), le concepteur peut concevoir des dialogues riches et interactifs avec des réponses adaptatives. Comme pour l'édition des AgJ (qui simulent les PNJs), un éditeur de dialogue à été mis en place pour la conception de dialogues. Les dialogues sont enregistrés au format XML. Ils peuvent être importés et réutilisés pour élaborer des dialogues plus complexes. La figue V.5 montre une capture de le l'éditeur de dialogue, à gauche on voit l'arborescence du dialogue, et à droite l'édition d'une phrase (qui représente un nœud de l'arbre) du dialogue avec les contenus, pré-conditions, etc.

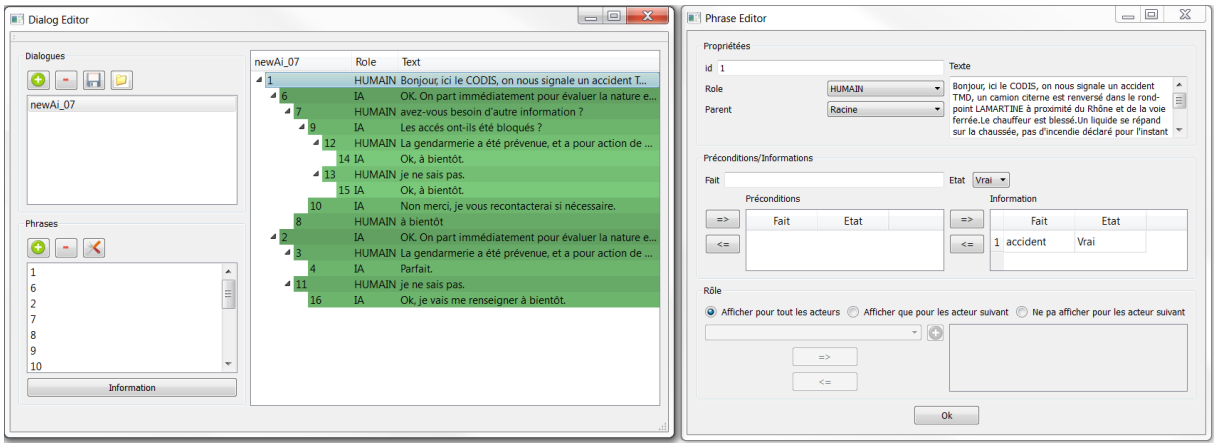


Figure V.5 – Capture d'écran de l'édition de dialogue.

Pour donner une idée de la structure d'un dialogue, la figure V.6 illustre l'arborescence d'un dialogue en détaillant les différents attributs d'une phrase (nœud de l'arbre). Ce dialogue représente une conversation de l'acteur CODIS (Centre Opérationnel Départemental d'Incendie et de Secours) et l'acteur pompier pour un scénario TMD (Transport de Matière Dangereuse). Le CODIS appel le pompier pour le prévenir de l'accident. La première phrase va transmettre le fait *(accident, vrai)*, et est destinée à l'acteur pompier uniquement (type d'affichage : acteur dans la liste). Le pompier a la possibilité entre deux choix de réponse, si l'acteur est un joueur humain (un apprenant), il choisira une des deux selon sa volonté. Si l'acteur est un PNJ (similé par un AgJ), il choisira une

réponse selon sa base de fait, car la phrase : "*Ok, on part immédiatement pour évaluer la nature et l'importance de l'accident*" a une pré-condition *(route bloquée, vrai)*. Si l'AgJ n'a pas l'information, il va poser la question sur l'état de la route. Même chose pour la réponse du CODIS. Si le CODIS possède l'information, il va répondre que la route est bloquée sinon il va répondre qu'il en sais rien.

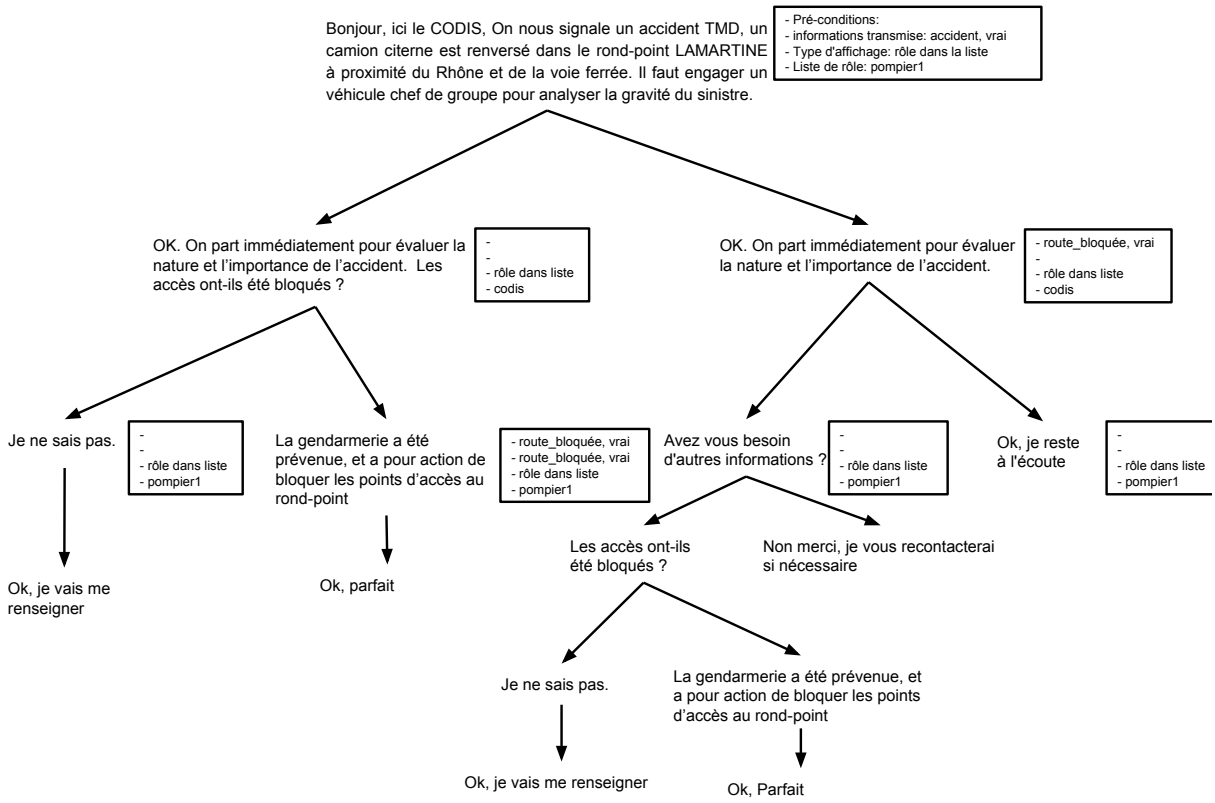


Figure V.6 – Exemple d'arborescence d'un dialogue.

V.5 Architecture

Dans cette section nous verrons l'architecture Client/Serveur de SIMFOR, comment lancer un serveur SIMFOR, comment se connecter à un serveur SIMFOR ainsi que les différentes actions et outils de suivi.

V.5.1 Serveur SIMFOR

La figure V.7 illustre la fenêtre de création d'un serveur SIMFOR. Tout d'abord on commence par donner un nom à la session, ensuite on choisit un scénario parmi les scénarii disponibles (crée avec l'aide d'un expert du domaine), puis on choisit une ver-

sion du scénario (on peut avoir plusieurs variations d'un même scénario, par exemple les comportement des PNJs, déclenchement ou pas de scripts spécifique, etc.), et enfin on choisit un port (réseau). De plus, on peut choisir un mot de passe pour verrouiller la session (afin de sélectionner les apprenants autorisés à rejoindre la partie). Les informations de la base de données sont liées au système d'information géographique (données SIG). Ces informations sont issues de données réelles fournies par les communes (dans notre cas la commune d'Arles). Ces informations contiennent des méta-données très importantes pour évaluer l'impact d'une crise majeure sur la commune. Parmi ces données on peut trouver par exemple, le nombre d'habitants pour chaque bâtiment (par exemple 350 élèves dans le collège Frédéric Mistral), l'emplacement des bouches d'égouts, la position des arbres ainsi que leur type (propriété de combustion), etc. Le serveur de base de données SIG est lancé soit sur une machine en local, soit sur un serveur internet, et les informations affichées dans la figure V.7 servent à se connecter à cette base.

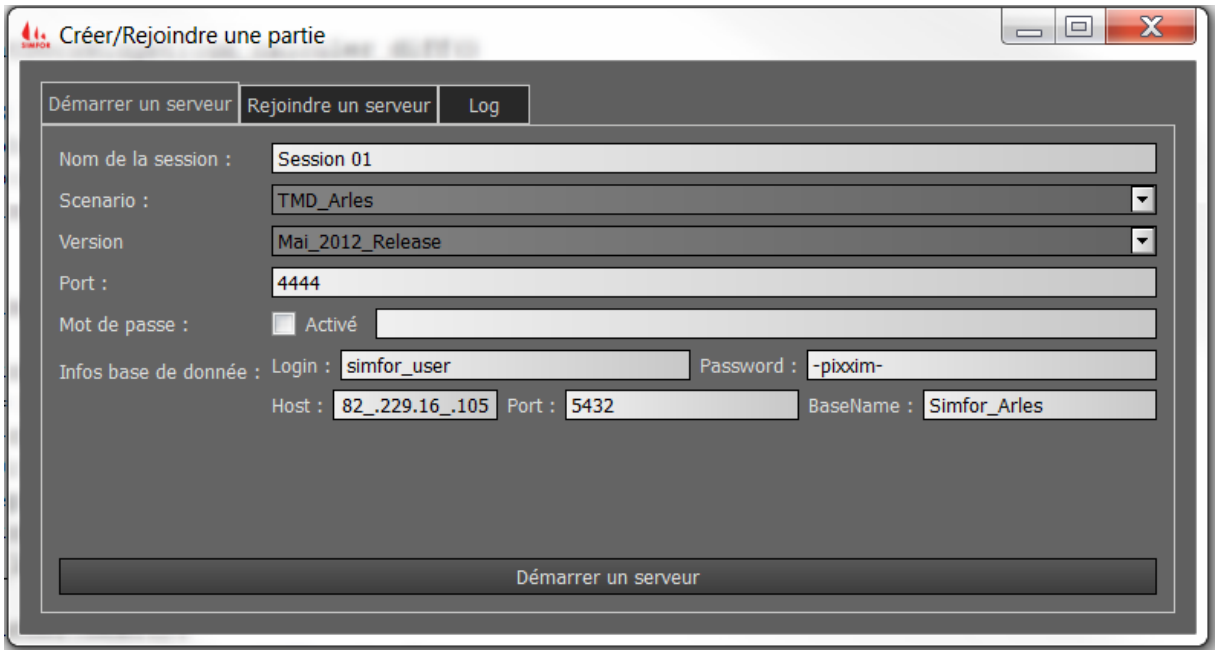


Figure V.7 – Création d'un serveur SIMFOR.

Une fois le serveur lancé, nous aurons une nouvel interface (figure V.8), cette interface permet de suivre se qui se passe durant la session de jeu. Dans cette interface, nous avons tout d'abord le temps de simulation, le taux de rafraichissement (fixé à 50 si la machine peut fournir le calcul nécessaire), et enfin la liste des joueurs (trois joueurs connectés dans la capture). Dans la deuxième partie de la fenêtre nous avons plusieurs onglets qu'on peut masquer ou afficher dans une fenêtre à part. Nous décrivons brièvement le rôle de chaque onglet :

– *Script* : Cet onglet permet d'exécuter des scripts Lua qui peuvent influer le déroulement de la session.

– *Exploration des modèles* : Cet onglet permet de parcourir tous les modèles du scénario et de voir leurs propriétés (état, position, etc.), cet onglet est utilisé surtout par les développeurs.
– *IA* : cet onglet permet de suivre l'évolution des AgJ durant la partie (nous verrons plus en détails cette fonctionnalité dans la section V.6).
– *Évaluation* : Cet onglet permet de consulter l'état social de l'organisation (nous reviendrons en détails dans la section V.7)
– *Log développeur* : Cet onglet permet aux développeurs de détecter d'éventuelles erreurs (problème de connexion par exemple) et ainsi faciliter le débugage.
– *Action* : Cet onglet permet de déclencher des actions dans SIMFOR, par exemple démarrer une partie, déclencher l'inflammation du carburant, etc. et ainsi influencer directement le déroulement de la partie.

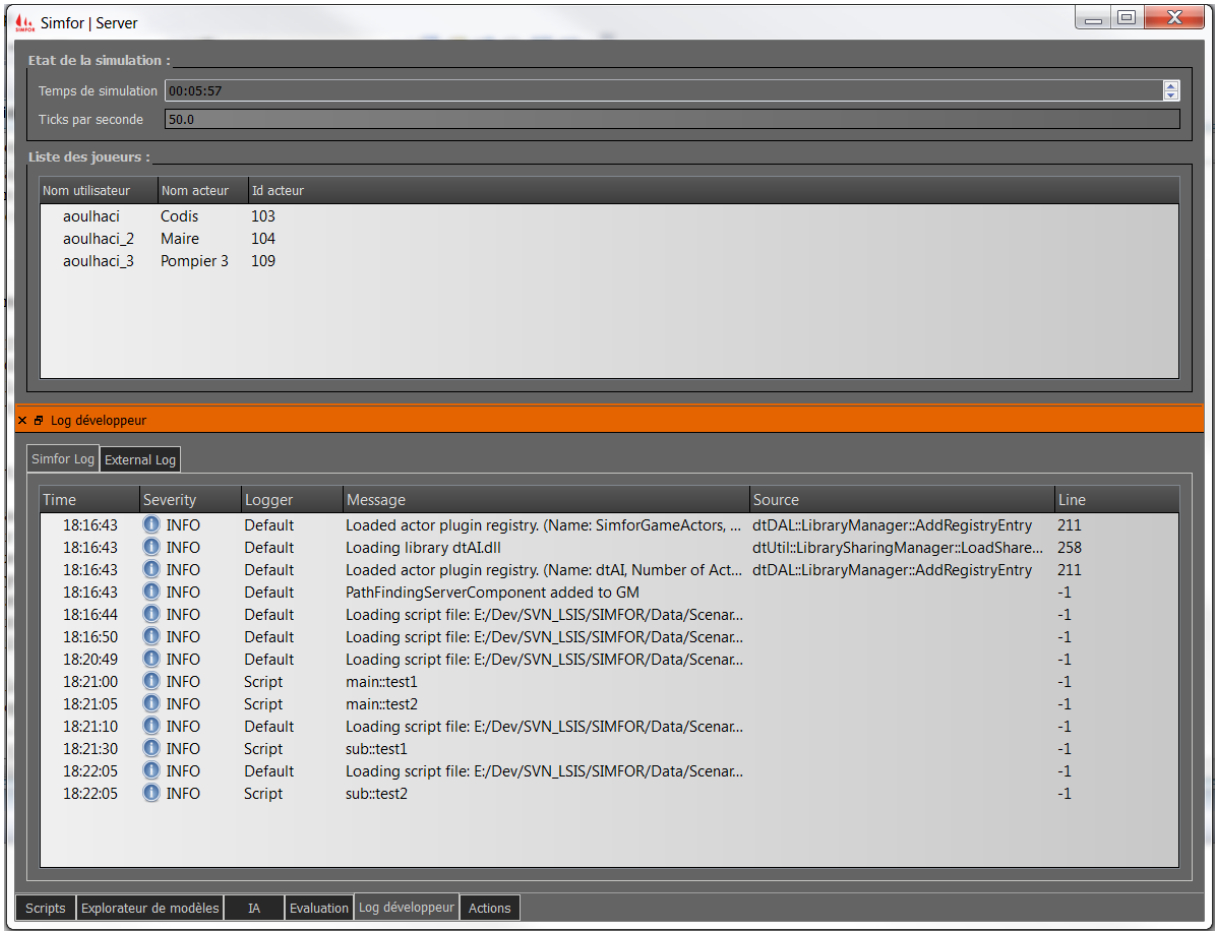


Figure V.8 – Capture d'écran du serveur SIMFOR.

V.5.2 Client SIMFOR

Avant de commencer à jouer, l'apprenant doit d'abord rejoindre une session de jeu (figure V.9). Pour cela l'apprenant va dans l'onglet "*rejoindre partie*" et peut soit rejoindre

une partie en réseau (figure V.9(a)), ou bien sur internet en renseignant les informations nécessaires tel que l'adresse IP, le port, etc. (figure V.9(b)).

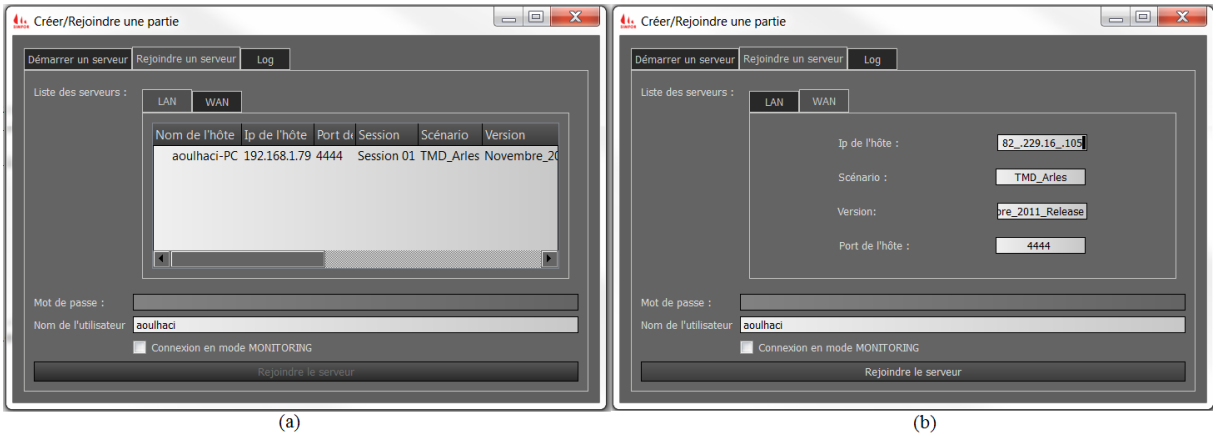


Figure V.9 – Rejoindre un serveur SIMFOR.

Une fois connecté au serveur, l'apprenant accède à l'interface de sélection du personnage (rôle à incarner, figure V.10). Dans cette interface, une courte description du scénario est présentée avec l'objectif du scénario, l'horaire ainsi qu'un court synopsis. L'apprenant doit choisir un rôle à incarner et attendre le début de la session (attendre la connexion des autres apprenants). Les rôles qui sont grisés ne peuvent pas être choisis car, soit ils ont été sélectionnés pas d'autres apprenants, ou bien sont incarnés par des AgJ.

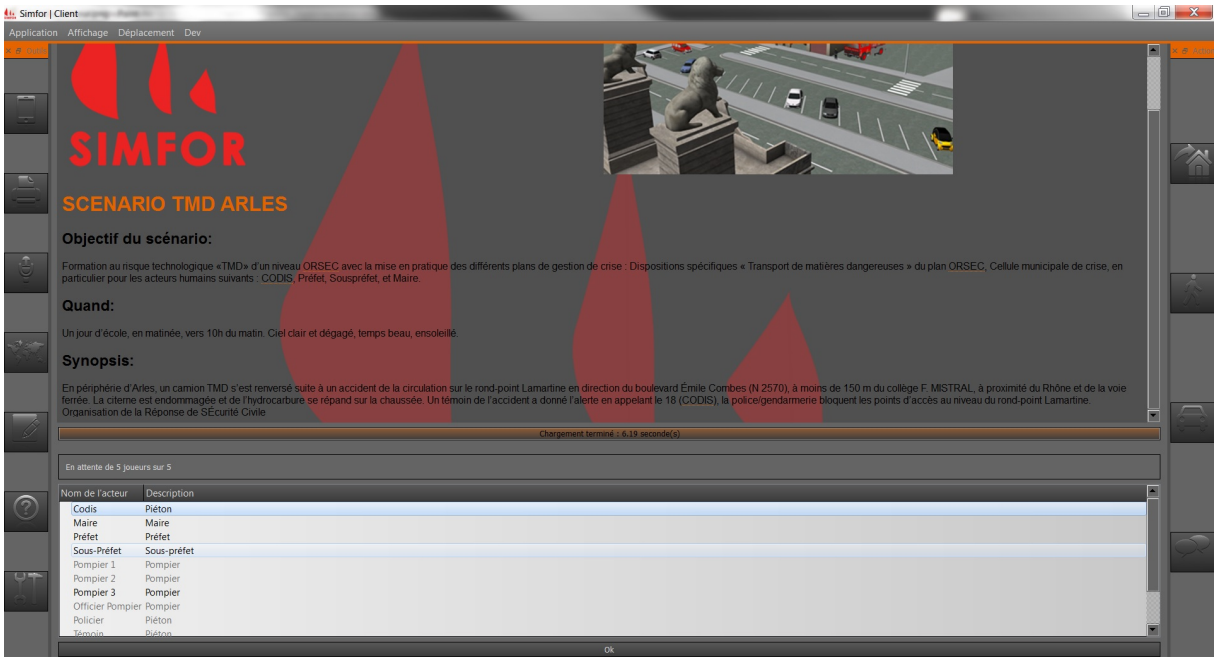


Figure V.10 – Interface de sélection du personnage.

Une fois que la session commence, l'apprenant accède à l'interface du jeu (figure V.11). De cette interface l'apprenant a une vue du monde virtuel 3D (avec un mode

de vu à la troisième personne) et peut se déplacer dans cet environnement en cliquant sur un point avec la souris (style de *jeu point & clic*). L'apprenant peut aussi avoir un autre mode de vue (voir annexe B) qui est la vue à la première personne (First Person Shooter, FPS), dans ce mode de vue l'apprenant se déplace avec les flèches (haut, bas gauche, droite) ou bien via une manette "*joystick*". L'apprenant peut aussi réaliser plusieurs actions (via les boutons de l'interface) :

- *Utiliser le téléphone* : quand l'apprenant appuie sur le bouton téléphone, une nouvel interface apparait (voir annexe B) avec un champ pour saisir un numéro de téléphone, ou bien sélectionner directement un numéro dans l'annuaire.
- *Envoyer un fax* : comme pour le bouton téléphone, le bouton fax donne accès à une nouvelle interface (voir annexe B) qui permet de sélectionner un fax à partir d'une liste de documents, que l'apprenant remplit avec les informations nécessaires, puis saisit (ou sélectionne dans l'annuaire) un ou plusieurs destinataires, et enfin envoie le fax.
- *Utiliser la radio* : le bouton radio donne accès à une nouvelle interface (voir annexe B) où l'apprenant sélectionne un canal puis appuie sur le bouton parler pour communiquer avec les acteurs qui sont connectés au même canal.
- *Ouvrir la carte* : le bouton carte permet d'ouvrir une interface qui permet de voir la carte de la commune (voir annexe B), via cette interface l'apprenant peut marquer des points sur la carte, tracer des périmètres, voir la position des moyens, etc.
- *Ouvrir l'outil Main courante* : le bouton main courante donne accès à une interface (voir annexe B) qui simule un portail web qui relai l'information sur le sinistre. Dans cette interface, l'apprenant peut poster des nouvelles sur le sinistre et suivre en temps réel les informations postées par les autres apprenants.
- *Ouvrir l'aide* : le bouton aide permet d'afficher l'aide et les propositions de l'agent pédagogique (voir annexe B).
- *Ouvrir l'outil Action* : le bouton outil action, permet de déclencher des actions propres au scénario, et cela en sélectionnant une action disponible (dépend du rôle incarné et du scénario) via l'interface (voir annexe B).
- *Sortir/entrer dans un bâtiment* : ce bouton permet d'accéder à l'extérieur si l'apprenant se trouve à l'intérieur d'un bâtiment ou bien d'entrer dans un bâtiment si l'apprenant se trouve à l'extérieur.
- *Se déplacer vers une adresse* : ce bouton permet à l'apprenant de déplacer son avatar de manière automatique en sélectionnant une adresse dans l'annuaire d'adresse (voir annexe B).
- *Utiliser un véhicule* : ce bouton permet à l'apprenant de déplacer en utilisant un véhicule (l'apprenant doit être à proximité d'un véhicule), cela se fait de manière automatique en sélectionnant une adresse dans l'annuaire d'adresse (voir annexe B).

– *Dialogue de proximité* : ce bouton permet à l'apprenant de parler aux acteurs qui se trouvent à proximité (voir annexe B).



Figure V.11 – Interface de jeu durant la session d'entraînement.

V.6 Suivi et interaction des agents

La simulation des comportements a été développée comme une libraire externe à SIMFOR qui communique avec ce dernier via un système de commande. Durant la partie, dans l'interface du serveur SIMFOR on peut suivre l'évolution des AgJ, comme leur état, les messages qu'ils reçoivent, etc. (voir figure V.12). Via cette interface on peut aussi influer sur le comportement des AgJ en leur envoyant directement des commandes (par exemple envoyer une commande lui ajoutant un fait), ou bien le réinitialiser.

Pour avoir plus détails sur un AgJ, on peut "*zoomer*" sur ces activités en appuyant sur le bouton "*Voir détail*" (figure V.12). En appuyant sur le bouton "*Voir détail*", une nouvelle interface apparait (figure V.13) dans laquelle on peut voir la base de faits de l'AgJ, l'ensemble de ces buts (ceux qui sont satisfaits et ceux qui ne le sont pas), l'action en cours d'exécution (avec sa progression), le plan en exécution (avec les actions effectuées, en cours, et futures), ainsi qu'un log retraçant son activité. Via cette interface on peut modifier les propriétés de l'AgJ, comme ajouter, modifier ou supprimer un fait dans sa base de faits, on peut aussi réinitialiser un but (l'AgJ va le réaliser à nouveau).

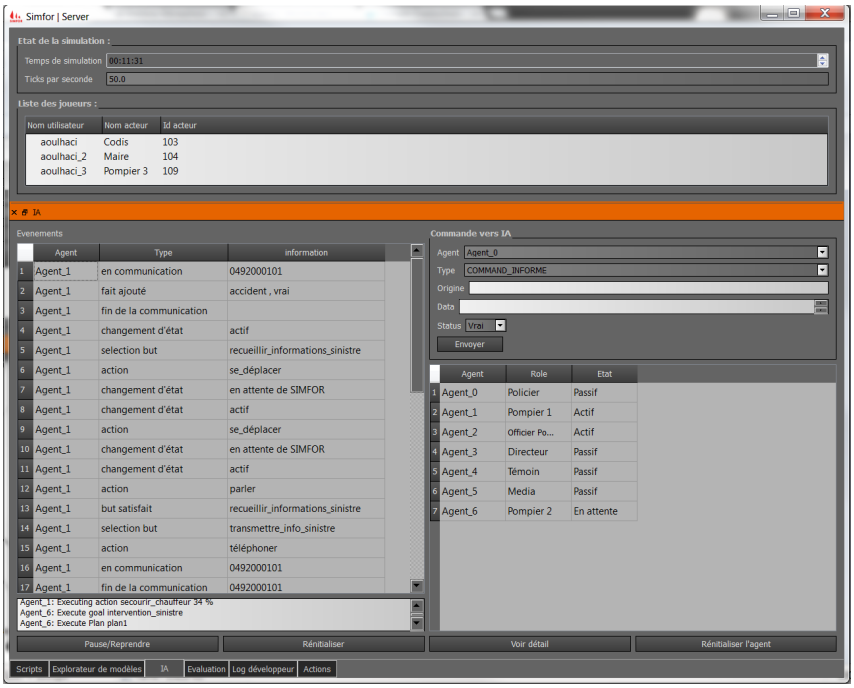


Figure V.12 – Interface de suivi et d'interaction des AgJ.

V.7 Évaluation

Nous avons intégré plusieurs outils de suivi dans SIMFOR. A tout moment, on peut voir l'historique des actions (figure V.14) pour la session en cours. On peut aussi calculer le couplage du réseau ainsi que le couplage différence (section III.4.3), et cela avec les différents types d'interaction (téléphone, radio, fax, parler), comme illustré dans la figure V.15.

Pour produire une évaluation, il faut d'abord instancier le modèle du domaine qui nous servira de référence pour calculer l'écart entre les actions des apprenants et celles

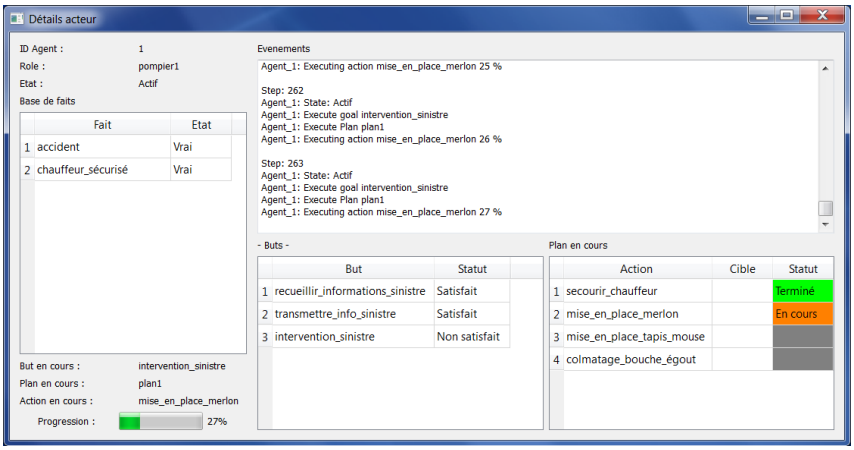


Figure V.13 – Interface détaillée du suivi d'un AgJ.



Figure V.14 – Suivi des actions durant la session d'entrainement.

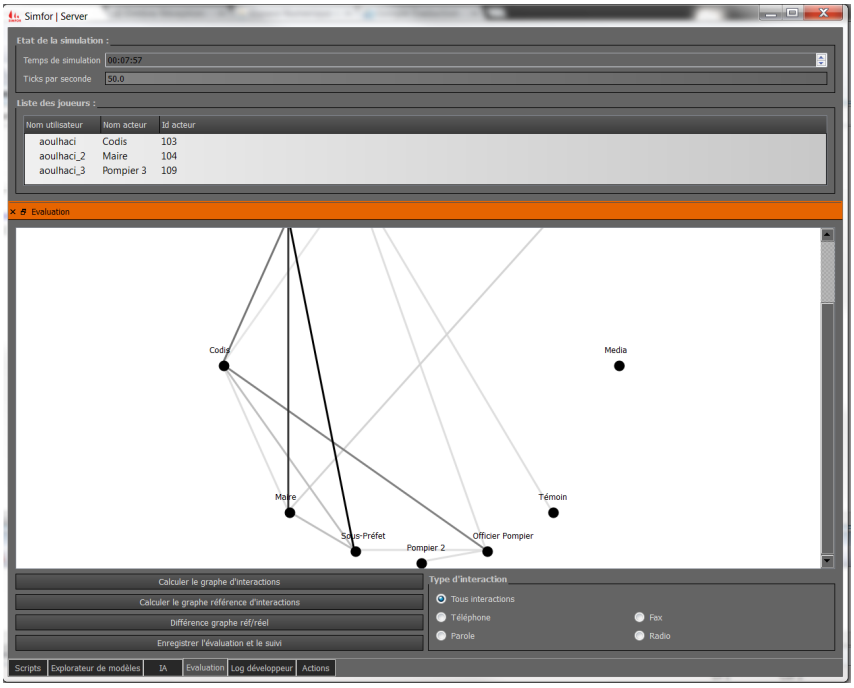


Figure V.15 – Calcul du couplage via le serveur SIMFOR.

prévues par l'expert. Pour faciliter la conception du modèle du domaine, nous avons mis à disposition du concepteur (de scénario) un éditeur qui permet de spécifier pour chaque rôle (apprenant) un modèle du domaine qui regroupe un ensemble de missions à réaliser pour un scénario donné. Dans la section V.7.1, nous présentons l'éditeur du modèle du domaine, et dans la section V.7.2, nous présentons un outil pour visualiser et analyser les résultats obtenus après une session d'entrainement.

V.7.1 Éditeur du modèle du domaine

L'éditeur du modèle du domaine permet au concepteur (de scénario) de spécifier pour chaque rôle du scénario (figure V.16(a)) un modèle du domaine qui regroupe un

ensemble de missions à réaliser pour un scénario donné (figure V.16(b)).

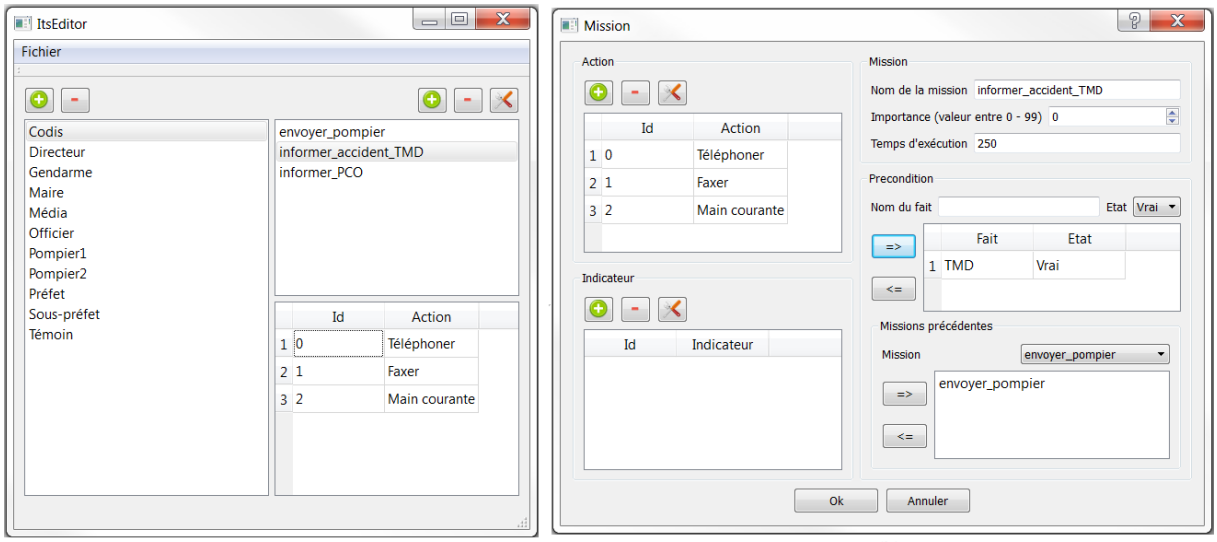


Figure V.16 – Éditeur du modèle du domaine.
(a) : interface principale.
(b) : édition des missions.

L'interface principale de l'éditeur (figure V.16(a)) permet d'ajouter des rôles au modèle du domaine. En cliquant sur un rôle dans la liste, l'éditeur affiche l'ensemble des missions liées à ce rôle. En cliquant sur une mission, l'éditeur affiche l'ensemble des actions liées à cette mission. Le concepteur a la possibilité d'ajouter ou éditer une mission en cliquant sur le bouton *ajouter* (ou *éditer* pour l'édition). Dans l'interface d'édition des missions (figure V.16(b)), le concepteur donne un nom à la mission, l'importance de cette mission, le temps d'exécution estimé, des pré-conditions, des missions à réaliser au préalable, un ensemble d'actions, et enfin un ensemble d'indicateurs spécifiques (utilisés pour l'évaluation).

V.7.2 Moniteur d'évaluation

Une fois la session d'entrainement finie, SIMFOR enregistre les données d'évaluation (actions, interactions, supports, ...) dans un fichier XML. Le moniteur d'évaluation permet la manipulation et l'exploitation de ces données. La figure V.17 illustre l'interface principale du moniteur d'évaluation.

L'interface principale du moniteur d'évaluation affiche l'ensemble des acteurs du scénario. En cliquant sur un acteur, l'interface affiche l'ensemble des actions réalisées par l'acteur, l'évaluation des actions et missions réalisées ainsi que l'historique de support apporté (si l'acteur est joué par un AgJ, seul l'historique des actions est affiché).

Le moniteur d'évaluation peut aussi produire l'évaluation individuelle pour chaque

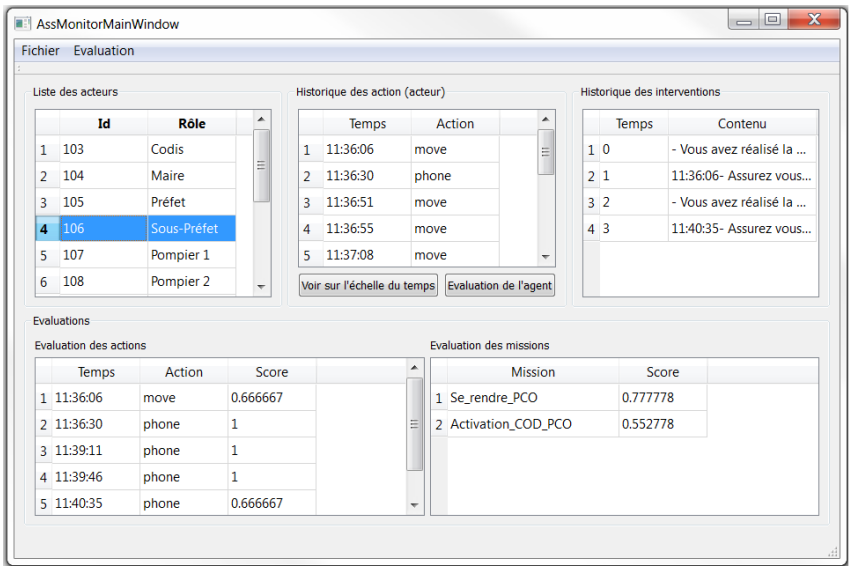


Figure V.17 – Interface principale du moniteur d'évaluation.

apprenant ainsi qu'une évaluation globale du groupe. L'évaluation individuelle est calculée sur la base du score social (calculer sur la base des différences entre le couplage du réseau et couplage référence, voir section III.4.3) et la moyenne des scores des missions (figure V.18(b)). Le score global du groupe est basé sur une moyenne des scores individuelles combinée au score social ainsi que des indicateurs globaux (par exemple : pertes matérielles, coût de l'intervention, etc), l'évaluation globale est illustrée dans la figure V.18(b).

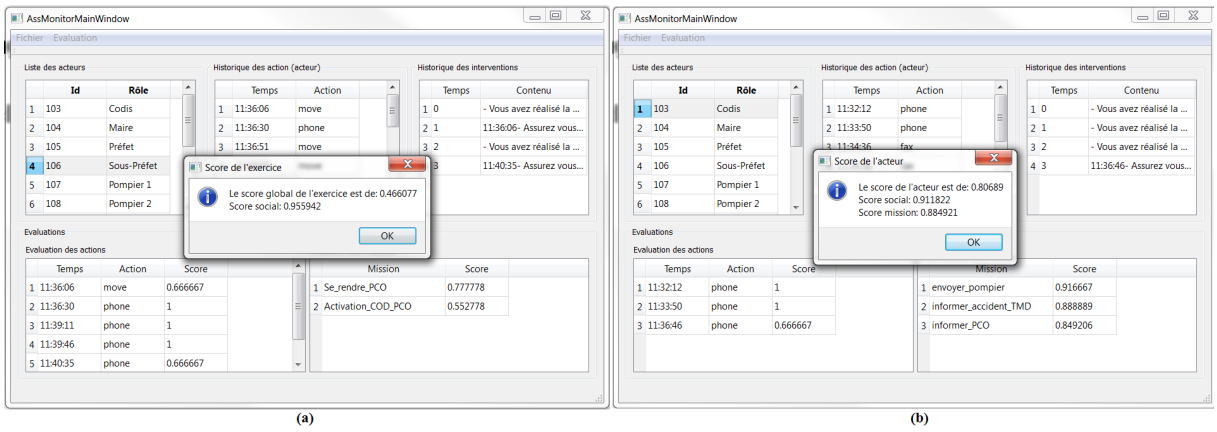


Figure V.18 – Évaluation individuelle et collective du moniteur d'évaluation
(a) : évaluation collective.
(b) : évaluation individuelle de l'acteur *codis*.

Le moniteur d'évaluation permet aussi de visualiser l'historique du couplage (figure V.19), qui est calculé tout au long de la session (voir III.4.3). Le fort couplage dans le graphe permet d'identifier les moments clés de l'exercice qui reflète un grand nombre d'échanges avec une courte durée.

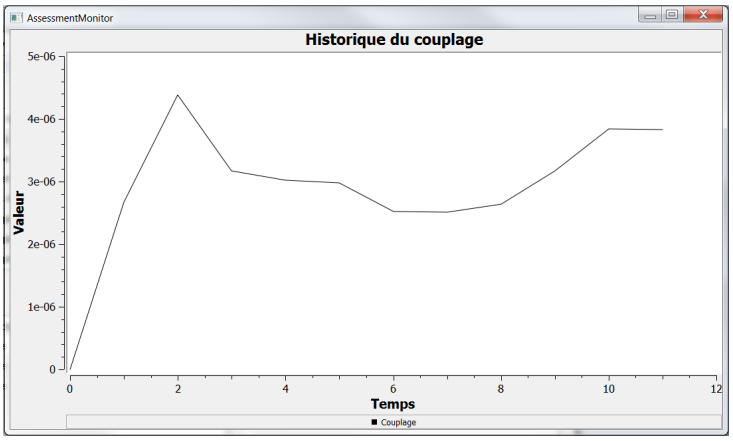


Figure V.19 – Historique du couplage.

Le moniteur d'évaluation peut aussi afficher le couplage du réseau (figure V.20(a)), le couplage référence (figure V.20(b)) ainsi que le couplage différence (figure V.20(c)) entre les deux, ces graphes permettent d'avoir une vision globale sur les échanges entre les différents acteurs. Les traits bleus dans le graphe différence correspondent à une différence positive (ex : entre l'acteur CODIS et le pompier1), cela veut dire qu'il y a eu un plus fort couplage que prévu. L'intensité de la couleur correspond à la valeur de la différence (plus la couleur approche du bleu, plus la différence est grande). Les traits rouges dans le graphe différence correspondent à une différence négative (par exemple : entre l'acteur préfet et le maire), cela veut dire qu'il y a eu un plus faible couplage que prévu. L'intensité de la couleur correspond à la valeur de la différence comme pour la différence positive.

En plus d'afficher l'historique des actions via l'interface principale, le moniteur d'évaluation a aussi la possibilité d'afficher l'historique des actions sur l'échelle du temps (figure V.21), et cela que ça soit pour un acteur donné ou bien pour l'ensemble des acteurs. Cet outil permet d'avoir une vision globale sur la réalisation des objectifs de chaque acteur (et en déduire les causes et effets), et avoir un "*débriefing*" avec les apprenants.

V.8 Les agents

Dans cette section nous verrons en détail comment sont implémentés les différents agents qui interviennent dans l'architecture de SIMFOR.

Les AgJ

Un AgJ est basé sur l'architecture BDI avec des buts, des désirs et des intentions. L'AgJ commence par mettre à jour sa base de faits en percevant l'environnement et en

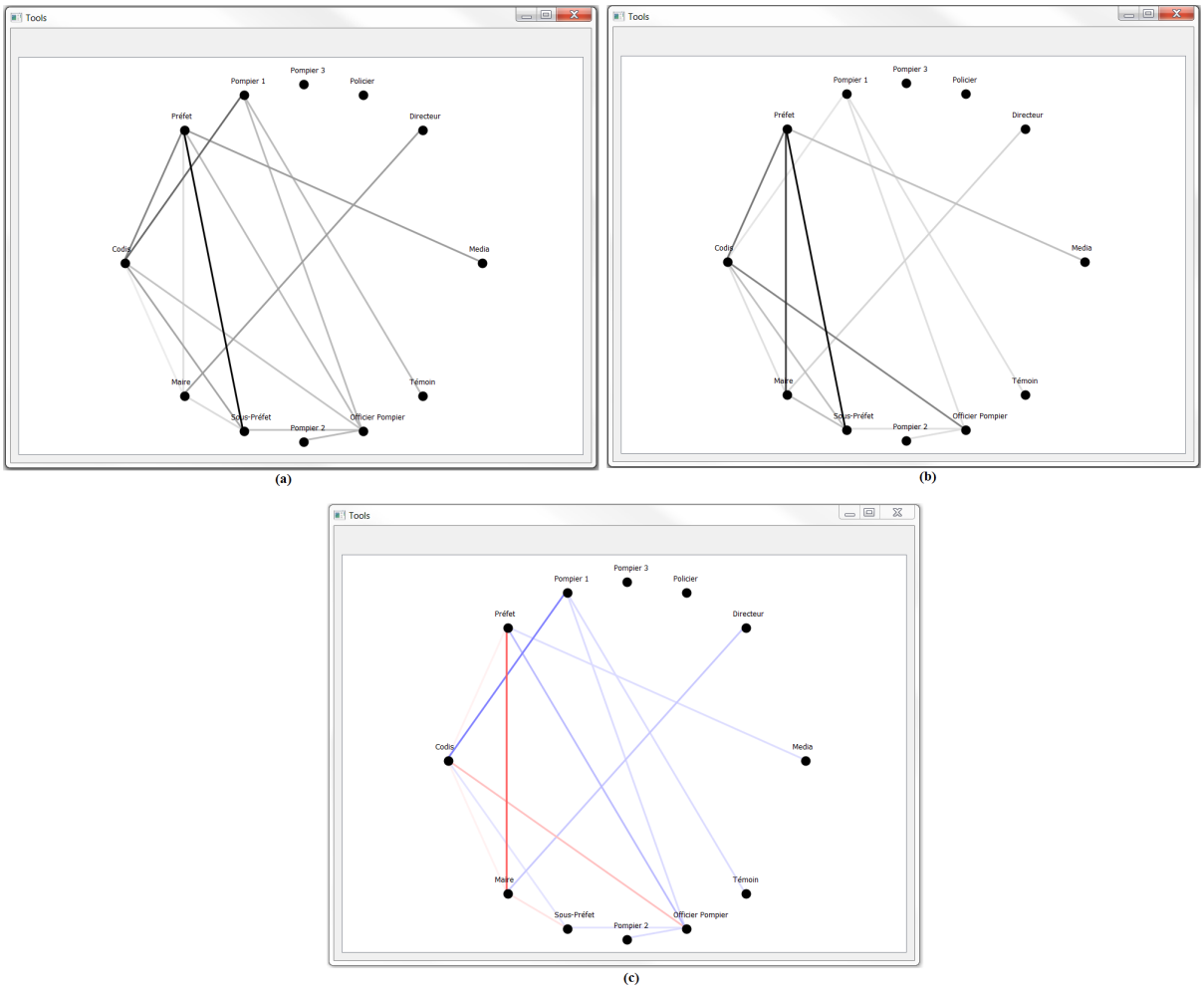


Figure V.20 – Les différents graphes de couplage.
(a) : couplage du réseau (exercice).
(b) : graphe référence (calculer à partir du modèle du domaine).
(c) : graphe différence.

lisant les messages des autres agents, puis va sélectionner le but adéquat à sa situation (rôle, état) puis sélectionne l'intention (plan) qui lui permettra de réaliser ce but et enfin exécute le plan. La figure V.22 décrit le moteur de raisonnement de l'AgJ. Ce raisonnent est principalement guidé par la base de faits, les buts, et les changements dans l'environnement.

Une fois un but sélectionné, et le plan choisi (pour réaliser le but), l'AgJ commence à exécuter l'action (qui peut durer plusieurs pas de temps). A chaque pas de temps (durant l'exécution de l'action), l'AgJ vérifie la faisabilité de l'action (vérifie si les pré-conditions sont toujours vérifiées). Si les pré-conditions ne sont plus vérifiées, alors l'AgJ interrompt l'action et abandonne le but en cours. Durant chaque pas de temps, l'AgJ vérifie aussi si un but plus prioritaire (que celui en cours) est devenu réalisable (pré-conditions vérifiées). Si un but plus prioritaire devient réalisable, alors l'AgJ interrompt

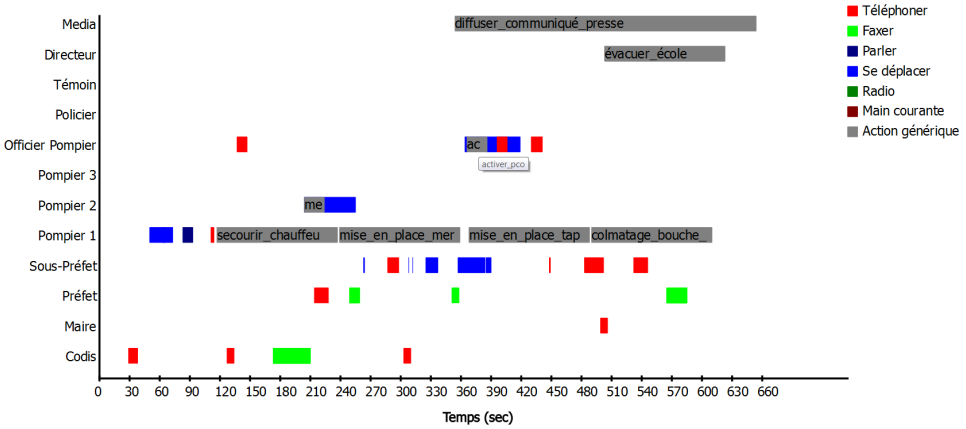


Figure V.21 – Historique des actions pour l'ensemble des acteurs du scénario.

le but en cours et le met dans une file d'attente. Une fois un but réalisé, et si la file d'attente n'est pas vide, l'AgJ sélectionne le but le plus prioritaire dans la file d'attente, et reprend là où il s'est arrêté.

Dans notre cas d'application (le jeu sérieux SIMFOR), le pas temps a été fixé à une seconde, c'est à dire que l'AgJ exécute son raisonnement chaque seconde. Nous avons décidé de ne pas prendre un pas de temps plus court pour ne pas impacter les performances de la simulation.

L'agent évaluateur

L'agent évaluateur (AgE) a pour mission de fournir des évaluations à l'agent pédagogique (AgP). Ces évaluations peuvent être liées à la réalisation d'une action ou d'une mission. La figure V.23 décrit le moteur de raisonnement de l'agent évaluateur. Lorsque l'AgE reçoit une requête d'évaluation de la part de l'AgP, l'AgE commence par vérifier si l'apprenant est en train de réaliser une mission. Si c'est le cas, l'AgE vérifie que l'action réalisée entre bien en compte dans la mission en cours, si l'action référence est trouvée alors l'AgE réalise un ActionAssessment (évaluation d'une action, voir section III.4.2), sinon l'AgE cherche une autre mission pour l'action réalisée dans le modèle du domaine (pour récupérer l'action référence). Si L'AgE ne trouve pas la mission pour l'action réalisée alors le score de l'évaluation sera zéro, sinon la mission trouvée sera la mission en cours et l'AgE réalise un ActionAssessment.

Si l'apprenant n'a pas de mission en cours, alors l'AgE cherche une mission pour l'action réalisée dans le modèle du domaine (pour récupérer l'action référence). Une fois la mission trouvée l'AgE réalise un ActionAssessment, si l'AgE n'arrive pas à déterminer la mission pour laquelle l'apprenant a réalisé l'action, alors le score sera nul.

Après chaque ActionAssessment, L'AgE vérifie si l'apprenant a finalisé une mission

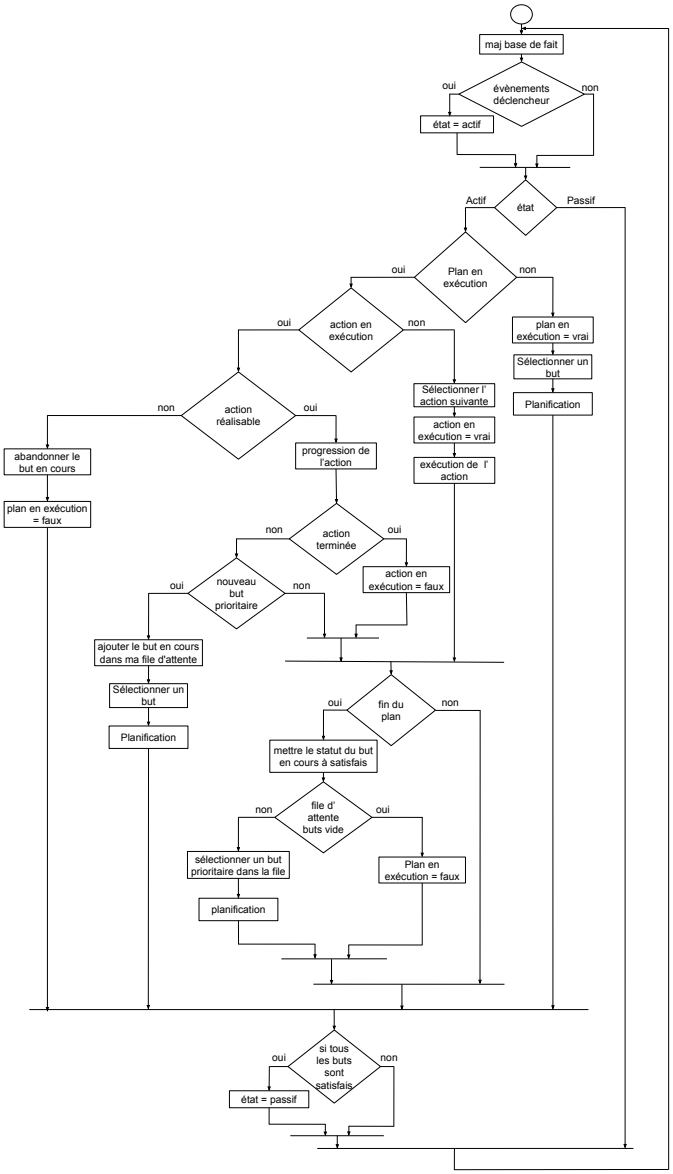


Figure V.22 – Moteur de raisonnement de l'AgJ.

(l'apprenant a réalisé l'ensemble des actions de la mission). Si c'est le cas alors l'AgE réalise un MissionAssessment (évaluation d'une mission, voir section III.4.2) et envoi un signal à l'AgP pour l'informer en lui transmettant les informations nécessaires (nom de la mission, score global, score actions, etc. voir section III.4.2).

L'agent pédagogique

L'agent pédagogique a pour mission d'analyser la situation (résultat d'évaluation, modèle apprenant, modèle du domaine, etc.) et de fournir un support à l'apprenant, ce support peut prendre plusieurs formes :

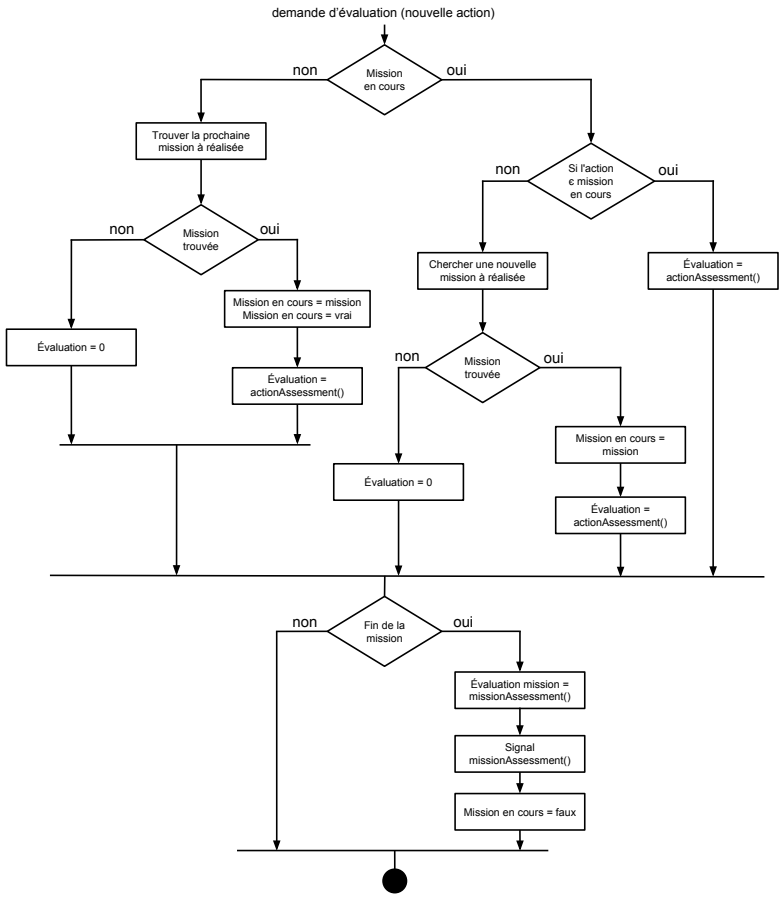


Figure V.23 – Moteur de raisonnement de l'agent évaluateur.

- **Correction (PopUp) :** ce type de support survient lorsque l'évaluation fourni par l'AgE est faible. L'AgP fournit alors une correction à l'apprenant en indiquant l'action à réaliser dans une fenêtre qui affiche au premier plan (popup, voir annexe B).
- **Astuce :** ce type de support survient lorsque l'évaluation fournie par l'AgE est élevée ou que l'apprenant reste inactif un certain temps (ne sachant pas quoi faire par exemple). L'AgP fournit alors une astuce (prochaine action à réalisé, comment faire pour envoyer un fax, etc.) qui sera affichée dans la fenêtre d'aide et que l'apprenant pourra consulter en appuyant sur le bouton d'aide.
- **Résultats d'évaluation :** ce type de support survient lorsque l'AgE fourni l'évaluation d'une mission. L'AgP prévient l'apprenant qu'il vient de finir une mission (objectif) en spécifiant le nom de la mission, score, efficacité des actions, etc. Ces informations permettent de tenir au courant l'apprenant de sa performance et peut lui servir pour se corriger, par exemple essayer de réaliser la prochaine mission plus rapidement si le score lié au temps d'exécution était faible. Ce type de support peut être désactivé pour ne pas biaiser les résultats de l'exercice.

La figure V.24 décrit le moteur de raisonnement de l'agent pédagogique. L'AgP est sollicité de deux manières, soit par l'agent de l'apprenant lors d'une nouvelle action (ou connaissance), soit par l'agent évaluateur lorsque ce dernier détecte (et évalue) la fin d'une mission.

Si l'AgP est sollicité pour la réalisation d'une mission, alors l'AgP prépare le texte à afficher dans la boite de dialogue, pour cela l'AgP récupère le résultat de l'évaluation (requête à la l'AgE) puis construit le texte à afficher avec le nom de la mission, le score ainsi que les scores détaillés (score temps, scores actions, etc.). Si la mission est réalisée par un apprenant novice alors l'AgP va chercher la prochaine mission à réaliser (à partir du modèle du domaine) pour l'intégrer au texte affiché (indication à l'apprenant).

Si l'AgP est sollicité pour une action, alors l'AgP demande à l'AgE de faire une évaluation en temps réel. Si le résultat de l'évaluation est -1, cela veut dire que soit l'apprenant n'a réalisé aucune action, soit l'AgE n'arrive pas à déterminer la mission référence de l'action. Dans ce cas de figure L'AgP considère que l'apprenant est dans une phase d'inactivité (ou blocage) et comptabilise le temps d'inactivité.

Lorsque le temps d'inactivité dépasse un certain seuil (réglé par l'expert du domaine) alors l'AgP fourni un support à l'apprenant (qui dépend du niveau de l'apprenant). Si l'AgE arrive à fournir une évaluation pour l'action réalisée alors l'AgP fournit un support à l'apprenant. Si le score est élevé, alors l'AgP fourni un support de type astuce (indiquer la prochaine action à réaliser par exemple). Si le score est faible et que l'apprenant n'est pas expert alors l'AgP fournit un support de type PopUp (correction de l'action avec fenêtre au premier plan).

L'agent apprenant

L'agent apprenant (AgA) a pour mission de collecter les actions et les connaissances liées à un acteur donné et mettre à jour son modèle du domaine. À chaque acteur est associé un agent apprenant. Pour collecter les informations liées à l'acteur suivi, l'AgA doit s'abonner à un agent source de données (AgSD), un AgSD liée à SIMFOR pour les apprenants et un AgSD lié à la simulation de comportement pour les AgJ. Lorsque le modèle de l'apprenant est mis à jour, l'AgA informe l'AgP pour déclencher la rétroaction (si l'acteur suivi est un apprenant).

V.9 Conclusion

Dans ce chapitre nous avons présenté l'implémentation de l'architecture présentée dans le chapitre IV avec le project SIMFOR, un jeu sérieux pour la formation à la gestion

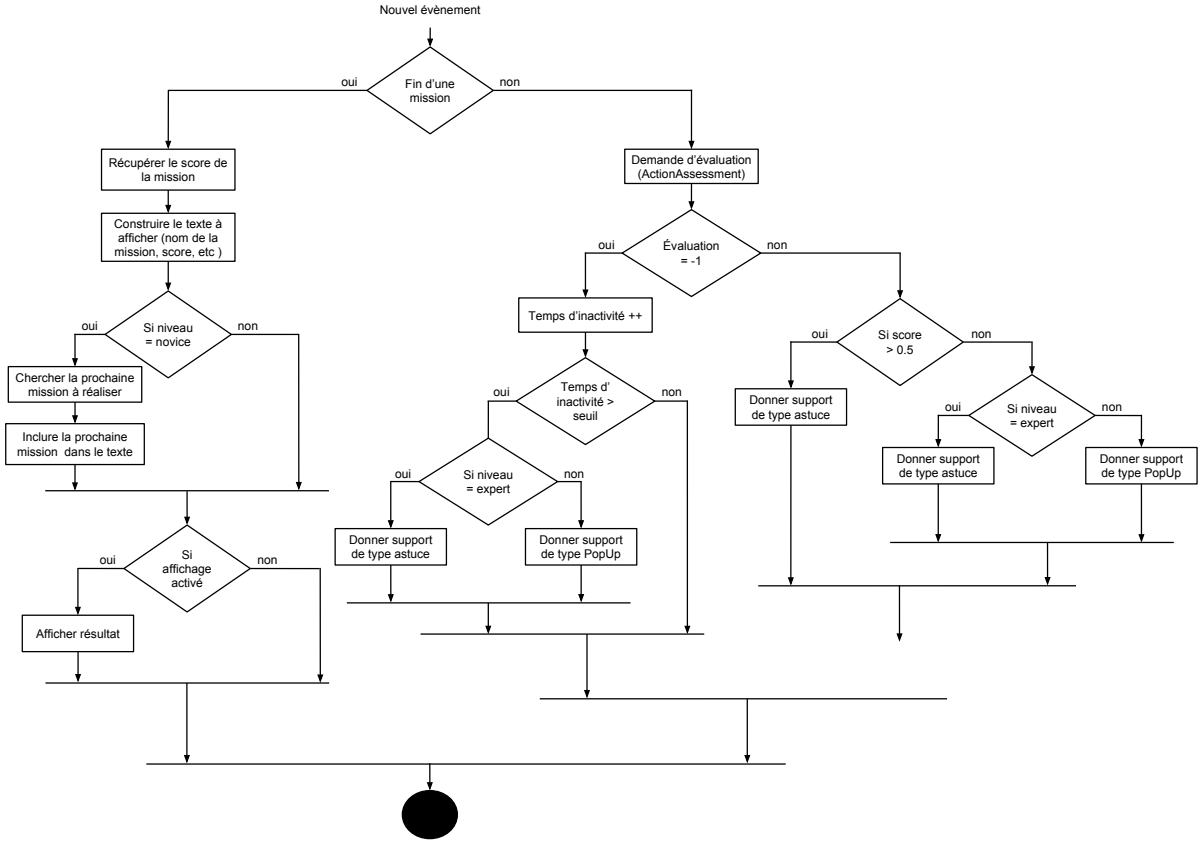


Figure V.24 – Moteur de raisonnement de l'agent pédagogique.

de crise. Nous avons aussi présenté les différents outils d'édition et de suivi des agents impliqués dans cette architecture. En plus des applications implémentées, nous avons aussi présenté le moteur de raisonnent des agents mis en œuvre dans le projet SIMFOR. Ainsi nous avons présenté le serveur SIMFOR qui héberge une session d'entrainement (avec un scénario donné), le client SIMFOR qui permet de participer à une session d'entrainement en se connectant à un serveur SIMFOR (en réseau local ou par internet). Nous avons ensuite présenté l'éditeur des AgJ qui permet de configurer le comportement des AgJ pour un scénario donné (comportements nominaux ou volontairement erronés), l'éditeur de dialogue qui permet de créer des dialogues interactifs entre les apprenants et les AgJ (ou entre les AgJ), ainsi que l'éditeur du modèle du domaine qui permet à l'expert du domaine de spécifier les missions (objectifs) à atteindre pour un rôle dans un scénario donné. Nous avons enfin présenté le moniteur d'évaluation qui permet de parcourir les résultats d'évaluation pour une session d'entrainement ainsi que le calcul d'un score basé sur des évaluations individuelles et collectives.

Dans le chapitre suivant nous présentons un exemple de scénario de gestion de crise ainsi que l'expérimentation de ce scénario avec des sessions de jeu et les résultats de l'évaluation.

CHAPITRE VI

Illustration et validation sur un exemple de scénario de gestion de crise

Sommaire

VI.1 Introduction 134

VI.2 Quelques définitions 134

VI.2.1 Définition d'un scénario dans SIMFOR 134

VI.2.2 Les attributs d'un scénario SIMFOR 135

VI.2.2.1 Les acteurs 135

VI.2.2.2 Les actions 135

VI.2.2.3 Les moyens 135

VI.2.2.4 Les environnements 135

VI.2.2.5 Les sinistres 136

VI.3 Présentation du scénario TMD 137

VI.3.1 Fonctionnement général 137

VI.3.2 Déroulement du scénario 139

VI.4 Validation sur le Scénario TMD 139

VI.4.1 Protocole de validation 139

VI.4.2 Résultats et discussions 141

VI.5 Conclusion 144

VI.1 Introduction

Dans ce chapitre nous présentons un exemple de scénario de gestion de crise en l'occurrence un scénario de transport de matières dangereuses (TMD) qui a servi de référence durant le développement initial de SIMFOR et de nos outils de suivi et d'évaluation des apprenants. Le scénario TMD est un scénario très intéressent car il implique plusieurs corps de métier comme les pompiers, la police, la gendarmerie, ainsi que différents postes administratifs (Maire, Préfet, Sous-Préfet, etc). De plus le scénario TMD peut évoluer en un scénario de pollution environnemental (en cas de fuite de produits chimiques), ou d'incendie de grande ampleur (en cas d'embrasement de produits inflammables), ce qui peut constituer un exercice très pertinent pour les intervenants de la gestion de crise. Nous présentons tout d'abord quelques définitions relatives au scénario de jeu SIMFOR, puis nous présentons le scénario TMD ainsi que son déroulement. Nous présentons aussi le protocole de validation que nous avons utilisé pour valider à la fois l'évaluation automatique des apprenants ainsi que l'amélioration de l'apprentissage des apprenants en utilisant l'outil SIMFOR. Enfin, nous présentons les résultats obtenus à travers notre expérimentation pilote ainsi qu'une discussion autour de ces résultats.

VI.2 Quelques définitions

Avant de présenter le scénario TMD, nous allons présenter les concepts de base qui nous permettent de définir un exercice de gestion de crise dans le cadre du projet SIMFOR.

VI.2.1 Définition d'un scénario dans SIMFOR

Un scénario correspond à la mise en scène d'une situation de crise due à la découverte d'un sinistre majeur, avec l'intervention et la mobilisation d'acteurs et de moyens dans un ou plusieurs environnements dans l'objectif de maîtriser le sinistre et de retourner à une situation normale. Les entités qui le composent, évoluent dans le temps par le biais d'actions et de comportements.

VI.2.2 Les attributs d'un scénario SIMFOR

VI.2.2.1 Les acteurs

Les acteurs sont définis dans un scénario par leur :
- Identité,
- Niveau d'aptitude,
- Fonction/ Rôle.

Les spécificités d'un acteur sont principalement contextuelles et non fonctionnelles. Les attributs d'un acteur n'ont qu'un caractère informatif. Les possibilités offertes à chaque acteur dépendent de leur environnement, des acteurs qui l'entourent et des moyens à sa disposition. Les acteurs sont joués soit par des apprenants, soit par des AgJ avec une substitution possible dans les deux sens à tout moment.

VI.2.2.2 Les actions

Les acteurs disposent d'une liste d'actions par défaut comme se déplacer dans l'environnement 3D ou parler. En plus de cela, les acteurs disposent d'actions contextuelles dépendant des moyens dont ils disposent, de l'environnement dans lequel ils sont, ainsi que du rôle incarné.

VI.2.2.3 Les moyens

Un moyen est une entité, plus précisément un outil qui propose des actions contextuelles propres à son utilisation et qui entraîne une ou plusieurs actions. Il existe plusieurs sous-ensembles fonctionnels de moyens comme par exemple des moyens de transport, des moyens de communication, etc.

VI.2.2.4 Les environnements

Un environnement désigne un lieu dans lequel sont placés des moyens et où des acteurs évoluent. Il dispose d'au moins une, voire plusieurs entrées/sorties. Il peut y avoir plusieurs environnements définis dans un exercice comme par exemple une ville, école, Bureau du préfet, Caserne des pompiers, etc. Un environnement est décrit de façon réaliste à travers un environnement visuel 3D (SIG 3D).

VI.2.2.5 Les sinistres

Un sinistre correspond à un aléa d'origine naturelle, technologique ou humain. C'est l'initiateur de la situation de crise par sa gravité ou son étendue géographique. Un sinistre est défini dans un scénario par :

- Un libellé,
- Un type,
- Une origine,
- Un état,
- Une étendue géographique,
- Un mode de propagation,
- Des actions.

Exemple de sinistre :

- Libellé : Une échappée de produits toxiques suite à un accident de la circulation.
- Type : TMD,
- Origine : technologique,
- État : en expansion,
- Étendue : Route du Tholonet à Aix et Quartiers du Tholonet,
- Mode : linéaire scénarisée.

Tous les éléments d'un scénario SIMFOR sont définis à l'aide de l'éditeur de scénario et enregistrés dans un fichier au format XML. La figure VI.1 montre la structure d'un fichier scénario de SIMFOR.

```
<?xml version="1.0" encoding="utf-8"?>
<scenario name="Scenario TMD Arles" xsi:noNamespaceSchemaLocation="Scenario.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <description>Scenario Transport matières dangereuses</description>
  <disasters>
    <FluidFlowDisaster name="Ecoulement Matière Dangereuse" label="Ecoulement de produits toxiques" origin="technologique" state="stop" location="Place Lamartine à Arles" dataPath=
    "Disaster/FluidFlow/FluidFlow_01" />
    <FireDisaster name="Incendie" label="Incendie" origin="technologique" state="stop" location="Place Lamartine à Arles" dataPath="Disaster/Fire/Fire_01">
      <position x="780" y="63" z="0" />
      <rotation h="0" p="0" r="0" />
      <CellCount x="133" y="137" />
      <CellSize x="1" y="1" />
    </FireDisaster>
  </disasters>
  <actors>
    <actor name="Codis" skill="medium" function="Piéton" mesh="Notable2/Notable2.xml">
      <position x="21.8" y="21.8" z="-99.9" />
      <phone name="Phone 1" />
      <fax name="Fax 1" />
      <radio name="Radio 1" />
      <collision mesh="ActorCollisionMesh.ive" />
    </actor>
    <actor name="Maire" skill="medium" function="Maire" mesh="Notable2/Notable2.xml">
      <position x="59" y="60" z="-99.9" />
      <phone name="Phone 2" />
      <fax name="Fax 2" />
      <radio name="Radio 2" />
      <collision mesh="ActorCollisionMesh.ive" />
    </actor>
    <actor name="Préfet" skill="medium" function="Préfet" mesh="Notable1/Notable1.xml">
      <position x="100" y="100" z="-99.9" />
      <phone name="Phone 3" />
      <fax name="Fax 3" />
      <radio name="Radio 3" />
      <collision mesh="ActorCollisionMesh.ive" />
    </actor>
    <actor name="Sous-Préfet" skill="medium" function="Sous-préfet" mesh="Notable1/Notable1.xml">
      <position x="40" y="40" z="-99.9" />
      <phone name="Phone 4" />
      <fax name="Fax 4" />
      <radio name="Radio 4" />
      <collision mesh="ActorCollisionMesh.ive" />
    </actor>
```

Figure VI.1 – Structure d'un scénario SIMFOR.

VI.3 Présentation du scénario TMD

Le scénario TMD a été élaboré avec l'aide d'un expert de la gestion de crise et a pour but de sensibiliser les intervenants aux différentes conséquences que peut avoir un accident TMD.

Objectif du scénario

Formation au risque technologique TMD d'un niveau ORSEC avec la mise en pratique des différents plans de gestion de crise : Dispositions spécifiques "Transport de matières dangereuses" du plan ORSEC, Cellule municipale de crise, en particulier pour les acteurs humains suivants : CODIS, Préfet, Sous-Préfet, et Maire.

Quand

Un jour d'école, en matinée, vers 10h du matin. Ciel clair et dégagé, temps beau, ensoleillé.

Synopsis

En périphérie d'Arles, un camion TMD s'est renversé suite à un accident de la circulation sur le rond-point Lamartine en direction du boulevard Émile Combes (N 2570), à moins de 150 m du collège F. MISTRAL, à proximité du Rhône et de la voie ferrée. La citerne est endommagée et de l'hydrocarbure se répand sur la chaussée. Un témoin de l'accident a donné l'alerte en appelant le 18 (CODIS), la police/gendarmerie bloque les points d'accès au niveau du rond-point Lamartine.

VI.3.1 Fonctionnement général

Pour démarrer l'exercice, il faut tout d'abord lancer un serveur SIMFOR avec le scénario de l'exercice. Une fois le serveur lancé, les joueurs (apprenants) se connectent au serveur, soit par internet ou en réseau local (pour nos tests, l'exercice a été réalisé en réseau local mais les apprenants étaient dans des pièces différentes). Une fois que tous les apprenants sont connectés, la partie peut commencer. Pour le scénario TMD nous avons quatre apprenants et sept AgJ comme illustré dans la figure VI.2.

L'expert du domaine définit au préalable (via l'éditeur du modèle du domaine) les les missions relatives à chaque rôle ainsi que la scénarisation de l'exercice et cela via

Les acteurs

N°	Nom	Type	3D	Icône
1	Unité de Gendarmerie	GA		GA Gendarme
2	CODIS	JOUEUR	✓	PL CODIS 3D
3	Pompier n°1 (COS 1)	GA	✓	GA Pompier 1 3D
4	Officier Chef de Groupe (COS 2)	GA		GA Officier
5	Préfet (DOS)	JOUEUR	✓	PL Préfet 3D
6	Sous-préfet	JOUEUR	✓	PL Sous-préfet 3D
7	Maire	JOUEUR	✓	PL Maire 3D
8	Directeur du collège F. Mistral	GA		GA Directeur
9	Témoin	GA	✓	GA Témoin 3D
10	Média	GA		GA Média
11	Pompier n°2	GA	✓	GA Pompier 2 3D

Figure VI.2 – Liste des acteurs pour le scénario TMD

des scriptes qui permettent de déclencher des événements sous certaines conditions comme par exemple l'inflammation de l'hydrocarbure si les premières mesures ne sont pas prises. La figure VI.3 liste ces événements sur l'échelle du temps avec l'enchainement des actions à réaliser par les acteurs. Par exemple l'action A1 va correspondre à l'appel du CODIS au pompier pour le prévenir de l'accident.

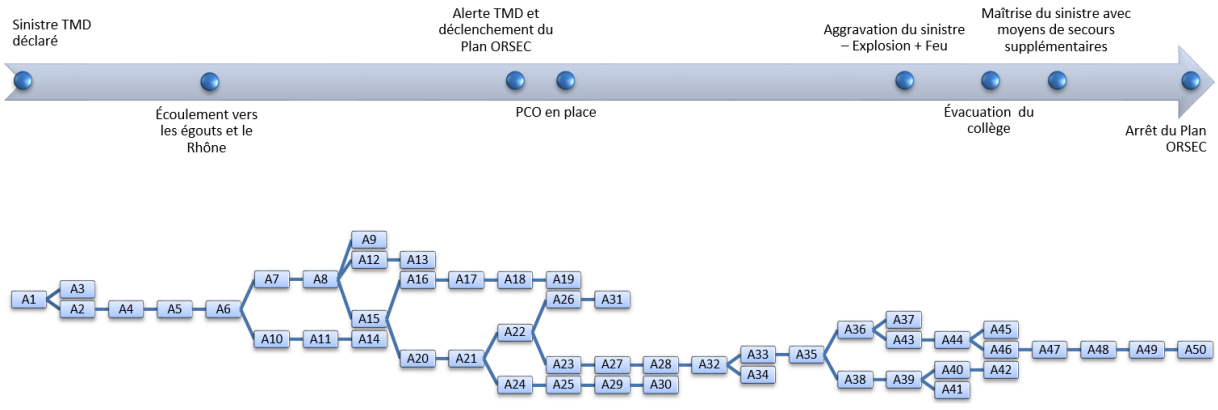


Figure VI.3 – Enchainement des événements et des actions du scénario TMD

Le premier événement du scénario concerne l'écoulement de l'hydrocarbure vers les égouts et qui provoquera par la suite une pollution du Rhône. Cet événement sera déclenché si le pompier qui arrive sur les lieux ne prend pas les bonnes mesures (mise en place de merlons de sable pour contenir l'écoulement par exemple).

VI.3.2 Déroulement du scénario

Nous allons résumer le déroulement du scénario en quelques lignes : tout d'abord au début de la partie, le CODIS envoi un pompier sur le lieu de l'accident pour interroger le témoin et recueillir les informations sur l'accident. Une fois les informations transmises au CODIS et que celles-ci confirment l'accident TMD, le CODIS prévient l'officier pompier et rédige une fiche de renseignements sur le sinistre et l'envoie par fax à l'officier pompier, au maire, au préfet et au sous-préfet. L'officier pompier quant à lui doit envoyer un second pompier sur les lieux avec un fourgon pompe-tonne (FPT) et donner les instructions de première mesure au pompier sur les lieux. Une fois la fiche de renseignement reçue, le préfet s'entretient avec le sous-préfet et ils doivent se mettre d'accord sur le lieu du poste de commandement avancé (PCO) et du déclenchement du plan ORSEC (rempli et diffusé via fax par le préfet). Les acteurs de la sécurité civile doivent rendre compte de leurs actions via d'outil main courante après chaque action importante. Le sous-préfet se charge par la suite de prévenir le CODIS et la gendarmerie du lieu du PCO avant de s'y rendre, et de s'entretenir avec le maire de la ville pour repérer d'éventuels risques autour du sinistre (évacuation d'école, zone à risque, etc.). L'officier pompier utilise le camion PCO pour se rendre au PCO, l'officier pompier prend note de l'avancement de la situation (avec les pompiers sur les lieux) et rend compte au sous-préfet au PCO. Selon la gravité de la situation (pollution, inflammation de l'hydrocarbure si les premières mesure n'ont pas été prise), le sous-préfet demande de diffuser un communiqué de presse aux média et des moyens supplémentaires si besoins. Si la situation l'exige, le sous-préfet doit contacter le maire pour évacuer les lieux à proximité du sinistre comme le collège. Une fois le sinistre maitrisé le préfet envoi l'arrêté préfectoral de levée du plan ORSEC au CODIS, maire et gendarme.

VI.4 Validation sur le Scénario TMD

Dans cette section, nous présentons la mise en œuvre du scénario TMD à travers l'outil SIMFOR. Nous présentons tout d'abord un protocole de validation afin de valider notre cadre méthodologique, nous présentons ensuite les résultats obtenus ainsi qu'une discussion autour de ces résultats.

VI.4.1 Protocole de validation

Pour une première validation (scénario décrit ci-dessus), nous avons sélectionné deux groupes de personne constitués chacun de quatre apprenants situés dans différentes pièces (figure VI.4). Ces deux groupes d'apprenants sont constitués de doctorants

volontaires qui n'ont aucune connaissance de la gestion de crise. Cette expérimentation vise à valider deux points : Le premier est la validation de l'évaluation (individuelle et collective) automatique fournie par le système, le deuxième point est la progression des apprenants en utilisant l'outil SIMFOR.

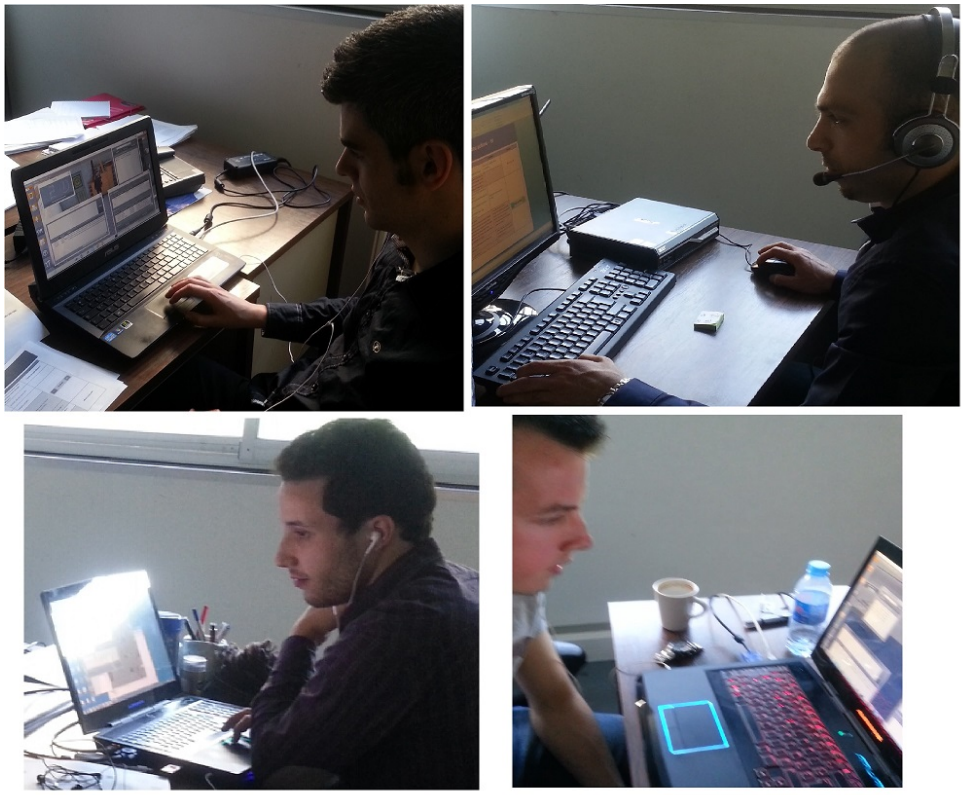
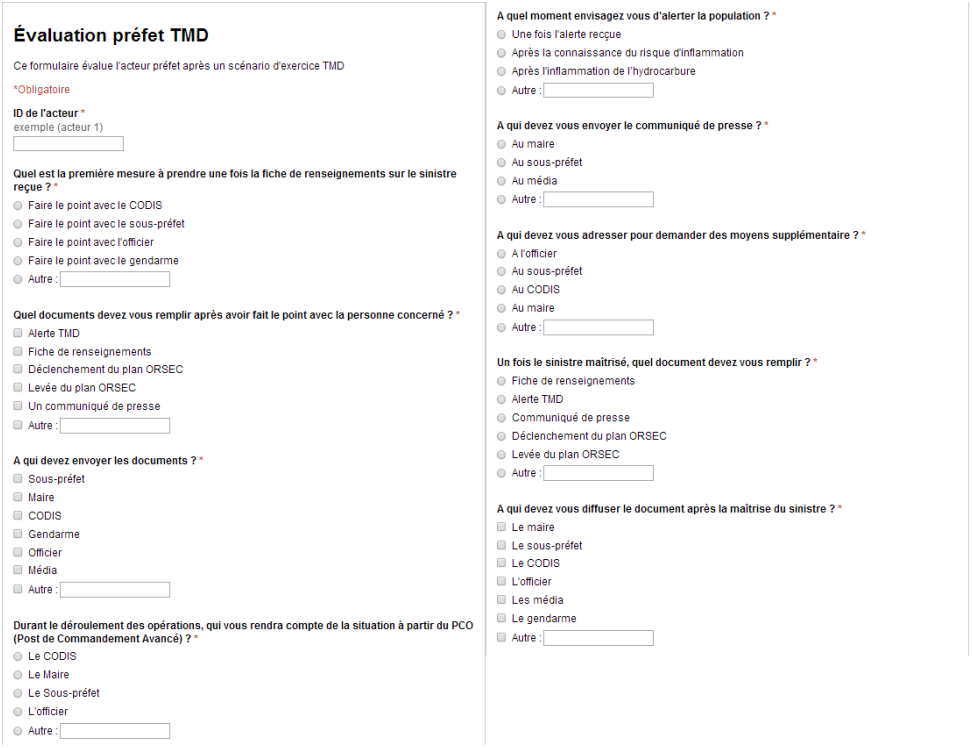


Figure VI.4 – Les participants de l'expérimentation pilote.

Pour valider le premier point (l'évaluation automatique qui vise essentiellement l'espace comportemental), nous proposons aux apprenants de répondre à un formulaire qui contient des questions sur les objectifs d'apprentissage du scénario (un formulaire est attribué à chaque rôle du scénario visant à évaluer son apprentissage) à la fin chaque exercice. Un exemple de formulaire est présenté par la figure VI.5 (formulaire liée aux objectifs d'apprentissage du rôle Préfet). Chaque question permet d'évaluer un objectif d'apprentissage défini au préalable par l'expert du domaine. Une question peut avoir des réponses multiples et le résultat est obtenu par un pourcentage de réponses justes.

Pour valider le second point (progression des apprenants), nous réalisons deux séries d'exercices et nous comparons les résultats du premier et du second exercice pour mesurer l'écart de performance. Cet écart peut être constaté à la fois par le biais de l'évaluation automatique (fournie par le système) ainsi que par les résultats des formulaires.

Avant le début de l'exercice, les apprenants ont eu droit à une courte présentation du concept de la gestion de crise et du scénario TMD (scénario, objectifs, etc.) ainsi



Évaluation préfet TMD

Ce formulaire évalue l'acteur préfet après un scénario d'exercice TMD

*Obligatoire

ID de l'acteur *
exemple (acteur 1)

Quel est la première mesure à prendre une fois la fiche de renseignements sur le sinistre reçue ? *
- Faire le point avec le CODIS
- Faire le point avec le sous-préfet
- Faire le point avec l'officier
- Faire le point avec le gendarme
- Autre :

Quel documents devez vous remplir après avoir fait le point avec la personne concerné ? *
- Alerte TMD
- Fiche de renseignements
- Déclenchement du plan ORSEC
- Levée du plan ORSEC
- Un communiqué de presse
- Autre :

A qui devez envoyer les documents ? *
- Sous-préfet
- Maire
- CODIS
- Gendarme
- Officier
- Média
- Autre :

Durant le déroulement des opérations, qui vous rendra compte de la situation à partir du PCO (Post de Commandement Avancé) ? *
- Le CODIS
- Le Maire
- Le Sous-préfet
- L'officier
- Autre :

A quel moment envisagez vous d'alerter la population ? *
- Une fois l'alerte reçcue
- Après la connaissance du risque d'inflammation
- Après l'inflammation de l'hydrocarbure
- Autre :

A qui devez vous envoyer le communiqué de presse ? *
- Au maire
- Au sous-préfet
- Au média
- Autre :

A qui devez vous adresser pour demander des moyens supplémentaire ? *
- A l'officier
- Au sous-préfet
- Au CODIS
- Au maire
- Autre :

Un fois le sinistre maîtrisé, quel document devez vous remplir ? *
- Fiche de renseignements
- Alerte TMD
- Communiqué de presse
- Déclenchement du plan ORSEC
- Levée du plan ORSEC
- Autre :

A qui devez vous diffuser le document après la maîtrise du sinistre ? *
- Le maire
- Le sous-préfet
- Le CODIS
- L'officier
- Les média
- Le gendarme
- Autre :

Figure VI.5 – Exemple de formulaire pour évaluer l'apprenant incarnant le rôle du préfet.

qu'une initiation à l'outil SIMFOR (comment se déplacer, envoyer des fax, etc).

VI.4.2 Résultats et discussions

Tous les résultats d'évaluation et les rétroactions faites aux apprenants sont enregistrés dans fichier résultat au format XML. Le fichier résultat contient pour chaque apprenant :

- Liste des actions réalisées.
- Résultat d'évaluation pour chaque action réalisée.
- Historique des rétroactions apportées à l'apprenant.
- Évaluation de chaque mission réalisée.
- Le score social de l'apprenant ainsi que le score final individuel de l'apprenant.

En plus des évaluations individuelles, le fichier résultat contient l'historique de l'ensemble des interactions entre les différents acteurs, ce qui permet de calculer la force d'interaction, le graphe de différence et ainsi calculer le score social et produire une évaluation globale et collective (avec l'intégration d'autres indicateurs globaux selon le type de scénario). De plus le fichier résultat fournit d'autres indicateurs de nature purement observable comme par exemple l'historique du couplage globale de l'organisation (on peut déduire les moments clés de l'exercice grâce aux pics de couplage), ou l'enchainement des actions sur l'échelle du temps.

Pour la validation du premier point qui concerne la validation de l'évaluation fournie par le système, les résultats obtenus sont illustrés dans le tableau VI.1.

Tableau VI.1 – Résultats du scénario d'exercice TMD.

Apprenants	Exercice 1 individuel	Exercice 1 social	Exercice 1 missions	Temps	Global	Exercice 2 individuel	Exercice 2 social	Exercice 2 missions	Temps	Global
CODIS	0.817268	0.804907	0.82963			0.853495	0.845879	0.861111		
Maire	0.943310	0.882632	1			0.957254	0.914508	1		
Préfet	0755899	0.865317	0.646481			0.763133	0.852076	0.675990		
Sous-Préfet	0.763071	0.88662	0.639522			0.783950	0.889330	0.678570		
Temps				31(m)0.65					22(m)0.76	
Globale		0.859869	0.778090		0.762653		0.875448	0.803918		0.813122

Le tableau VI.1 montre que tous les apprenants ont amélioré leurs scores entre le premier et le second exercice. Les scores affichés en bleu représentent le score individuel final (obtenu avec la moyenne entre le score social individuel et le score individuel des missions). Les scores affichés en vert représentent les scores collectifs : le score social collectif (moyenne des scores social individuels), le score comportemental collectif (moyenne des scores individuels de mission), et le score physique collectif. Le score physique collectif est calculé par rapport à la zone sinistrée durant la session de jeu. Cette zone grandit tant que le sinistre n'est pas maitrisé (simulation d'incendie urbain). Les scores affichés en rouge représentent la performance globale du groupe, et sont obtenus par la moyenne des scores collectifs social, comportemental, et physique. Tous les scores calculés sont normalisés pour une valeur maximale de 1 (action, mission, social, physique, individuelle et collective).

Pour la validation du deuxième point qui concerne la progression des apprenants à travers l'outil SIMFOR, les résultats des formulaires ainsi que les scores obtenus par le biais l'évaluation automatique sont illustrés dans le tableau VI.2.

Tableau VI.2 – Résultats des formulaires du scénario d'exercice TMD.

Apprenants	Exercice 1 réponse au formulaire	Exercice 1 score mission-score individuel	Exercice 2 réponse au formulaire	Exercice 2 score mission-score individuel
CODIS	60%	0.82963 - 0.817268	91%	0.861111 - 0.853495
Maire	100%	1 - 0.943310	100%	1 - 0.957254
Préfet	81%	0.646481 - 0.755899	81%	0.675990 - 0.763133
Sous-Préfet	75%	0.639522 - 0.763071	87%	0.678570 - 0.783950

Le tableau VI.2 illustre les résultats des formulaires comparés aux résultats de l'évaluation automatique présentés dans le tableau VI.1. Premièrement, nous remarquons que tous les apprenants ont amélioré leur performance (évaluation automatique et formulaires). Deuxièmement, nous remarquons une cohérence entre les scores obtenus par l'évaluation automatique et le résultat des formulaires, à l'exception des résultats du Préfet et du Sous-Préfet. Cet écart peut s'expliquer aux nombreuses et longues procédures à réaliser dans le cadre du scénario TMD par le Préfet et Sous-Préfet, ce qui

peut introduire du bruit dans l'évaluation (action en plus, temps d'attente, etc.). Néanmoins, l'association du score des missions avec le score social (évaluation collective) rend l'évaluation automatique plus précise.

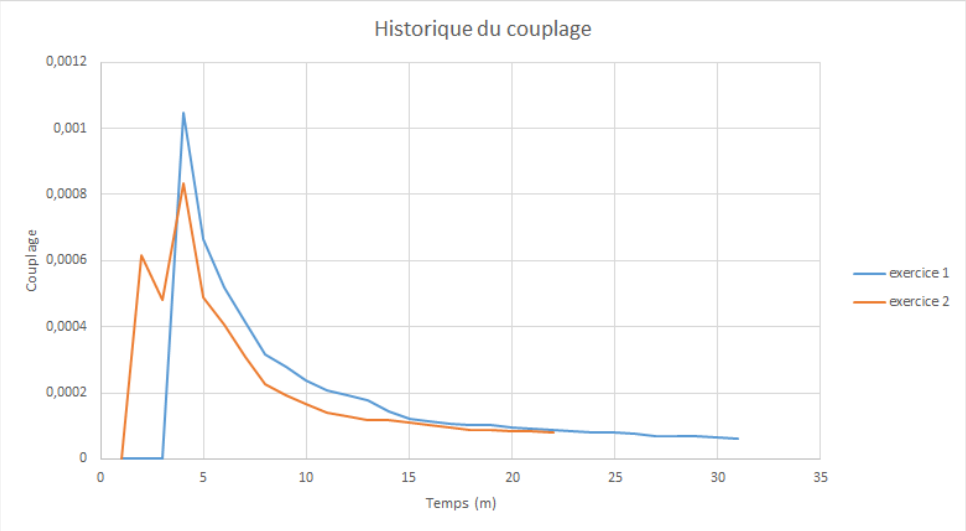


Figure VI.6 – Historique du couplage du réseau pour le premier et le deuxième exercice.

Durant notre expérimentation, nous avons désactivé le support à l'apprenant (case à cocher dans le serveur SIMFOR), représenté par l'agent pédagogique, et ceci afin de ne pas biaiser les résultats de l'évaluation. Avec la pertinence de l'évaluation automatique (fournit par le système), le support (aide) de l'apprenant sera plus efficace et l'apprentissage sera meilleur. Comme nous l'avons mentionné dans la chapitre V, le système fournit désormais de nombreux outils qui permettent de visualiser des phénomènes observables comme l'historique du couplage (pique de communication, figure VI.6), historique des actions (figure VI.7), ou encore le couplage du réseau (figure III.6).

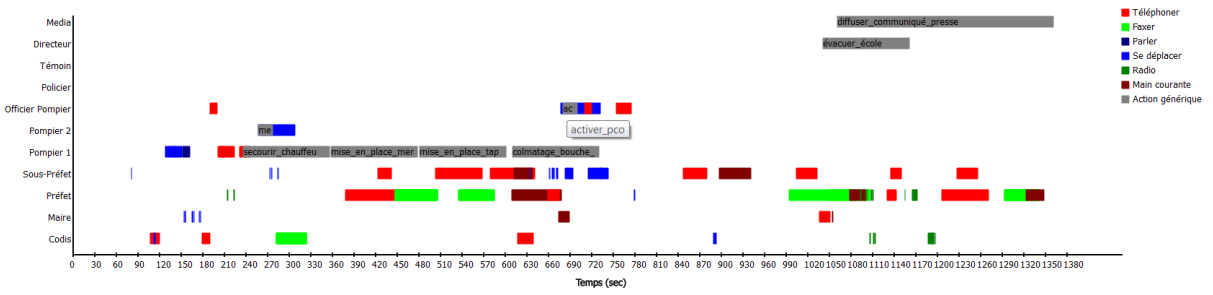


Figure VI.7 – Séquence d'action durant l'exercice TMD.

Grâce au moniteur d'évaluation (section V.7.2), nous pouvons aussi voir l'évaluation des apprenants en détail, mission par mission (ainsi que les actions composant la mission). Ainsi l'apprenant peut avoir un "*débriefing*" avec le moniteur pour analyser et combler ses lacunes.

Le retour des apprenants était très positif par rapport à l'outil SIMFOR, et ont apprécié l'aspect ludique et collaboratif de l'outil. La plupart des apprenants ont répondu qu'ils se sont parfaitement immergés dans le rôle incarné et qu'ils ont une idée plus précise de la gestion de crise après avoir participé aux scénarios d'exercice TMD.

VI.5 Conclusion

Dans ce chapitre nous avons présenté un scénario d'exercice de gestion de crise pour illustrer et valider notre cadre méthodologique pour l'évaluation individuelle et collective des apprenants dans un jeu sérieux collaboratif ainsi que l'architecture proposée illustrée avec le JS SIMFOR. Le scénario d'exercice a été élaboré avec l'aide d'un expert du domaine et porte sur un accident de transport de matières dangereuses (TMD). Comme nous l'avons vu, le scénario TMD est très intéressant, car il implique plusieurs corps de métiers. De plus le scénario TMD peut évoluer en un scénario de pollution environnemental, ou d'incendie de grande ampleur. Pour une première expérimentation pilote, nous avons défini un protocole de validation qui consiste à valider deux points : Le premier est la validation de l'évaluation (individuelle et collective) automatique fournie par le système. Le deuxième, est la progression des apprenants en utilisant l'outil SIMFOR.

La première expérimentation pilote réalisée a consisté à faire deux séries d'exercice avec deux groupes de quatre apprenants (doctorants volontaires au sein du laboratoire) jouant le rôle du CODIS, Préfet, Sous-Préfet, et Maire. Les résultats obtenus ont montré une amélioration des performances des apprenants ainsi que la cohérence des évaluations fournies par le système par rapport aux réponses des apprenants aux questions après les exercices (formulaires). Néanmoins, cette validation reste partielle (validation des processus de collaborations, d'évaluations, etc), et il faudra réaliser d'autres expérimentations avec des apprenants impliqués dans la gestion de crise (Maire, Préfet, pompier, etc).

Conclusion et perspectives

S'inscrivant dans le contexte des jeux sérieux, le travail de recherche présenté dans cette thèse a porté sur l'évaluation individuelle et collective des apprenants dans les jeux sérieux collaboratifs abordés avec un regard orienté-agent. Avant de résumer la démarche suivie, ainsi que les contributions proposées, rappelons le contexte de notre travail de recherche : nous nous sommes intéressés aux besoins d'évaluation dans les jeux sérieux visant à la formation de procédures collectives, caractérisée par l'hétérogénéité des compétences des différents rôles joués soit par des joueurs humains, soit par leur équivalent artificiel.

Cet objectif posé en termes d'évaluation des apprenants ainsi que de simulation de comportements, a été suivi avec une volonté de généricité dans la conception, notamment en vue de rendre possible l'amélioration des capacités d'évaluation dans des sérieux pré-existants.

Démarche

Afin de tenter d'apporter une réponse à la problématique de l'évaluation des apprenants dans les jeux sérieux, nous avons caractérisé les besoins et les spécificités de l'évaluation des apprenants dans les jeux sérieux.

Nous avons tout d'abord noté qu'il existait plusieurs types d'évaluation des apprenants. Nous avons classé ces différents types dévaluation selon deux perspectives : *individuelle* vs *collective* et *temps réel* vs *finale*. L'é*valuation dans le temps* (avant, pendant ou après l'apprentissage) constitue l'objectif général de l'évaluation (situer le niveau de l'apprenant, aider l'apprenant ou certifier son apprentissage). L'*évaluation individuelle et collective* représente le type d'apprentissage dans le jeu sérieux (individuel ou colla-

boratif). Cet apprentissage peut être individuel, afin d'acquérir une ou des compétences dans un certain domaine, ou bien collectif dans le but d'acquérir des aptitudes sociales (collaboration). Dans l'apprentissage collaboratif, l'évaluation est plus complexe, car elle doit prendre en compte les actions individuelles des apprenants mais également le résultat de leurs interactions (actions collectives), surtout si l'objectif d'apprentissage diffère d'un apprenant à un autre.

Nous avons par la suite défini les besoins d'évaluation des apprenants dans les jeux sérieux, en soulignant la place de la représentation des connaissances et des traces des apprenants. Dans un contexte d'évaluation de compétences hétérogènes, on peut s'attendre à des besoins de représentation variés, nécessitant des capacités de manipulation adaptées. Il peut alors être nécessaire de recourir à plusieurs modèles de représentation de connaissances au sein d'un même environnement en fonction des besoins d'évaluation. Le recueil de traces représente le processus qui permet de récupérer l'activité de l'apprenant dans l'environnement virtuel. Ces traces seront exploitées pour des fins d'évaluation. Cette évaluation peut servir aussi à fournir un support à l'apprenant, pour qu'il puisse visualiser son activité.

Dans le contexte de jeux sérieux collaboratif et des besoins spécifiques de l'évaluation individuelle et collective des apprenants, nous proposons un cadre méthodologique basé sur le concept d'espace d'évaluation. Le cadre est ensuite opérationnalisé par une architecture multi-agents. Nous avons illustré la mise en œuvre de ce cadre méthodologique dans le domaine de la gestion de crise, en montrant bien le caractère distribué et hétérogène de l'évaluation.

Contributions

La volonté de généricité dans notre démarche a impliqué dans un premier temps d'établir les besoins de représentation de connaissances, d'informations ou de données, produites ou déduites des actions des joueurs. Si le domaine d'application de la gestion de crise a confirmé le caractère hétérogène de ces informations, il est fort probable que d'autres problèmes présentent également des besoins nouveaux. Aussi avons nous cherché a définir un cadre d'analyse le plus générique possible, et bien qu'elle n'apparaisse explicitement qu'au niveau architectural, la modélisation orienté agents a guidé notre travail en séparant ce qui relève purement de problèmes d'hétérogénéité logicielle, de ce qu'on pourrait qualifier de compétences métiers voire pédagogiques.

Ces compétences doivent être exprimées en termes de critères qualitatifs et quantitatifs, ainsi que leur niveau de contribution dans la décision finale de valider ou non l'acquisition de ces compétences. L'organisation des informations, entrant dans le rensei-

gnement de ces critères, en groupes homogènes relève d'une démarche d'agentification par le regroupement des primitives dédiées (autonomes et indépendantes) manipulant ces informations et leur contexte et ensuite une combinaison (forme d'interaction) de ces critères pour produire une évaluation.

Notre première contribution a ainsi tenu compte de ce caractère hétérogène et distribué avec le concept d'*Espace d'Evaluation* (*ES*). Un *ES* englobe tous les éléments nécessaires pour produire des évaluations, en considérant un scénario de jeu à travers des vues différentes, chacune correspondant à un objectif d'évaluation particulier. Un *ES* se défini comme un quadruplet représenté par : une représentation des connaissances ***Kw***, un ensemble d'indicateur ***I***, un ensemble de métriques ***M***, et enfin un ensemble de modèle d'évaluation ***AM***.

La représentation des connaissances ***Kw*** permet de formaliser la manière dont les connaissances sont représentées afin de proposer un gain de généricité dans l'espace d'évaluation. Les indicateurs ***I*** contribuent ou guident un processus d'évaluation renvoyé au moniteur du jeu ou aux joueurs eux-mêmes. La métrique ***M*** est associée à la notion de distance, qui peut se reformuler en terme de proximité, et permet de quantifier les indicateurs. Un modèle d'évaluation ***(AM)*** peut être vu comme une fonction utilitaire qui produit une évaluation (score) à partir d'indicateurs ***I*** quantifiés par des métriques ***(M)*** dans un sous ensemble de connaissances exprimé selon un mode de représentation ***(Kw)***. Ainsi, un même ensemble de connaissance peut se voir appliquer des modèles d'évaluation et des métriques différentes pour répondre à des questions d'évaluation différentes.

Nous avons par la suite illustré le concept d'espace d'évaluation dans le domaine de la gestion de crise, caractérisée par un apprentissage collaboratif et hétérogène. Pour appréhender le domaine de la gestion de crise, nous avons défini trois espaces d'évaluation : un *espace Physique* pour représenter l'environnement virtuel (dans lequel évoluent les différents apprenants), un *espace Comportemental* pour représenter l'aspect procédural de la gestion de crise, et enfin un *espace Social* pour représenter l'aspect collaboratif de la gestion de crise.

Notre deuxième contribution a consisté à mettre en œuvre le concept d'espace d'évaluation dans un jeu sérieux. Pour cela, nous avons proposé une architecture multi-agent qui reprend différents modules d'un STI. Cette architecture est chargée de : collecter les traces de l'apprenant, de produire une évaluation (individuelle et collective), et enfin d'apporter un support à l'apprenant durant son apprentissage.

Pour implémenter ce système, que nous avons souhaité garder le plus "*générique*" possible mais pas seulement à un niveau théorique, nous nous sommes basés sur le projet SIMFOR, un jeu sérieux pour la formation de la gestion de crise auquel manquait justement une aide à l'évaluation des compétences. Dans le cadre de ce projet, nous

avons développé plusieurs outils d'édition et de suivi pour faciliter le suivi et l'évaluation des apprenants par les concepteurs (de scénario de gestion de crise).

Grâce à cette évolution du jeu sérieux SIMFOR, nous avons pu réaliser quelques expérimentations et valider (de façon partielle) notre cadre méthodologique pour l'évaluation individuelle et collective des apprenants. Ainsi, nous avons défini un protocole de validation qui a pour objectif de valider deux points : Le premier est la validation de l'évaluation (individuelle et collective) automatique fournit par le système, le deuxième point est la progression des apprenants en utilisant l'outil SIMFOR. Ce protocole à été réalisé dans le cadre d'un scénario de gestion de crise en l'occurrence un scénario TMD (Transport de Matières Dangereuses).

Limites

Pour valider notre cadre méthodologique, nous avons défini un protocole de validation. Ce protocole présente certaines limites d'un point de vue applicatif, car ce dernier implique des participants (apprenants) qui ne sont pas des intervenants réels de la gestion de crise et ne peut valider notre approche que de façon partielle. Néanmoins, ces expérimentations ont déjà permis d'assurer l'apprentissage de tâches collaboratives ainsi que les différents mécanismes de l'évaluation des apprenant (individuelle et collective).

Si les premiers résultats de validation sont encourageants, nous sommes conscients que le temps imparti par une thèse n'a pas permis de traiter toutes les questions soulevées par nos propositions. Ces questions se situent à deux niveaux : un niveau méthodologique et un niveau architectural. Ces questions ouvrent ainsi des perspectives intéressantes qui seront abordées dans la section suivante.

D'un point de vue méthodologique, une validation complète de l'approche requiert de confirmer plus complètement l'adéquation des indicateurs et leurs pondération en effectuant par exemple, des mesures de sensibilité de ces indicateurs et scores obtenus, soit en d'autres termes un calibrage des indicateurs.

Une autre limitation méthodologique, en relation avec des préoccupations de type STI (Systèmes Tutoriels Intelligents), concerne l'évaluation des connaissances déclaratives. En effet, même si le cadre méthodologique le permet, le cas d'application portant davantage sur des connaissances procédurales, nous n'avons pas abordé totalement la construction de modèle de connaissances des joueurs tel qu'on peut l'aborder dans les Environnements Informatiques d'Apprentissage Humain. Ainsi, l'ontologie de domaine a surtout permis de conceptualiser les connaissances du domaine mais n'a pas été employé pour valider les connaissances déclaratives des apprenants en appliquant par exemple des méthodes de type "*Overlay*". La généralisation du cadre méthodolo-

gique au travers d'autres expérimentations (ie jeux sérieux avec d'autres objectifs de formation) permettra de répondre à ce nouveau besoin.

Enfin, la prise en compte des besoins de scénarisation propres aux jeux sérieux doit encore être adressée de manière cohérente avec le cadre méthodologique, par exemple en s'appuyant les enchaînements d'actions et interactions des joueurs qui renverraient vers une classification pré définie des actions en lien avec des Espaces d'Évaluation.

D'un point de vue architectural, l'outil SIMFOR permet une approche ludique et immersive pour apprendre la gestion de crise, malheureusement SIMFOR présente quelques limites logicielles caractérisées par la manipulation de certains outils qui ne reflètent pas exactement ce qui ce passe en conditions réelles (fax, radio, etc.). Aussi, SIMFOR ne permet pas de retransmettre la pression exercée (contraintes techniques des matériels employés, stress temporel...) lors d'une crise majeur. Néanmoins, l'outil SIMFOR peut être une première étape d'initiation à la gestion de crise, et il permet de se familiariser avec les différentes procédures du plan ORSEC, et permettre ainsi à des exercices réels complémentaires de mieux cibler les objectifs de formation (afin de réduire les coûts et gagner du temps).

Par ailleurs, le prototype développé doit encore être confronté à une expérimentation plus poussée afin de vérifier, et éventuellement corriger, ses performances (augmentation du nombre de joueurs) ainsi que sa généricité au travers d'expérimentations complémentaires notamment sur d'autres cas d'application.

Perspectives

Le travail réalisé jusqu'ici a permis d'établir les contributions citées précédemment, ainsi qu'un début de validation expérimentale de la pertinence de notre démarche. Les perspectives de recherche que nous dégageons de tout ce travail peuvent se résumer en une poursuite de cette volonté de validation du cadre méthodologique proposé et de l'architecture logicielle associée, ainsi qu'en la réponse aux limites identifiées précédemment.

Dans un contexte expérimental et pour enrichir notre protocole de validation, nous souhaitons réaliser d'autres scénarios d'exercice pour la formation à la gestion de crise, comme par exemple des scénarii de tremblement de terre ou d'inondation avec des intervenants de gestion de crise comme le Maire, le Préfet, etc. Ces expérimentations permettront non seulement de récolter d'avantage de données (pour valider et affiner notre cadre méthodologique) mais aussi pour ajouter d'autres fonctionnalités au JS SIMFOR (actions supplémentaires, animations, simulation, etc).

Dans un contexte de validation plus complet de notre cadre méthodologique, ainsi

que pour répondre aux limites conceptuelles déjà identifiées, nous souhaitons appliquer notre cadre méthodologique (le concept d'espace d'évaluation) sur d'autres jeux collaboratifs afin valider (ou de consolider) l'aspect générique de notre approche. Dans cette optique, nous allons mettre en œuvre notre approche dans le cadre d'un autre environnement de formation qui permet d'incarner un agent sanitaire pour la lutte contre les rongeurs [Chadli et al., 2010]. Si la dimension collective est plutôt absente dans ce cas d'application, on retrouve cependant la nécessité d'aborder une évaluation au travers de différents points de vue, chacun représentant un type de compétence à certifier (capacité à identifier les informations pertinentes d'une campagne de lutte contre les rongeurs, respect des procédures, capacité d'évaluer correctement l'ampleur d'une prolifération). Chaque compétence se base sur des informations et traces spécifiques, confirmant l'intérêt d'un cadre méthodologique basé sur des Espaces d'Evaluation.

Dans un contexte d'amélioration de l'apprentissage (à travers notre cadre méthodologique et de l'architecture associée), nous souhaitons enrichir les stratégies d'intervention de l'agent pédagogique avec les différents points de vue de l'évaluation, par exemple utiliser l'évaluation collective (graphe social). Dans ce cas nous aurons des interventions spécifiques à chaque espace d'évaluation.

Une perspective ambitieuse à plus long terme, consiste à passer d'un cadre méthodologique à une véritable méthodologie de conception de Jeux Sérieux impliquant des compétences hétérogènes. Ceci requiert de compléter notre cadre méthodologique par :

- *La définition d'un langage de description générique des besoins d'information et d'évaluation* : le concept d'Espace d'Évaluation constitue déjà en soit une grille d'analyse qui peut aider à caractériser les besoins d'informations et d'évaluation, mais un niveau d'abstraction supplémentaire permettrait de guider encore mieux cette phase d'expression des besoins. Le rôle des ontologies de domaine notamment s'en trouverait renforcé. Cela aboutirait également à une bibliothèque d'agents pré-définis auxquels ne manqueraient plus qu'une phase (non triviale) de paramétrage. L'application sur d'autre cas d'application peut répondre à cette limite.
- *La définition d'étapes de conception, chacune associée à un langage de description et une notation graphique facilitant le travail des concepteurs* : dont l'objectif final serait justement le paramétrage des agents directement exploitable par l'architecture logicielle proposée.

Si l'on considère ce travail davantage d'un point de vue modélisation et architecture multi-agents, une extension du travail déjà réalisé pourrait consister en l'ajout de stratégies de négociation multi-agents pour l'évaluation et la validation de l'apprentissage comme dans [Chadli et al., 2010]. De fait, l'architecture s'y prête naturellement et intégrait dés le départ cette possibilité. Toujours au niveau architecture multi-agents,

l'objectif de généricité peut se traduire à deux niveaux :

- *Définition d'un langage de manipulation de données :* ce langage de manipulation de données permettra un gain de généricité afin d'aboutir à une meilleurs interopérabilité entre les différents modules (agents). Ainsi à l'image des travaux dans [Serment, 2007], le travail d'évolution d'un jeu sérieux vers un environnement de formation pourrait se limiter au développement des interfaces logicielles entre les sources de données du jeu sérieux et notre infrastructure.
- *Définition d'une bibliothèque d'agent :* cette bibliothèque a déjà été défini de manière partielle (pour le projet SIMFOR), et sa généralisation faciliterait encore la conception d'un jeu sérieux en réduisant l'adaptation à un problème (non trivial malgré tout) de paramétrage d'agents sélectionnés en fonction de critères issus du cadre applicatif (domaine d'application et objectifs du jeu).

Au travers de ces perspectives, se dessine un champs d'étude intéressant dépassant le seul périmètre de cette thèse.

Annexes

ANNEXE A

Les moteurs de jeux vidéo

Sommaire

A.1 Introduction 155
A.2 Qu'est qu'un moteur de jeu ? 156
A.3 Quelque exemple de moteur de jeu 156
A.3.1 Delta 3D 157
A.3.2 Unreal engine 158
A.3.3 CryEngine 159
A.3.4 Unity 3D 160
A.4 Conclusion 161

A.1 Introduction

Aujourd'hui, la seule façon d'avoir une simulation réaliste et rapide, avec un rendu (graphismes) sophistiqué est un PC avec des outils issus de jeux vidéo. La réalité virtuelle (RV), la réalité augmentée (RA), et la simulation physique de haute-fidélité ont longtemps posé une trop forte barrière pour tout chercheur. Les outils de simulation de haute-fidélité (graphiques et physique) et le rendu réaliste (graphisme) sont restés longtemps un marché de niche, cher et en grande partie réservés aux laboratoires de RV (avec des périphériques coûteux tels que des gants, casques de RV, etc). Durant les dernières années, la situation a considérablement changé. Maintenant, le marché des jeux vidéo engendre des revenus encore plus importants que l'industrie du cinéma, et exploite le meilleur du matériel informatique avec des outils à la pointe de la technologie, comme la bibliothèque DirectX 11 de Microsoft (tessellation, DirectCompute, amélioration du multi-threading, etc.), le moteur physique de nVidia, ou encore le TressFX

d'AMD (rendu des cheveux), etc. Les rendus les plus sophistiqués ne se trouvent pas maintenant sur les machines scientifiques spécialisés, mais sur PC de bureau (avec une bonne carte graphique) grâce aux moteurs de jeux vidéo [Lewis et Jacobson, 2002].

A.2 Qu'est qu'un moteur de jeu ?

Avec la structure modulaire des jeux d'aujourd'hui, le moteur de jeu fait référence à une collection de modules qui ne spécifie pas directement le comportement (logique de jeu) ou l'environnement du jeu (niveau de données). Un moteur de jeu comprend en général des modules de gestion des entrées, de sorties (rendu 3D, dessin 2D, son), et de la physique/dynamique génériques du monde virtuel [Lewis et Jacobson, 2002].

Dans le milieu des années 90, le jeu Doom était l'un des jeux les plus populaire à tel point qu'il a créé une catégorie de jeu qui est le FPS (First Person Shooter, jeu de tir à la première personne), à l'époque les FPS prenaient plus l'appellation Doom-like (du même genre que le jeu Doom). Les développeurs de Doom (id software) ont été les premiers à séparer le cœur du jeu (tels que le rendu 3D, le système de détection de collision, ou le système audio) et le contenu (tel que le monde virtuel, les règles de jeu, etc.). La valeur de cette séparation est devenue évidente quand les développeurs ont commencé à concevoir d'autres jeux en changeant simplement de contenu (nouveau monde virtuel, nouvelles armes, nouveaux personnages, etc). Aujourd'hui, il existe plusieurs moteurs de jeux sur le marché, à tel point que certains développeurs en font leur principale activité [Gregory et al., 2009].

De nos jours un moteur de jeu ne se contente pas seulement du rendu et de la physique, les derniers moteurs offre un framework complet qui permet à des particuliers (pas forcement des informaticiens) de développer leur propre jeu. Les moteurs les plus perfectionnés permettent par exemple : la prise en charge des échanges réseau (pour les parties multi-joueurs), l'édition de scénario, simuler des personnages non joueurs (PNJ), ou la création (et l'édition) de monde virtuel.

A.3 Quelque exemple de moteur de jeu

Dans cette section nous allons voir différents exemples de moteur de jeu en détaillons un peu plus leur architecture et leurs fonctionnalités. Nous verrons notamment le moteur Delta 3D utilisé dans SIMFOR, l'Unreal engine, le CryEnngine et l'Unity 3D.

A.3.1 Delta 3D

Delta3D est un moteur de jeu gratuit open source basé sur une licence Lesser GNU Public License (LGPL). Delta3D est constitué d'une fine couche d'unification qui se trouve au sommet de nombreux modules open source. Il dispose d'un haut niveau, multi-plateforme (Win32 et Linux) avec une API C++ conçu pour faciliter l'apprentissage de l'outil, mais qui laisse toujours la possibilité à l'utilisateur d'accéder aux niveaux inférieurs. Les développeurs peuvent concevoir du contenu via l'éditeur, via des scripts Python grâce à l'API Delta3D ou bien directement avec des outils sous-jacents. Il est complètement modulaire et permet une approche "*best-of-breed*" (meilleur de sa catégorie) avec laquelle chaque module peut être remplacé si une meilleure alternative est disponible [Darken et al., 2005]. La figure A.1 montre l'architecture générale de Delta3D.

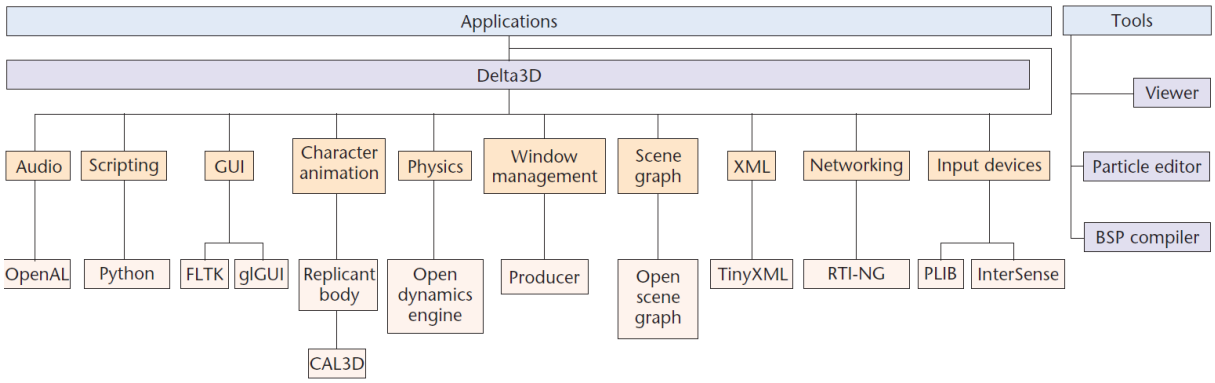


Figure A.1 – Architecture général du moteur de jeu Delta3D. [Darken et al., 2005]

Pour le développement rapide d'applications, Delta3D dispose d'outils de haut niveau comme : une visionneuse d'objet 3D, un éditeur de particules, un compilateur d'espace de partition binaire (BSP), une interface graphique de débogage, un éditeur de niveau complet, etc. Delta3D peut aussi s'occuper des échanges réseau (gestion client serveur) ainsi que la capture et l'enregistrement du gameplay à des fins de rejeu. Delta3D est multi-genre grâce à une application à base de classe qui optimise les conditions générales du jeu, par exemple pour un jeu de tir à la première personne, un jeu de stratégie en temps réel, etc.

Il y a beaucoup de travaux basés sur le moteur Delta3D [1] surtout dans le domaine de la simulation. Malheureusement la communauté du moteur est de moins en moins nombreuse et la dernière version en date de Delta3D est de mars 2012 avec des évolutions mineurs. À noter que certaines fonctions ou fonctionnalités existent mais ne sont pas implémentées (corps de la fonction vide).

1. http ://www.delta3d.org/index.php ?topic=projects

A.3.2 Unreal engine

L'Unreal Engine est un moteur de jeu développé par Epic Games et disponible gratuitement (pour des produits non commerciaux), mais sans code source, il est aussi vendu dans une version complète pour des produits commerciaux. Il est développé en C++, la première itération est parue en 1998 avec le jeu *Unreal Tournament* et est à sa quatrième itération parue à la fin de l'année 2013, la figure A.2 montre l'évolution des différentes itérations.



Figure A.2 – Évolution graphique de l'UE

L'Unreal Development Kit (UDK) est le kit de développement fourni par Epic Games, la version 3 de l'UE 3 offre un nombre important de possibilités de création, que ce soit pour le contenu comme pour le code source (UnrealScript). Ces outils inclus sont [2] :

- **UnrealEd :** l'éditeur de jeu de l'UE 3 offre un environnement d'édition entièrement intégrée. Tous les principaux outils du moteur sont accessibles par UnrealEd. Le nouveau *Unreal Content Browser* facilite la gestion des ressources comme les mailles, les matériaux, les sons et les animations ;
- **Unreal Kismet :** éditeur de scripts (en Flowgraph) ;
- **AnimSet Viewer :** cet outils permet créer et d'organiser des animations.
- **Unreal AI :** le système d'intelligence artificielle de l'UE 3 fournit des systèmes robustes de navigation de l'IA. UE 3 supporte la création automatique et l'utilisation de mailles de navigation, donnant personnages contrôlés par l'IA une conscience spatiale de leur environnement et la capacité de prendre des décisions intelligentes pour leur déplacement.

2. http ://www.unrealengine.com/features/

- **Audio :** UE permet d'immerger l'utilisateur grâce au son 3D. UE 3 supporte les dernières méthodes de compression audio, y compris Ogg Vorbis, et donne un contrôle complet sur terrain, les niveaux, en boucle, le filtrage, la modulation et la randomisation.
- **Terrain :** l'UE permet de concevoir des cartes dynamiques déformable en hauteur et en profondeur avec plusieurs couches de matériaux comme l'herbe, végétation ou la neige.
- **Réseau :** l'UE 3 fournit une architecture réseau de haut niveau flexible adapté à de nombreux genres de jeux, ainsi que des projets de simulation.
- **Scaleform :** cet outil facilite la création d'interface utilisateur tels que les menus du jeu.
- **Unreal PhAT :** est l'outil qui gère la physique dans l'UE 3, cet outil est basé sur le système PhysX de NVIDIA. PhAT permet de créer des environnements destructibles qui immerge l'utilisateur dans le monde virtuel.
- **Unreal Matinee :** cet outil permet l'édition de cinématiques dans l'UE 3 et fournit des outils solides pour créer des cinématiques superbes avec le moteur de jeu.
- **Unreal Swarm :** est un système évolutif qui permet de distribuer et optimiser la charge de travail sur les réseaux à grande vitesse les ordinateurs multi-cœurs.
- **FaceFX :** est un outil pour l'animation des visages. Cet outil permet au développeur de créer des animations faciales à partir de fichiers audio.

L'UE possède d'autres fonctionnalités qui le rendent très complet, malheureusement il faut se tourner vers la version payante pour bénéficier de l'ensemble des fonctionnalités (ainsi au code bas niveau) et il faut pas mal de temps pour maîtriser un outil aussi complexe et riche. Ceci est dû au fait que l'UE est avant tout destiné au développement de jeu vidéo. Néanmoins on peut trouver des travaux de simulation basés sur le moteur UE (en général la version 1 et 2) comme la simulation de robot dans zone de secours [Wang et al., 2003] [Nourbakhsh et al., 2005] ou une simulation de d'évacuation [Mól et al., 2008].

A.3.3 CryEngine

Le CryEngine est un moteur de jeu développé par le studio Crytek. À l'origine, développé comme une démo technologique (X-Isle Dinosaurs Island) pour nVidia, ensuite transposé pour le développement du Far Cry. Il a connu de nombreuse itérations mais durant l'année 2013 Crytek décide de ne plus donner de numérotation pour son moteur de jeu. Il est disponible en version payante et gratuite sous forme de licence shareware (avec des fonctionnalités limitées) et propose même une licence pour les jeux sérieux. Le CryEngine est développé en C++ et est basé sur la Sandbox qui permet de créer : des environnements (terrains, montagnes, océans, forêts, etc.), des scénarios, sons, des

PNGs, etc. La Sandbox est au cœur de la création de contenu, la figure A.3 montre l'interface de la Sandbox.



Figure A.3 – La Sandbox du CryEngine.

La Sandbox du Cryengine fournie la plupart des fonctionnalités de son homologue UE, avec un éditeur d'environnement, gestion du réseau, l'IA, etc [3]. Comme l'UE, il a les même avantages comme les différentes fonctionnalités (qui sont en pointe aux niveaux technologique, tessellation, gestion dynamique de la lumière, réflexion temps réel, etc.) et les même inconvenants (outils complexes et difficile à maîtriser) d'autant plus que la version gratuite est bridée et ne permet pas une modification du code. Il existe plusieurs travaux dans la littérature qui utilisent ce moteur de jeu comme [Martínez-Durá et al., 2011] pour un jeu sérieux pour la sécurité et la formation médical ou bien dans la domaine de la simulation comme la simulation de vie virtuelle [Stone, 2009] ou bien la simulation de conception de parc éolien [MANYOKY et al., 2012].

A.3.4 Unity 3D

Unity 3D est moteur de jeu multi-plateforme développé par Unity. La dernier version (4.x) est compatible avec Windows, Mac OS, Linux, Android, iOS, ainsi que la plupart des consoles de jeux actuellement sur le marché. Le cœur d'Unity 3D est développé en C++ avec une IDE (environnement de développement intégré) en C#. Il propose une version gratuite et une autre professionnelle payante (1500 $ ou 75 $ par mois) [4].

Comme l'UE et le CryEngine, Unity 3D fournit différents outils pour la conception de jeu comme un éditeur de contenus (environnement modèle 3D, animation), gestion du réseau, script, etc [5]. Mais le principal avantage d'Unity 3D est son aspect multi-plateforme que ça soit PC, console, ou périphérique mobile (smartphone et tablette), la figure A.4 illustre l'éditeur Unity 3D. Unity 3D est moteur encore jeune et ne bénéficie pas des dernières techniques de rendu (3D, lumière, etc.) mais est assez simple à

3. http ://mycryengine.com/
4. http ://unity3d.com/
5. http ://unity3d.com/unity

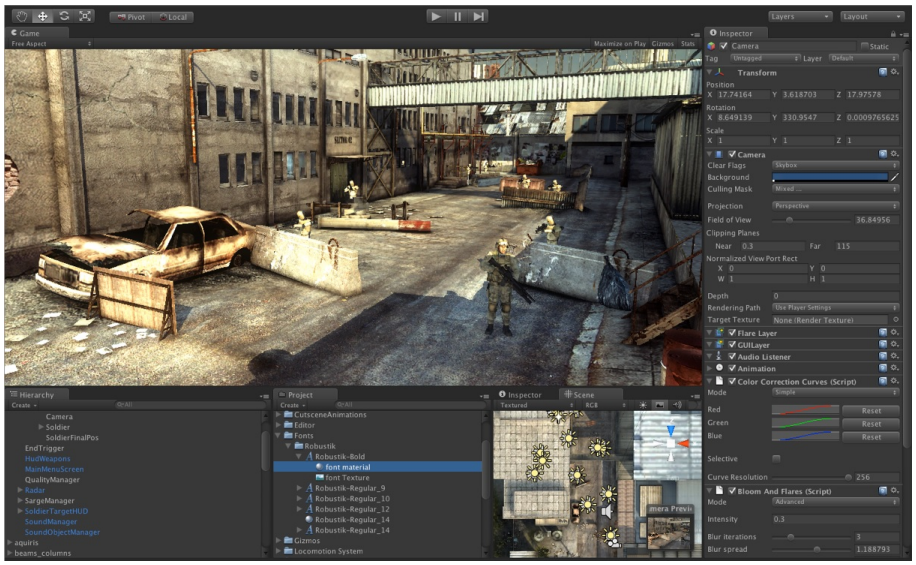


Figure A.4 – L'éditeur Unity 3D.

utiliser et s'adresse avant tout aux développeurs indépendants, néanmoins on retrouve quelques travaux dans la littérature basés sur Unity 3D comme [Cherry, 2013] qui utilise le moteur pour la simulation de foule.

A.4 Conclusion

Dans cette annexe nous avons vu que les moteurs de jeu ne sont pas simplement des outils de rendu 3D/2D mais une base solide pour le développement de jeux vidéo, de jeux sérieux ou autres simulations. Nous avons vu différents moteurs de jeux avec des approches différentes (gratuits, payants, open source) qui répondent à des besoins particuliers. Pour bien choisir un moteur de jeu, il faut tout d'abord bien définir ses besoins afin de choisir celui qui y répond le mieux.

ANNEXE B

Les interfaces de SIMFOR

Sommaire

B.1 Introduction 163
B.2 Les actions dans SIMFOR 163
B.2.1 L'interface téléphone 164
B.2.2 L'interface fax 165
B.2.3 L'interface radio 166
B.2.4 L'interface carte 167
B.2.5 L'interface main courante 168
B.2.6 L'interface de déplacement 169
B.3 Autres interfaces 170
B.4 Conclusion 171

B.1 Introduction

Dans cette annexe nous verrons les différentes interfaces graphiques de SIMFOR qui permettent aux apprenants et au moniteur de réaliser des actions, de récupérer des informations, ou encore de lancer des scripts.

B.2 Les actions dans SIMFOR

Dans cette section nous verrons toutes les interfaces liées aux actions dans SIMFOR. Nous détaillerons l'interface téléphone, fax, radio, carte, main courante, et le déplacement (à pied et en voiture).

B.2.1 L'interface téléphone

Pour appeler un acteur dans SIMFOR, l'apprenant clique sur le bouton téléphone de l'interface principale (figure V.11). Après cela l'interface téléphone apparait en avant plan (figure B.1). L'interface téléphone contient un champ pour noter le numéro de téléphone de l'acteur à appeler, un bouton appeler, un bouton décrocher (qui est activé lorsque l'apprenant reçoit un appel), un bouton raccrocher pour terminer la communication et enfin un bouton annuaire.

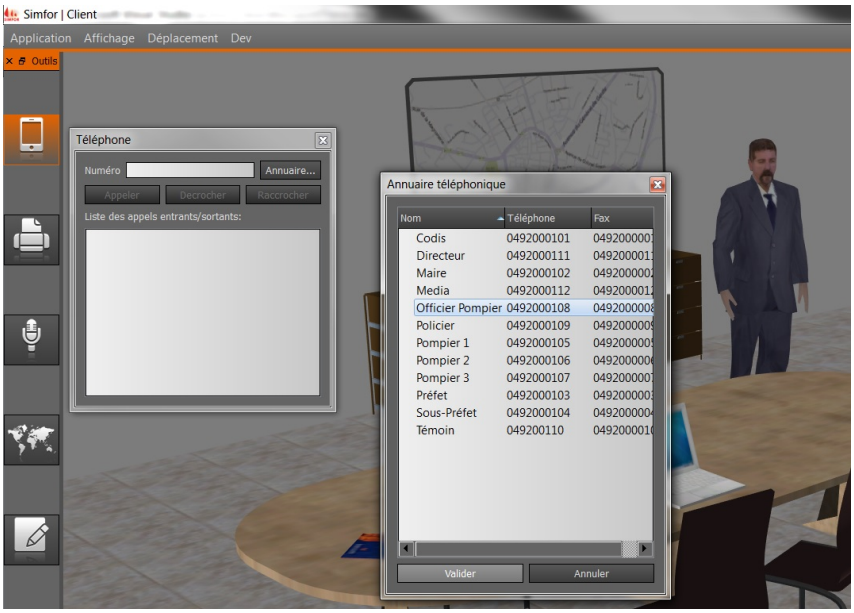


Figure B.1 – Interface de l'action téléphone dans SIMFOR

Si l'apprenant ne connait pas le numéro de l'acteur cible, il peut appuyer sur le bouton annuaire, une autre fenêtre s'ouvre alors avec la liste de l'ensemble des acteurs ainsi que leurs numéros (téléphone et fax). L'apprenant n'a plus qu'à sélectionner l'acteur cible et appuyer sur le bouton valider. Si l'acteur cible est joueur humain (un autre apprenant), la communication se fait par voix oral via VOIP. Si l'acteur cible est PNJ controlé par un AgJ (Game Agent), la communication se fait sous forme de dialogue textuel avec des échanges de phrases à choix multiple comme illustré dans la figure B.2.

Le premier champ de la fenêtre de dialogue présente la phrase de l'acteur cible (incarné par un GA). Le deuxième champ présente les choix des phrases que l'apprenant a à sa disposition avec la possibilité de changer entre ces différents choix avec les boutons suivant et précédent. Une fois que l'apprenant a fait son choix, il doit appuyer sur le bouton répondre pour valider son choix. Le dernier champ de cette fenêtre retrace l'historique de la conversation.

L'interface du téléphone permet aussi d'afficher l'historique des communications passées avec l'heure de l'appel, le type de l'appel (entrant ou sortant) ainsi que son statut

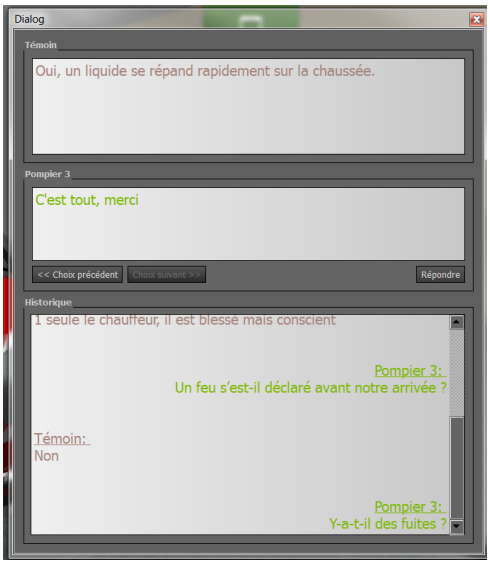


Figure B.2 – Communication avec un GA.

(réussi ou échoué), ceci est illustré dans la figure B.3.

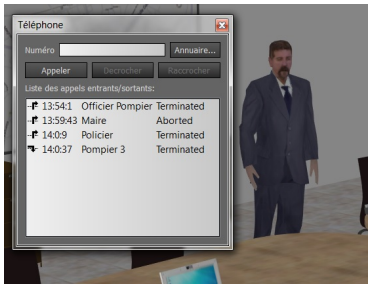


Figure B.3 – Historique des appels dans l'interface téléphone.

B.2.2 L'interface fax

Pour envoyer un fax, l'apprenant doit appuyer sur le bouton fax (figure V.11). Après cela l'interface fax apparait en avant plan (figure B.4). Le champ en haut de l'interface fax représente la boite de réception de l'apprenant, et le champ en bas de l'interface fax représente la boute envoi. Pour consulté un fax (émis ou reçu) l'apprenant fait un double clique sur le fax à consulté.

Pour envoyer un fax, l'apprenant a deux possibilité, soit en retransmettant un fax reçu ou déjà envoyer (en appuyant sur le bouton retransmettre), soit éditer un nouveau (bouton nouveau). En appuyant sur le bouton *nouveau*, une nouvelle fenêtre s'affiche qui propose de parcourir les fiches de fax et de saisir un ou plusieurs destinataires (voir figure B.5). En appuyant sur le bouton *parcourir* et dépendant du scénario jouer, une boite de dialogue s'affiche en proposant à l'apprenant un ensemble de fiches (pré-formatées au format officiel de la gestion des risques). Une fois la fiche sélectionnée,

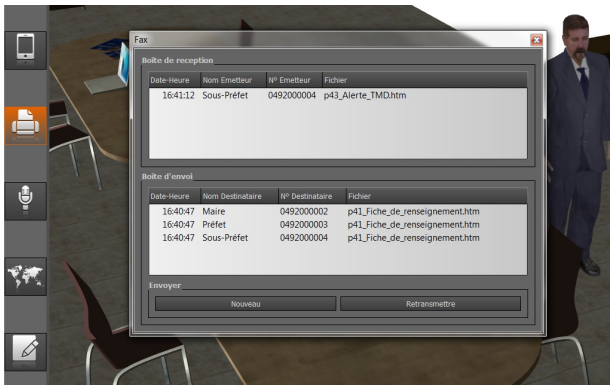


Figure B.4 – Interface de l'action téléphone dans SIMFOR.

l'apprenant édite les différents champs du formulaire avec les informations en sa possession et appuie sur le bouton *envoyer* pour envoyer le fax. Ces étapes sont illustrées dans la figure B.5.

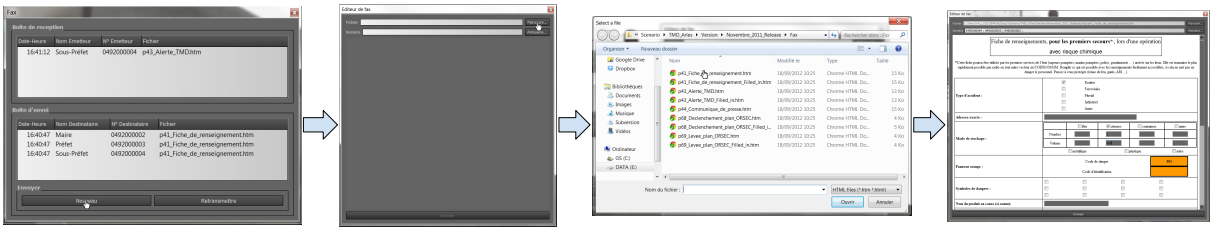


Figure B.5 – Les étapes pour envoyer un nouveau fax dans SIMFOR.

B.2.3 L'interface radio

Pour communiquer avec un acteur via radio dans SIMFOR, l'apprenant clique sur le bouton radio de l'interface principale (figure V.11). Après cela l'interface radio apparait en avant plan (figure B.6). L'interface radio contient un bouton pour régler le canal de communication et un bouton pour parler. Pour discuter avec les acteurs qui sont dans le même canal, l'apprenant doit maintenir le bouton parler et transmettre son message (par VOIP).

Quand l'apprenant utilise la radio, tous les acteurs qui possèdent le moyen radio et qui sont sur le même canal vont entendre le message de l'apprenant et peuvent éventuellement participer à la l'échange (communication de groupe). Quand l'apprenant incarne un rôle qui ne possède pas la radio, le bouton radio est désactiver et l'apprenant n'entend pas les messages qui y sont diffusés.

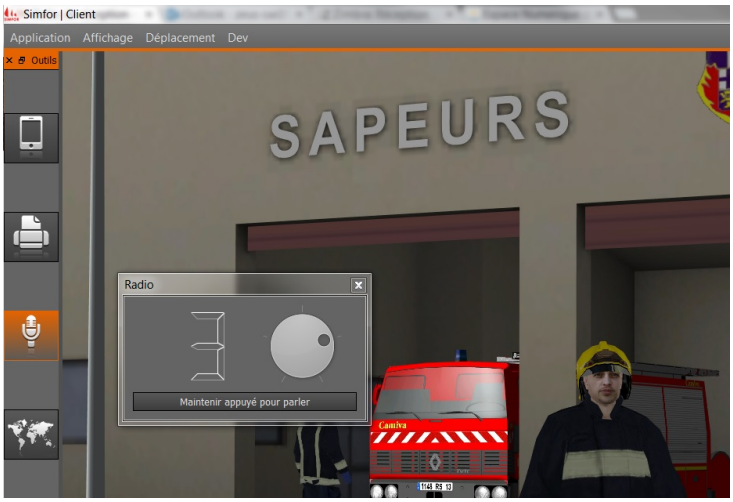


Figure B.6 – L'interface radio dans SIMFOR.

B.2.4 L'interface carte

Durant la partie, l'apprenant peut se situer sur la carte et peut annoter et dessiner différentes formes pour signaler des zones sur la carte comme le périmètre de sécurité, la zone d'explosimétrie, etc. Ceci est possible via l'interface carte (figure B.7). Pour accéder à l'interface carte l'apprenant doit appuyer sur le bouton carte de l'interface principale de SIMFOR (figure V.11).



Figure B.7 – L'interface carte dans SIMFOR.

L'interface carte offre de nombreux outils d'édition. L'apprenant a la possibilité d'annoter des points de différentes couleurs sur la carte (voir figure B.8a) ainsi que la possibilité de tracer des zones de différentes formes et de couleurs (voir figure B.8a). De plus, grâce l'interface carte, l'apprenant peut se situer sur la carte et voir la position des autres acteurs ainsi que les adresses clés du scénario (figure B.8b).

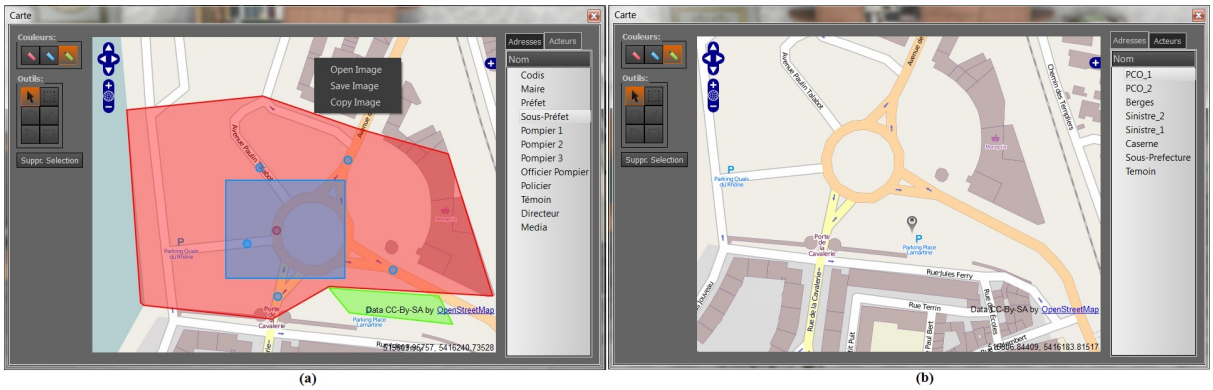


Figure B.8 – Édition de la carte dans SIMFOR.

Une fois la carte éditée, l'interface carte donne la possibilité d'enregistrer la carte. En cliquant sur le bouton droit de la souris sur la carte, un menu apparait en proposant d'enregistrer l'image, de copier l'image ou d'ouvrir l'image (l'image issue de la carte éditée). Ceci est illustré dans la figure B.8a. Dans le menu des onglets sur la droite de l'interface carte (figure B.8b), nous avons deux onglets, un pour les acteurs, et un autre pour les adresses. En cliquant sur une adresse, la carte se focalise sur la position de l'adresse avec un petit marqueur comme illustré dans la figure B.8b.

B.2.5 L'interface main courante

La main courante est un outil qui permet aux intervenants de la gestion de crise de retranscrire l'évolution du sinistre en temps réel. En condition réel, l'outil main courante est un portail web sur lequel les intervenants de la gestion de crise postent des messages. Ainsi, ces messages seront consultés par les autres intervenants qui ont accès à ce portail. Dans SIMFOR, pour ajouter un message au portail, l'apprenant doit appuyer sur le bouton main courante (voir figure V.11). Une fois que c'est fait, l'interface main courante apparait en premier plan (figure B.9).

L'interface main courante contient un champ pour écrire le message à transmettre, un bouton envoyer pour valider l'envoi du message, et enfin un champ qui affiche tous les messages envoyés depuis le début de l'exercice. Ce dernier champ (affichage des messages) est composé de trois colonnes, la première colonne représente l'heure à laquelle le message a été envoyé, la deuxième colonne représente l'acteur à l'origine du message, et enfin la dernière colonne représente le contenu du message.

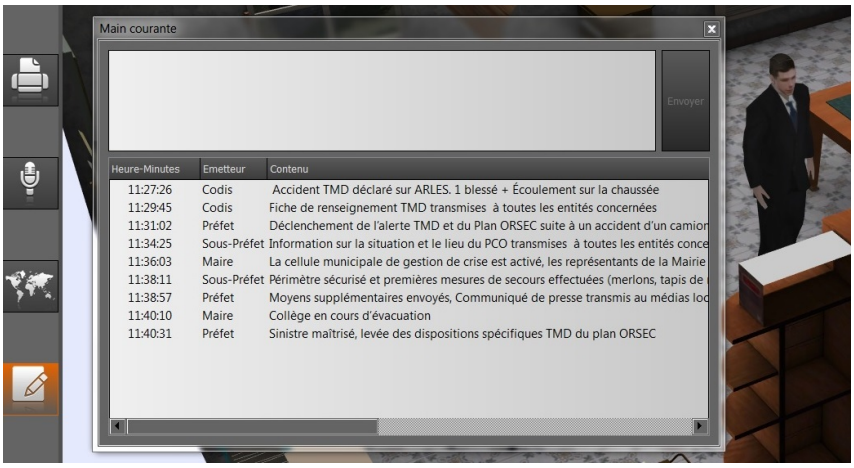


Figure B.9 – Interface de l'outil la main courante dans SIMFOR.

B.2.6 L'interface de déplacement

Pour se déplacer dans l'environnement 3D, l'apprenant peut soit cliquer avec le bouton gauche de la souris sur un point de l'environnement, soit utiliser le déplacement automatique en cliquant sur le bouton déplacement piéton (voir figure V.11). Lorsque l'apprenant appuie sur le déplacement automatique, une fenêtre s'affiche au premier plan qui propose à l'apprenant une liste d'adresse (adresse clé du scénario), ainsi l'apprenant n'a plus qu'a sélectionné une adresse et son avatar s'y rendra automatiquement. La fenêtre des adresses est illustrée dans la figure B.10.

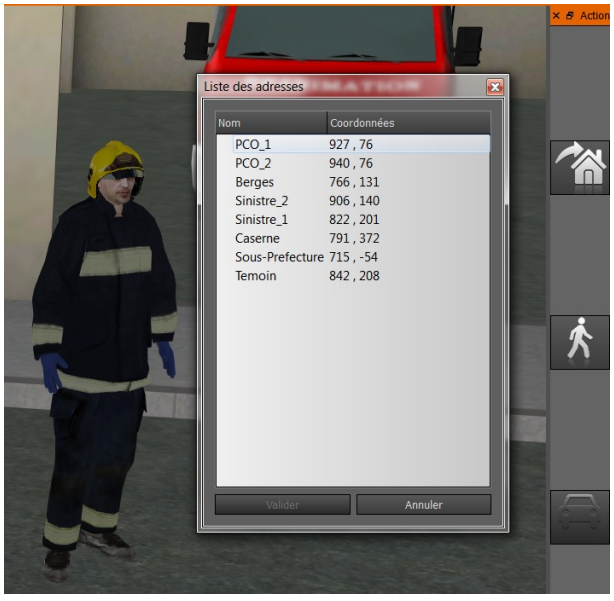


Figure B.10 – Fenêtre adresse pour le déplacement automatique dans SIMFOR.

Pour le déplacement en véhicule, l'apprenant doit être à proximité d'un véhicule et appuyer sur le bouton déplacement en véhicule (voir figure V.11). Si le rôle de l'apprenant lui permet d'utiliser ce véhicule, alors la fenêtre adresse (comme dans la figure

B.10) apparait et l'apprenant n'a plus qu'à choisir sa destination. Si l'apprenant n'est pas à proximité d'un véhicule ou que l'apprenant n'a pas le droit d'utiliser ce véhicule, alors le bouton déplacement en véhicule est grisé (ne pas être actionné) comme dans la figure B.10. Le déplacement en véhicule est automatique et est illustré dans la figure B.11.



Figure B.11 – Déplacement en véhicule dans SIMFOR.

B.3 Autres interfaces

Dans certains scénarios SIMFOR et dépendant du rôle incarné, les apprenants sont amenés à réaliser des actions prédéfinies qui ne sont pas intégrées dans l'interface principale de SIMFOR. Ces actions peuvent être réalisées via l'interface outil action (voir figure V.11). Cette interface permet d'exécuter différents types d'actions selon le scénario joué et/ou le rôle incarné, par exemple le détachement de sirène d'alerte à la population (voir figure B.12) pour le maire, ou bien la mise en place des merlons de fortune pour le pompier. Cette interface propose une liste d'actions que l'apprenant peut réaliser ainsi qu'une petite description de cette dernière.

Durant l'exercice, l'apprenant peut avoir des indications sur les actions qu'il fait, par exemple si l'apprenant réalise une action téléphone et qu'il n'a pas appelé la bonne personne, le menu aide va apparaître en premier plan et lui indiquer qu'il n'a pas appelé la bonne personne comme illustré dans la figure B.13. L'apprenant peut aussi avoir des indications sur les missions réalisées avec des résultats d'évaluations. Ces indications peuvent être désactivées pour ne pas biaiser le résultat de l'apprentissage. A tout moment l'apprenant peut consulter l'interface aide en appuyant sur le bouton aide (voir figure V.11).

Dans l'environnement virtuel, l'apprenant a la possibilité de consulter les métadonnées de chaque objet 3D grâce au SIG (Système d'Informations Géographique), et cela en appuyant sur la touche *i* du clavier et en cliquant sur l'objet 3D. En cliquant sur l'objet une fenêtre d'informations apparait en premier plan qui décrit l'objet incriminé

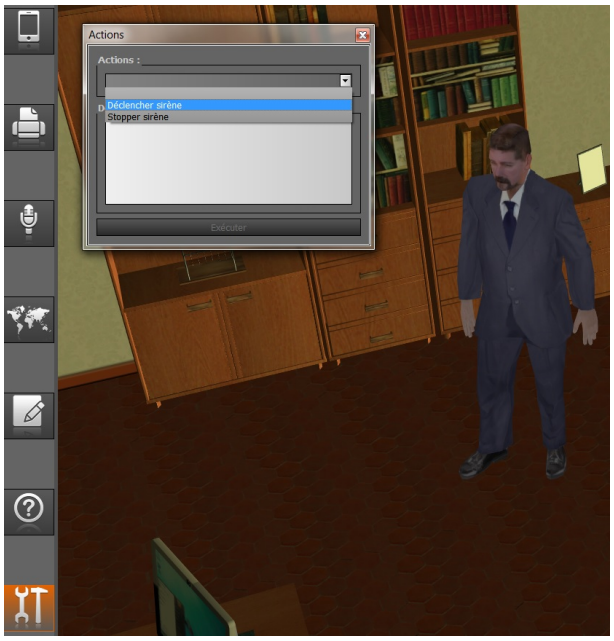


Figure B.12 – Interface de l'outil action dans SIMFOR.

comme illustré dans la figure B.14.

Certains rôles comme les pompiers peuvent donner accès à des actions dédiées à la gestion des risques comme par exemple la mesure d'explosimétrie ou l'établissement du périmètre de sécurité comme illustré dans la figure B.15.

Dans le menu action du serveur SIMFOR, le moniteur (ou le maître de jeu) peut déclencher différents actions (dépendant du scénario joué), par exemple déclencher ou arrêter l'inflammation de l'hydrocarbure, lancer la partie, etc. Le menu action du serveur SIMFOR est illustré dans la figure B.16

B.4 Conclusion

Dans cette annexe nous avons présenté les différentes interfaces de SIMFOR et les différentes possibilités offertes par l'environnement de formation. Grâce au jeu sérieux SIMFOR, les intervenants de la gestion de crise ont à leur disposition un outil complet et facile à prendre en main pour se familiariser avec la gestion de crise. SIMFOR offre la possibilité de jouer différents types de scénario grâce à sa modularité et ces nombreux éditeurs (éditeur de scénario, de PNJs, de dialogues, etc.).

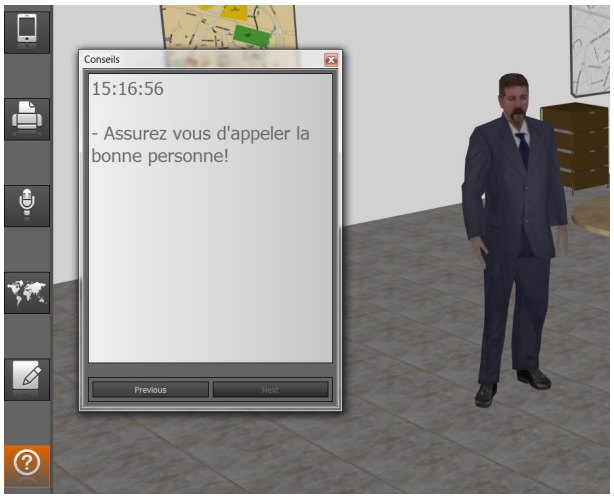


Figure B.13 – Interface d'aide dans SIMFOR.



(a) (b) (c)

Figure B.14 – Interface information méta-données dans SIMFOR.
(a) : Méta-données sur un arbre de l'environnement virtuel.
(b) : Méta-données sur un bâtiment de l'environnement virtuel.
(c) : Méta-données sur une bouche d'égout de l'environnement virtuel.



Figure B.15 – Mesure du périmètre de sécurité dans SIMFOR.

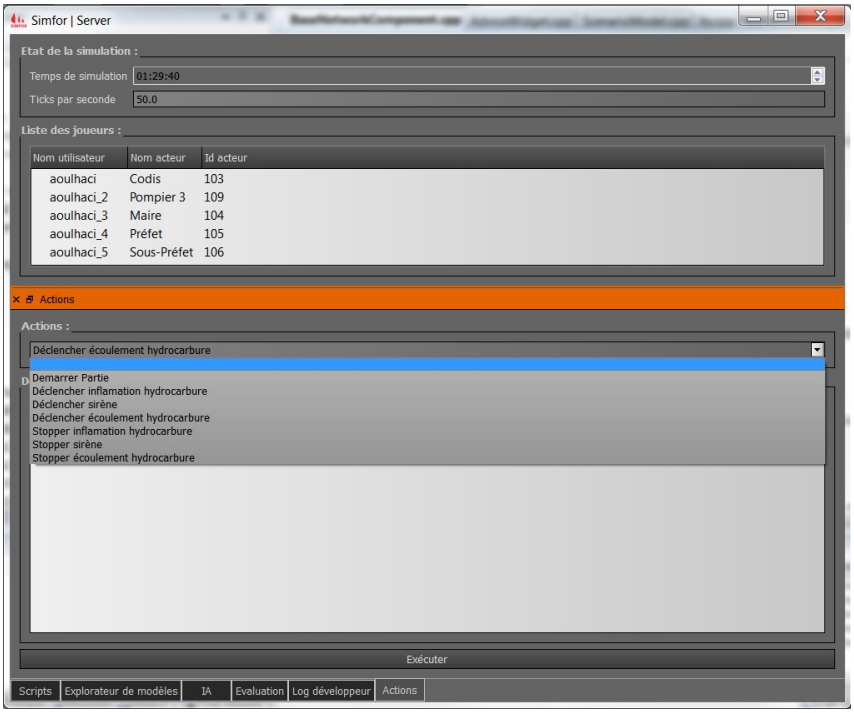


Figure B.16 – Menu action du serveur SIMFOR.

Liste des publications

A Multi-Agent Architecture for collaborative Serious Game applied to Crisis Management training : improving adaptability of Non Player Characters, A. Oulhaci, E. Tranvouez, S. Fournier, B. Espinasse, EAI endorsed transaction in Serious Game journal, (à paraître).

A Multi-Agent Architecture for collaborative Serious Game applied to Crisis Management training : improving adaptability of Non Played Characters, A. Oulhaci, E. Tranvouez, S. Fournier, B. Espinasse, the 7th European Conference on Games Based Learning, 2013, Porto.

Intelligent Tutoring Systems and Serious Game for Crisis Management : a Multi-Agents Integration Architecture, A. Oulhaci, E. Tranvouez, S. Fournier, B. Espinasse, 22nd WETICE IEEE conference - 3rd Track on Collaborative Technology for Coordinating Crisis Management, 2013, Hammamet.

A Multi-Agent System for Learner Assessment in Serious Games : application to learning processes in Crisis Management, A. Oulhaci, E. Tranvouez, S. Fournier, B. Espinasse, Seventh IEEE International Conference on RESEARCH CHALLENGES in INFORMATION SCIENCE, 2013, Paris.

Evaluation Multicritères et Distribuée pour l'Apprentissage Collectif de Procédures dans un Jeux Sérieux pour la Gestion de Crise, A. Oulhaci, E. Tranvouez, S. Fournier, B. Espinasse, Journée EIAH & IA 2013, Conférence EIAH 2013, Université Paul Sabatier, Toulouse, IRIT, 2013.

Un Système tutoriel intelligent pour SIMFOR : Un jeu sérieux pour la gestion des risques, A. Oulhaci, E. Tranvouez, S. Fournier, B. Espinasse, 1 er journée des systèmes collaboratifs, Sysco 2012, Sousse.

Simulation de déplacement de piétons : Vers un modèle de perception et de prédiction d'action chez autrui, L. Bourgois, A. Oulhaci, J.-M. Auberlet, Journées Francophone des Systèmes Multi Agents, 2010.

Références bibliographiques

[Alvarez, 2007] Alvarez, J. (2007). *Du jeu vidéo au serious game : approches culturelle, pragmatique et formelle*. Thèse de doctorat, Université de Toulouse.

[Amokrane, 2010] Amokrane, K. (2010). *Suivi de l'apprenant en environnement virtuel pour la formation á la prévention des risques sur des sites SEVESO*.

[Anderson, 1988] Anderson, J. (1988). The expert module. *Foundations of intelligent tutoring systems*, pages 21–53.

[Astin et al., 2012] Astin, A. et al. (2012). *Assessment for excellence : The philosophy and practice of assessment and evaluation in higher education*. Rowman & Littlefield Publishers.

[Bakkes et al., 2009] Bakkes, S., Spronck, P., et van den Herik, J. (2009). Rapid and reliable adaptation of video game ai. *Computational Intelligence and AI in Games, IEEE Transactions on*, 1(2) :93–104.

[Baudouin et al., 2007] Baudouin, C., Beney, M., Chevaillier, P., et LE, A. (2007). Recueil de traces pour le suivi de l'activité d'apprenants en travaux pratiques dans un environnement de réalité virtuelle. *Revue Sticef. org*, 16 :33.

[Baumard et Vidal, 2009] Baumard, P. et Vidal, R. (2009). Fiabiliser la gestion des feux de tres grande ampleur enhancing reliability in large scale willand fire response organization.

[Bénaben et al., 2008] Bénaben, F., Hanachi, C., Lauras, M., Couget, P., et Chapurlat, V. (2008). A metamodel and its ontology to guide crisis characterization and its collaborative management. In *Proceedings of the 5th International Conference on Information Systems for Crisis Response and Management (ISCRAM), Washington, DC, USA, May*, pages 4–7.

[Binsubaih et al., 2006] Binsubaih, A., Maddock, S., et Romano, D. (2006). A serious game for traffic accident investigators. *Interactive Technology and Smart Education*, 3(4) :329–346.

[Brans et Mareschal, 2005] Brans, J.-P. et Mareschal, B. (2005). Promethee methods. In *Multiple criteria decision analysis : state of the art surveys*, pages 163–186. Springer.

[Bruillard, 1997] Bruillard, E. (1997). *Les machines à enseigner*. Hermès.

[Buche, 2005] Buche, C. (2005). *Un système tutoriel intelligent et adaptatif pour l'apprentissage de compétences en environnement virtuel de formation*.

[Buche et al., 2010] Buche, C., Bossard, C., Querrec, R., et Chevaillier, P. (2010). Pegase : A generic and adaptable intelligent system for virtual reality learning environments. *International Journal of Virtual Reality*, 9(2) :73.

[Burns et Capps, 1989] Burns, H. et Capps, C. (1989). Foundations of intelligent tutoring systems : An introduction. *Foundations of intelligent tutoring systems*, pages 1–18.

[Burton et Brown, 1979] Burton, R. R. et Brown, J. S. (1979). An investigation of computer coaching for informal learning activities. *International Journal of Man-Machine Studies*, 11(1) :5–24.

[Chadli et al., 2010] Chadli, A., Tranvouez, E., et Bendella, F. (2010). Cooperative skills evaluation and participative multiagent simulation for rodent control training. In *Machine and Web Intelligence (ICMWI), 2010 International Conference on*, pages 41–47. IEEE.

[Cherry, 2013] Cherry, D. (2013). Rvo collision avoidance in unity 3d. *Distributed Research Experiences for Undergraduates in Computer Science and Engineering, DREU2013*.

[Darken et al., 2005] Darken, R., McDowell, P., et Johnson, E. (2005). Projects in vr : the delta3d open source game engine. *Computer Graphics and Applications, IEEE*, 25(3) :10–12.

[DDSC, 2006] DDSC (2006). *Direction de la défense et de la sécurité civiles : Guide ORSEC départemental méthodes générales*. www.interieur.gouv.fr.

[de Freitas et Jarvis, 2006] de Freitas, S. et Jarvis, S. (2006). A framework for developing serious games to meet learner needs. In *The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC)*, volume 2006. NTSA.

[de los Angeles Constantino-Gonzalez et al., 2003] de los Angeles Constantino-Gonzalez, M., Suthers, D. D., et de los Santos, J. G. E. (2003). Coaching web-based collaborative learning based on problem solution differences and participation. *International Journal of Artificial Intelligence in Education*, 13(2) :263–299.

[Dintilhac et Ignace, 2005] Dintilhac, J. et Ignace, R. (2005). Evaluation de la technologie en collége, académie de montpellier. Technical report, ACADEMIE DE MONTPELLIER.

[Edward et al., 2008] Edward, L., Lourdeaux, D., Barthès, J.-P. A., Lenne, D., et Burkhardt, J.-M. (2008). Modelling autonomous virtual agent behaviours in a virtual environment for risk. *IJVR*, 7(3) :13–22.

[El-Kechaï, 2007] El-Kechaï, N. (2007). *Suivi et Assistance des apprenants dans les environnements virtuels de formation*. Thèse de doctorat.

[El-Kechaï et Després, 2006] El-Kechaï, N. et Després, C. (2006). A plan recognition process, based on a task model, for detecting learner's erroneous actions. In *Intelligent tutoring systems*, pages 329–338. Springer.

[Fipa, 2001] Fipa (2001). Agent software integrationspecication. *Foundation for Intelligent Physical Agents*.

[Garris et al., 2002] Garris, R., Ahlers, R., et Driskell, J. (2002). Games, motivation, and learning : A research and practice model. *Simulation & gaming*, 33(4) :441–467.

[Giordani et al., 2005] Giordani, A., Gerosa, L., Soller, A., et Stevens, R. (2005). Extending an online individual scientific problemsolving environment to support and mediate collaborative learning. In *12th International Conference on Artificial Intelligence in Education*.

[Gregory et al., 2009] Gregory, J., Lander, J., et Whiting, M. (2009). *Game engine architecture*. AK Peters.

[Halff, 1988] Halff, H. M. (1988). Curriculum and instruction in automated tutors. *Foundations of intelligent tutoring systems*, pages 79–108.

[Helbing et al., 2002] Helbing, D., Farkas, I. J., Molnar, P., et Vicsek, T. (2002). Simulation of pedestrian crowds in normal and evacuation situations. *Pedestrian and evacuation dynamics*, 21 :21–58.

[Hocine et al., 2011a] Hocine, N., Gouaïch, A., Di Loreto, I., Abrouk, L., et al. (2011a). Etat de l'art des techniques d'adaptation dans les jeux ludiques et sérieux. *numéro spécial RIA jeux sérieux*, 1.

[Hocine et al., 2011b] Hocine, N., Gouaich, A., et al. (2011b). A survey of agent programming and adaptive serious games.

[Hollnagel, 1993] Hollnagel, E. (1993). The phenotype of erroneous actions. *International Journal of Man-Machine Studies*, 39(1) :1–32.

[Hübner et al., 2002] Hübner, J. F., Sichman, J. S., et Boissier, O. (2002). Moise+ : towards a structural, functional, and deontic model for mas organization. In *Procee-*

dings of the first international joint conference on Autonomous agents and multiagent systems : part 1, pages 501–502. ACM.

[Jermann et al., 2004] Jermann, P., Soller, A., et Lesgold, A. (2004). Computer software support for cscl. In *What we know about CSCL*, pages 141–166. Springer.

[Johnson et Johnson, 1987] Johnson, D. W. et Johnson, R. T. (1987). *Learning together and alone : Cooperative, competitive, and individualistic learning* . Prentice-Hall, Inc.

[Johnson et al., 1998] Johnson, W., Rickel, J., Stiles, R., et Munro, A. (1998). Integrating pedagogical agents into virtual environments. *Presence*, 7(6) :523–546.

[Johnson et Wu, 2008] Johnson, W. et Wu, S. (2008). Assessing aptitude for learning with a serious game for foreign language and culture. In *Intelligent tutoring systems*, pages 520–529. Springer.

[Johnson et al., 2004] Johnson, W. L., Beal, C., Fowles-Winkler, A., Lauper, U., Marsella, S., Narayanan, S., Papachristou, D., et Vilhjálmsson, H. (2004). Tactical language training system : An interim report. In *Intelligent Tutoring Systems*, pages 336–345. Springer.

[Kay et al., 2006] Kay, J., Maisonneuve, N., Yacef, K., et Reimann, P. (2006). The big five and visualisations of team work activity. In *Intelligent tutoring systems*, pages 197–206. Springer.

[Kitano et al., 1999] Kitano, H., Tadokoro, S., Noda, I., Matsubara, H., Takahashi, T., Shinjou, A., et Shimada, S. (1999). Robocup rescue : Search and rescue in large-scale disasters as a domain for autonomous agents research. In *Systems, Man, and Cybernetics, 1999. IEEE SMC'99 Conference Proceedings. 1999 IEEE International Conference on*, volume 6, pages 739–743. IEEE.

[Komatitsch et al., 2003] Komatitsch, D., Tsuboi, S., Ji, C., et Tromp, J. (2003). A 14.6 billion degrees of freedom, 5 teraflops, 2.5 terabyte earthquake simulation on the earth simulator. In *Proceedings of the 2003 ACM/IEEE conference on Supercomputing*, page 4. ACM.

[Kubera et al., 2008] Kubera, Y., Mathieu, P., Picault, S., et al. (2008). Interaction-oriented agent simulations : From theory to implementation. In *ECAI*, pages 383–387.

[Lewis et Jacobson, 2002] Lewis, M. et Jacobson, J. (2002). Game engines. *Communications of the ACM*, 45(1) :27.

[Lourdeaux, 2001] Lourdeaux, D. (2001). *Réalité virtuelle et formation : conception d'environnements virtuels pédagogiques*. Thèse de doctorat, Paris.

[Ma et al., 2011] Ma, M., Oikonomou, A., et Jain, L. (2011). *Serious games and edutainment applications*. Springer.

[MANYOKY et al., 2012] MANYOKY, M., HAYEK, U. W., KLEIN, T. M., PIEREN, R., HEUTSCHI, K., et GRÊT-REGAMEY, A. (2012). Concept for collaborative design of wind farms facilitated by an interactive gis-based visual-acoustic 3d simulation. *Proceedings of Digital Landscape Architecture*.

[Martínez-Durá et al., 2011] Martínez-Durá, R. J., Arevalillo-Herráez, M., García-Fernández, I., Gamón-Giménez, M. A., et Rodríguez-Cerro, A. (2011). Serious games for health and safety training. In *Serious Games and Edutainment Applications*, pages 107–124. Springer.

[Mathieu et al., 2011] Mathieu, P., Panzoli, D., Picault, S., et La pertinence des systèmes multi agents, S. (2011). Serious games et sma-application à un supermarché virtuel. In *JFSMA*, pages 181–190.

[Michael et Chen, 2005] Michael, D. R. et Chen, S. L. (2005). *Serious Games : Games That Educate, Train, and Inform*. Muska & Lipman/Premier-Trade.

[Miller, 1978] Miller, J. G. (1978). Living systems.

[Miller, 1988] Miller, J. R. (1988). The role of human-computer interaction in intelligent tutoring systems. *Foundations of intelligent tutoring systems*, pages 143–189.

[Mól et al., 2008] Mól, A. C. A., Jorge, C. A. F., et Couto, P. M. (2008). Using a game engine for vr simulations in evacuation planning. *Computer Graphics and Applications, IEEE*, 28(3) :6–12.

[Murray, 1999] Murray, T. (1999). Authoring intelligent tutoring systems : An analysis of the state of the art. *International Journal of Artificial Intelligence in Education (IJAIED)*, 10 :98–129.

[Nieborg, 2004] Nieborg, D. (2004). America's army : More than a game. In *Proceedings of 35th Annual Conference of the International Simulation And Gaming Association (ISAGA) and Conjoint Conference of SAGSAGA*.

[Nourbakhsh et al., 2005] Nourbakhsh, I. R., Sycara, K., Koes, M., Yong, M., Lewis, M., et Burion, S. (2005). Human-robot teaming for search and rescue. *Pervasive Computing, IEEE*, 4(1) :72–79.

[Noy et al., 2003] Noy, N. F., Crubézy, M., Fergerson, R. W., Knublauch, H., Tu, S. W., Vendetti, J., Musen, M. A., et al. (2003). Protege-2000 : an open-source ontology-development and knowledge-acquisition environment. In *AMIA Annu Symp Proc*, volume 953, page 953.

[Ohgai et al., 2007] Ohgai, A., Gohnai, Y., et Watanabe, K. (2007). Cellular automata modeling of fire spread in built-up areas—a tool to aid community-based planning for disaster mitigation. *Computers, environment and urban systems*, 31(4) :441–460.

[Paternò et al., 1997] Paternò, F., Mancini, C., et Meniconi, S. (1997). Concurtasktrees : A diagrammatic notation for specifying task models. In *Proceedings of the IFIP*

TC13 Interantional Conference on Human-Computer Interaction, volume 96, pages 362–369.

[Provitolo et al., 2009] Provitolo, D., Müller, J., et Dubos-Paillard, E. (2009). Validation of an ontology of risk and disaster through a case study of the 1923 great kanto earthquake. In *3rd International Conference on Complex Systems and Applications (ICCSA 2009a)*, page 11.

[Querrec, 2003] Querrec, R. (2003). *Les systèmes multi-agents pour les environnements virtuels de formation : application à la sécurité civile*. Thèse de doctorat, Brest.

[Rao et al., 1995] Rao, A., Georgeff, M., et al. (1995). Bdi agents, from theory to practice. In *Proceedings of the first international conference on multi-agent systems (ICMAS-95)*, pages 312–319. San Francisco.

[Rasmussen, 1983] Rasmussen, J. (1983). Skills, rules, and knowledge ; signals, signs, and symbols, and other distinctions in human performance models. *Systems, Man and Cybernetics, IEEE Transactions on*, (3) :257–266.

[Raybourn, 2005] Raybourn, E. M. (2005). Adaptive thinking and leadership training for cultural awareness and communication competence. *Interactive Technology and Smart Education*, 2(2) :131–134.

[Raybourn, 2007] Raybourn, E. M. (2007). Applying simulation experience design methods to creating serious game-based adaptive training systems. *Interacting with Computers*, 19(2) :206–214.

[Schmidt et Bjork, 1992] Schmidt, R. et Bjork, R. (1992). New conceptualizations of practice : Common principles in three paradigms suggest new concepts for training. *Psychological science*, 3(4) :207–217.

[Self, 1988] Self, J. (1988). Student models : what use are they. *Artificial Intelligence tools in education*, pages 73–86.

[Serment, 2007] Serment, J. (2007). *Une infrastructure d'intégration à base d'agents logiciels pour l'élaboration de systèmes d'aide à la décision environnementale : Application à la gestion hydraulique camarguaise*. Thèse de doctorat, Aix-Marseille 3.

[Serment et al., 2008] Serment, J., Espinasse, B., et Tranvouez, E. (2008). Une infrastructure d'intégration orientée agent. *Journal of Decision Systems*, 17(2) :269–300.

[Soller, 2006] Soller, A. (2006). Artificial intelligence methods for modeling and assessing. In *Collaborative Distance Learning Conference CILVR*.

[Stahl et al., 2006] Stahl, G., Koschmann, T., et Suthers, D. (2006). Cambridge handbook of the learning sciences. *Computer-supported Collaborative Learning : An Historical Perspective*, pages 409–426.

[Stone, 2009] Stone, R. (2009). Serious games : virtual reality's second coming ? *Virtual reality*, 13(1) :1–2.

[Suebnukarn et Haddawy, 2004] Suebnukarn, S. et Haddawy, P. (2004). A collaborative intelligent tutoring system for medical problem-based learning. In *Proceedings of the 9th international conference on Intelligent user interfaces*, pages 14–21. ACM.

[Tardif, 1993] Tardif, J. (1993). L'évaluation dans le paradigme constructiviste. *Hivon R.(Dir.) l'évaluation des apprentissages. Réflexions, nouvelles tendances et formation. Sherbrooke : Editions du CRP (Faculté d'éducation, Université de Sherbrooke)*, pages 27–56.

[Tchétagni et Nkambou, 2002] Tchétagni, J. et Nkambou, R. (2002). Hierarchical representation and evaluation of the student in an intelligent tutoring system. In *Intelligent Tutoring Systems*, pages 708–717. Springer.

[Thomas Benjamin et al., 2012] Thomas Benjamin, P., Labat, J.-M., Muratet, M., et Yessad, A. (2012). How to Evaluate Competencies in Game-Based Learning Systems Automatically ? In *11th International Conference in Intelligent Tutoring Systems*, Lecture Notes in Computer Science, pages 168–173. Springer.

[Tricot et Alain, 1999] Tricot, A. et Alain, R. (1999). Modalités et scénarios d'interaction dans des hypermédias d'apprentissage. *Revue des sciences de l'éducation*, 25(1) :105–129.

[VanLehn, 1988] VanLehn, K. (1988). Student modeling. *Foundations of intelligent tutoring systems*, pages 55–78.

[Vidani et Chittaro, 2009] Vidani, A. et Chittaro, L. (2009). Using a task modeling formalism in the design of serious games for emergency medical procedures. In *Games and Virtual Worlds for Serious Applications, 2009. VS-GAMES'09. Conference in*, pages 95–102. IEEE.

[Wang et al., 2003] Wang, J., Lewis, M., et Gennari, J. (2003). A game engine based simulation of the nist urban search and rescue arenas. In *Simulation Conference, 2003. Proceedings of the 2003 Winter*, volume 1, pages 1039–1045. IEEE.

[Weick, 1982] Weick, K. E. (1982). Management of organizational change among loosely coupled elements. *Change in organizations : New perspectives on theory, research and practice*, pages 375–408.

[Wendel et al., 2012] Wendel, V., Gutjahr, M., Göbel, S., et Steinmetz, R. (2012). Designing collaborative multiplayer serious games for collaborative learning. *Proceedings of the CSEDU*, 2012.

[Westra et al., 2009] Westra, J., van Hasselt, H., Dignum, F., et Dignum, V. (2009). Adaptive serious games using agent organizations. In *Agents for Games and Simulations*, pages 206–220. Springer.

[Wooldridge et al., 1995] Wooldridge, M., Jennings, N., et al. (1995). Intelligent agents : Theory and practice. *Knowledge engineering review*, 10(2) :115–152.

[Woolf, 2010] Woolf, B. P. (2010). *Building intelligent interactive tutors : Student-centered strategies for revolutionizing e-learning*. Morgan Kaufmann.

[Yusoff et al., 2009] Yusoff, A., Crowder, R., Gilbert, L., et Wills, G. (2009). A conceptual framework for serious games. In *Advanced Learning Technologies, 2009. ICALT 2009. Ninth IEEE International Conference on*, pages 21–23. IEEE.

[Zyda, 2005] Zyda, M. (2005). From visual simulation to virtual reality to games. *Computer*, 38(9) :25–32.

[Zyda et al., 2003] Zyda, M., Mayberry, A., Wardynski, C., Shilling, R., et Davis, M. (2003). The moves institute's america's army operations game. In *Proceedings of the 2003 symposium on Interactive 3D graphics*, I3D '03, pages 219–220, New York, NY, USA. ACM.

Index des illustrations

I.1 Schéma général du concept de jeu sérieux. 8

II.1 Le concept de jeu sérieux défini selon trois axes. 19

II.2 Modèle de jeux sérieux selon Garris [Garris et al., 2002] 20

II.3 Du jeu vidéo au jeu sérieux [Zyda, 2005] 21

II.4 Diagramme de classes du cadre conceptuel présenté par Yusoff [Yusoff et al., 2009] 21

II.5 Le cadre conceptuel à quatre dimensions pour la conception de jeu sérieux [de Freitas et Jarvis, 2006] 22

II.6 Architecture de TLCTS [Johnson et al., 2004] 26

II.7 Capture d'écran du projet TLCTS [Johnson et al., 2004].
(a) : Le Mission Skill Builder.
(b) : Le Mission Practice Environment. 27

II.8 Capture d'écran du projet FORMAT-STORE [Mathieu et al., 2011].
(a) : Évolution de l'apprenant dans un magasin à travers un avatar.
(b) : Les dialogues des situations-problèmes. 30

II.9 Exemple de vue 3D de l'environnement [Binsubaih et al., 2006] 31

II.10 Extrait d'un arbre CTT [Vidani et Chittaro, 2009]. 34

II.11 Capture d'écran du jeu médical [Vidani et Chittaro, 2009] 35

II.12 Capture d'écran du jeu EFWI [Wendel et al., 2012] 36

II.13 L'interaction des disciplines qui constituent un STI [Burns et Capps, 1989] 40

II.14 Architecture générale d'un STI [Burns et Capps, 1989] 41
II.15 Les différents module de l'ITS proposé par Buche [Buche, 2005] 47
II.16 Processus pédagogique proposé par Buche [Buche, 2005] 48
II.17 HAWAI-DL : Concepts et relations [Amokrane, 2010] 52

III.1 Diagramme d'activité UML de la mission du rôle CODIS 67
III.2 Représentation des différents composants de l'espace d'évaluation 72
III.3 Évolution du score du temps d'exécution selon le delta pour une durée de mission de 30s et 300s 81
III.4 Évolution du score du temps d'inactivité selon le temps d'inactivité . . . 81
III.5 Extraction du graphe référence à partir du modèle du domaine. 84
III.6 Exemple de calculs de couplages pour un exercice de gestion de crise . . 85

IV.1 Architecture générale. 91
IV.2 Des données à l'évaluation 92
IV.3 Les composants du module d'évaluation. 93
IV.4 Représentation UML du modèle BDI. 98
IV.5 Ontologie utilisée pour représenter le modèle du domaine pour la formation à la gestion de crise. 103
IV.6 Structure du modèle du domaine. 104
IV.7 Structure du modèle de l'apprenant. 105
IV.8 Diagramme de séquence UML décrivant les interactions entre les différents agents 106

V.1 Capture d'écran du projet SIMFOR 111
V.2 Architecture générale du projet SIMFOR. 112
V.3 Architecture générale du système. 113
V.4 Capture d'écran de l'édition d'agent (AgJ). 114
V.5 Capture d'écran de l'édition de dialogue. 115
V.6 Exemple d'arborescence d'un dialogue. 116
V.7 Création d'un serveur SIMFOR. 117
V.8 Capture d'écran du serveur SIMFOR. 118
V.9 Rejoindre un serveur SIMFOR. 119

V.10 Interface de sélection du personnage. 119

V.11 Interface de jeu durant la session d'entraînement. 121

V.12 Interface de suivi et d'interaction des AgJ. 122

V.13 Interface détaillée du suivi d'un AgJ. 122

V.14 Suivi des actions durant la session d'entrainement. 123

V.15 Calcul du couplage via le serveur SIMFOR. 123

V.16 Éditeur du modèle du domaine.
(a) : interface principale.
(b) : édition des missions. 124

V.17 Interface principale du moniteur d'évaluation. 125

V.18 Évaluation individuelle et collective du moniteur d'évaluation
(a) : évaluation collective.
(b) : évaluation individuelle de l'acteur *codis*. 125

V.19 Historique du couplage. 126

V.20 Les différents graphes de couplage.
(a) : couplage du réseau (exercice).
(b) : graphe référence (calculer à partir du modèle du domaine).
(c) : graphe différence. 127

V.21 Historique des actions pour l'ensemble des acteurs du scénario. 128

V.22 Moteur de raisonnement de l'AgJ. 129

V.23 Moteur de raisonnement de l'agent évaluateur. 130

V.24 Moteur de raisonnement de l'agent pédagogique. 132

VI.1 Structure d'un scénario SIMFOR. 136

VI.2 Liste des acteurs pour le scénario TMD 138

VI.3 Enchainement des événements et des actions du scénario TMD 138

VI.4 Les participants de l'expérimentation pilote. 140

VI.5 Exemple de formulaire pour évaluer l'apprenant incarnant le rôle du préfet. 141

VI.6 Historique du couplage du réseau pour le premier et le deuxième exercice. 143

VI.7 Séquence d'action durant l'exercice TMD. 143

A.1 Architecture général du moteur de jeu Delta3D. [Darken et al., 2005] . . 157

A.2 Évolution graphique de l'UE 158

A.3 La Sandbox du CryEngine. 160
A.4 L'éditeur Unity 3D. 161

B.1 Interface de l'action téléphone dans SIMFOR 164
B.2 Communication avec un GA. 165
B.3 Historique des appels dans l'interface téléphone. 165
B.4 Interface de l'action téléphone dans SIMFOR. 166
B.5 Les étapes pour envoyer un nouveau fax dans SIMFOR. 166
B.6 L'interface radio dans SIMFOR. 167
B.7 L'interface carte dans SIMFOR. 167
B.8 Édition de la carte dans SIMFOR. 168
B.9 Interface de l'outil la main courante dans SIMFOR. 169
B.10 Fenêtre adresse pour le déplacement automatique dans SIMFOR. 169
B.11 Déplacement en véhicule dans SIMFOR. 170
B.12 Interface de l'outil action dans SIMFOR. 171
B.13 Interface d'aide dans SIMFOR. 172
B.14 Interface information méta-données dans SIMFOR. (a) : Méta-données sur un arbre de l'environnement virtuel. (b) : Méta-données sur un bâtiment de l'environnement virtuel. (c) : Méta-données sur une bouche d'égout de l'environnement virtuel. 172
B.15 Mesure du périmètre de sécurité dans SIMFOR. 172
B.16 Menu action du serveur SIMFOR. 173

Index des tableaux

I.1 Les différents types dévaluation. 11

II.1 La matrice des interactions entre les différents types d'agents [Mathieu et al., 2011]. 29

II.2 Stratégies pédagogiques tactiques implémentées dans HAL. 45

II.3 Tableau comparatif de différents travaux dans le domaine des jeux sérieux (JS) et des systèmes tutoriels intelligents (STI). 60

III.1 les indicateurs pour l'évaluation d'une mission dans l'espace comportemental. 78

III.2 Les actions à réaliser par le CODIS dans le cadre de la mission : *informer les autorités*. 79

III.3 Les actions à réaliser par le CODIS dans le cadre de la mission : *informer les autorités*. 80

III.4 Exemple de calcul de la force d'interaction durant l'exercice. 84

IV.1 Les différentes actions ainsi que leurs effets définis dans le cadre de la formation à la gestion de crise. 99

IV.2 Structure d'une phrase dans un arbre de dialogue. 100

VI.1 Résultats du scénario d'exercice TMD. 142

VI.2 Résultats des formulaires du scénario d'exercice TMD. 142