Etude et définition de mécanismes sémantiques dans les environnements virtuels pour améliorer la crédibilité comportementale des agents : utilisation d'ontologies de services

Kenza Harkouken Saiah

▶ **To cite this version:**

Kenza Harkouken Saiah. Etude et définition de mécanismes sémantiques dans les environnements virtuels pour améliorer la crédibilité comportementale des agents : utilisation d'ontologies de services. Système multi-agents [cs.MA]. Université Pierre et Marie Curie - Paris VI, 2015. Français. NNT : 2015PA066690 . tel-01612932

HAL Id: tel-01612932
https://theses.hal.science/tel-01612932

Submitted on 9 Oct 2017

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

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

THÈSE DE DOCTORAT DE
l'UNIVERSITÉ PIERRE ET MARIE CURIE

Spécialité

Informatique

École doctorale Informatique, Télécommunications et Électronique (Paris)

Présentée par

Kenza HARKOUKEN SAIAH

Pour obtenir le grade de

DOCTEUR de l'UNIVERSITÉ PIERRE ET MARIE CURIE

Sujet de la thèse :

Étude et définition de mécanismes sémantiques dans les environnements virtuels pour améliorer la crédibilité comportementale des agents : utilisation d'ontologies de services

soutenue le 07 octobre 2015
devant le jury composé de :

M. Nicolas Sabouret	Directeur de thèse
Mme. Domitile Lourdeaux	Rapporteur
M. Ronan Querrec	Rapporteur
Mme. Amal El Fallah Seghrouchni	Examinatrice
Mme. Catherine Pelachaud	Examinatrice
M. Julien Saunier	Examinateur
M. Jean-Yves Donnart	Encadrant industriel

Je dédie cette thèse à mes chers parents à qui je dois tout.
À la mémoire de mes grands parents.
À mon cher mari et ma chère fille.
À mes chères sœurs et cher frère.

"Peu de choses sont impossibles à qui est assidu et compétent... Les grandes œuvres jaillissent non de la force mais de la persévérance."
Samuel Johnson

Remerciements

C'est avec une grande satisfaction que je rédige ces remerciements afin d'exprimer ma gratitude à toutes les personnes qui m'ont permis de décrocher le grade de docteur en informatique.

Tout d'abord, je tiens à remercier mon directeur de thèse *Nicolas SABOURET* qui m'a formé et orienté vers la bonne direction afin d'accomplir ma thèse.

Je remercie mon encadrant chez *THALES Training and Simulation Jean-Yves DONNART* qui m'a fait confiance dès le début, et qui m'a prodigué de bons conseils. Je tiens aussi à remercier *Pascal PEYRONET* qui a toujours veillé à ce qu'on ait tous les moyens nous permettant de mener à bien notre travail et à qui nous devons en grande partie la réussite du projet *Terra Dynamica*.

Je remercie *Amal EL FALLAH SEGHROUCHNI* qui m'a fait l'honneur de présider mon jury de thèse, et pour m'avoir soutenu et encouragé pendant les périodes difficiles. Je remercie tous les jurés de ma soutenance de thèse pour l'intérêt qu'ils avaient porté à la lecture de mon manuscrit. Je les remercie pour leurs commentaires sur mon travail et pour les débats qu'ils ont ouverts pendant la soutenance.

Je remercie tous mes camarades avec qui j'ai partagé des pauses café et pauses déjeuner très sympathiques et tant d'autres moments inoubliables et marquants tout au long de mes années de thèse. Je pense bien sûr à *Sabrina*, avec qui j'ai mis au point un protocole de communication très sophistiqué. Je tiens à la remercier pour son aide, ses conseils et ses relectures de plusieurs parties de mon manuscrit. *Brice*, pour sa bonne humeur et son amour pour les bons plats (*c'est trop bon !*). *Patrick*, mon camarade depuis le début, et le chanteur caché du groupe Magic Système. *Simon*, l'homme qui épargnait de l'argent pour se payer une maison de Hobbit. *Quentin*, l'aventurier qui voulait aller en Chine à vélo. *Cyril*, le fan de tatouages. *Etienne*, le représentant officiel de *Sabrina* à l'étranger. *Sylvain*, pour sa sympathie et son humour décalé. *Clément*, le chef d'entreprise à vélo. Je remercie également *Amel, Cédric, Corentin, Costin, Edouard, Emanuel, Feirouz, Hazaël, Javier, Jocelyn, Kevin, Key, Lancelot, Nicolas, Olivier, Shirley, Samir, Sara, Thomas* et *Zach*.

Je remercie mon amie *Abir* pour toutes les discussions et les emails très intellos et intéressants qu'on a partagés. Je remercie mon amie *Yasmine* et son mari pour leur aide inestimable. Je remercie également ma petite chef pâtissière *Yuki* pour tous ses encouragements et sa joie de vivre communicative.

Je remercie les maîtres de conférence et professeurs du *LIP6*, et en particulier ceux de l'équipe *Systèmes Multi-Agents* pour leur sympathie. Je remercie *Zahia GUESSOUM* pour son aide précieuse, sa disponibilité et ses conseils. Je remercie tout le personnel administratif du *LIP6* et de *THALES Training and Simulation*.

Je remercie tous les partenaires du projet *Terra Dynamica* qui ont donné le meilleur d'eux-mêmes pour la réussite du projet et les succès lors des salons Future en seine. Je remercie les ingénieurs qui ont travaillé sur le projet, *Sébastien, Joachim et François*. Je

remercie tout particulièrement *Pierre* pour l'intérêt qu'il avait porté à mon modèle et pour son aide.

Je remercie les financeurs du projet *Terra Dynamica* qui ont cru au projet et qui m'ont ainsi permis de réaliser cette thèse.

Je remercie chaleureusement, tous les membres de ma famille, qui m'ont soutenu tout au long de mes études, et qui sont maintenant soulagés qu'elles soient terminées. Je remercie mes chers parents *Nadia* et *Mehenni* pour tous les sacrifices qu'ils ont fait pour nous et pour avoir toujours été là pour nous soutenir et nous encourager. Je remercie ma grand-mère qui croyait que je passerai ma vie à faire des études. Je remercie, mes chères Sœurs *Lilia* et *Douniazed* et mon adorable petit frère *Abdelhakim* qui stressaient autant que moi pendant les moments difficiles. Je remercie mes neveux *Sidali, Amira* et le petit dernier *Zachary* pour toutes les bêtises qu'ils font chaque jour. Je remercie ma petite fille *Hannah Mariah* d'avoir ensoleillé ma vie et d'avoir participé à sa manière. Je remercie mon cher mari *Redouan* qui a vécu le stresse de la thèse deux fois en comptant la sienne et pour son grand soutien et ses encouragements.

Résumé

Ce travail de thèse se situe dans le cadre du projet *Terra Dynamica* visant à peupler une ville virtuelle avec des agents qui simulent des piétons et des véhicules. L'objectif de notre travail est de rendre l'environnement compréhensible par les agents de la simulation afin qu'ils puissent exhiber des comportements crédibles.

Les premiers travaux qui ont été proposés pour la modélisation sémantique des environnements virtuels gardent toujours un lien de dépendance avec la représentation graphique pré-existante de l'environnement. Cependant, l'information sémantique représentée dans ce genre d'approches est difficilement exploitable par les agents pour effectuer des procédures de raisonnement complexes en dehors des algorithmes de navigation.

Nous présentons dans cette thèse un modèle de représentation de la sémantique de l'environnement qui fournit aux agents des données sur l'utilisation des objets de l'environnement pour permettre au mécanisme d'aide à la décision de produire des comportements crédibles. Par ailleurs, en réponse à des contraintes inhérentes à la simulation urbaine, notre approche est capable de traiter un grand nombre d'agents, en temps réel.

Notre modèle est basé sur le principe que les objets de l'environnement proposent des services permettant de réaliser les actions avec différentes qualités. Nous avons donc représenté les informations sémantiques des objets liées à leur utilisation sous forme de services dans une ontologie de services. Nous avons utilisé cette ontologie de services pour calculer une qualité de service *QoS* qui nous permet de trier les différents objets permettant de réaliser une même action. Ainsi, nous pouvons comparer entre les services offerts par les objets. Ce qui nous permet de proposer aux agents *les meilleurs objets* permettant de réaliser leurs actions afin d'acquérir une *crédibilité comportementale*.

Afin d'évaluer l'impact de notre modèle sur la crédibilité des comportements produits, nous avons défini un protocole d'évaluation dédié aux modèles de représentation de la sémantique dans les environnements virtuels. Dans ce protocole, des observateurs doivent évaluer le caractère crédible des comportements produits par le simulateur à partir d'un modèle sémantique de l'environnement. Grâce à cette évaluation, nous montrons que notre modèle permet de simuler des agents dont le comportement est jugé comme *crédible* par des observateurs humains.

Nous présentons également une évaluation qualitative de la capacité de notre modèle de passer à l'échelle et de répondre aux contraintes d'une simulation temps-réel. Cette évaluation, nous a permis de montrer que les caractéristiques de l'architecture de notre modèle nous permettent de répondre en un temps raisonnable aux demandes d'un grand nombre d'agents.

Mots-clefs

Environnements virtuels sémantiques, Système multi-agents, Représentation des connaissances, Crédibilité comportementale.

Table des matières

1 Introduction **11**
1.1 Contexte Général . . . 11
1.2 Définition de la problématique . . . 12
1.3 Proposition . . . 14
1.4 Organisation de ce mémoire . . . 15

2 État de l'art **17**
2.1 Introduction . . . 17
2.2 Représentation des Environnements Virtuels . . . 17
2.2.1 Modélisation de l'EV avec son modèle sémantique . . . 19
2.2.1.1 Environnement Virtuel Informé . . . 19
2.2.1.2 Représentation selon le domaine d'application . . . 19
2.2.1.3 Représentation selon le niveau sémantique . . . 19
2.2.1.4 Utilisation de la sémantique dans la phase de conception de l'EV . . . 20
2.2.2 Méthodologie de développement d'un EV sémantique . . . 20
2.2.3 Les Objets sémantiques . . . 22
2.2.3.1 Smart Object . . . 22
2.2.3.2 Star Fish . . . 23
2.2.3.3 Conclusion . . . 23
2.2.4 Les solutions avec des ontologies . . . 23
2.2.4.1 Utilisation des ontologies dans la phase de conception . . . 24
2.2.4.2 Représentation sémantique générique de l'EV . . . 26
2.2.4.3 Raisonnement sur le contenu de l'environnement . . . 30
2.2.5 Synthèse . . . 36
2.2.6 Conclusion . . . 38
2.3 Sélection des services de l'environnement . . . 39
2.3.1 Représentation des Services Web Sémantiques avec OWL-S . . . 39
2.3.2 Différentes approches de représentation de la *QoS* . . . 40
2.3.3 Prise en compte des préférences dans la sélection . . . 43
2.4 Crédibilité . . . 44
2.4.1 Évaluation de la crédibilité . . . 45
2.4.2 Conclusion . . . 46
2.5 Conclusion . . . 47

3 Modèle de représentation sémantique d'un environnement virtuel urbain **49**
3.1 Introduction . . . 49
3.2 Principe général du modèle . . . 50

3.3 Formalisme du modèle de représentation de l'environnement 52
3.3.1 Ontologie de Services 53
3.3.1.1 Couche Méta (méta-modèle) 54
3.3.1.2 Définition des Types d'objet 54
3.3.1.3 Définition des services 54
3.3.1.5 Définition des critères d'évaluation des services C_r 56
3.3.2 Qualité de Service QoS 57
3.3.2.1 Paramètres de calcul de la QoS 58
3.3.2.2 Calcul de la QoS 60
3.4 Les interactions du module sémantique avec les modules d'IA 61
3.4.1 Mode interrogatif 61
3.4.2 Mode réactif 63
3.4.3 Mode proactif 64
3.5 Conclusion 64

4 Mise en Œuvre et Évaluation Objective 67
4.1 Introduction 67
4.2 Prototype 67
4.2.1 Présentation générale 67
4.2.2 Fonctionnement et mise en œuvre 68
4.2.3 Premiers résultats et conclusion 69
4.3 Intégration du modèle dans la plate forme du projet Terra Dynamica 70
4.3.1 Le projet Terra Dynamiaca 70
4.3.2 Architecture du projet 70
4.3.3 Contraintes liées au projet TD 73
4.3.4 Interfaces de communication entre le module sémantique de l'environnement et les autres modules de l'architecture du projet 73
4.3.5 Perception de l'environnement 74
4.3.6 Communication avec les modules d'IA de haut niveau 76
4.3.7 Implémentation des trois modes d'interactions entre notre module et les modules d'IA de haut niveau 77
4.4 Fonctionnement du module sémantique de l'environnement 81
4.4.1 Description de l'environnement 81
4.4.1.1 Description d'un Objet sémantique 82
4.4.2 Interface de communication avec l'ontologie de services 83
4.4.3 Implémentation du module sémantique 85
4.5 Conclusion sur la mise en oeuvre 86
4.6 Passage à l'échelle 87
4.6.1 Performances selon le nombre d'agents 87
4.6.2 Performances selon le nombre de concepts dans l'ontologie 88
4.6.3 Performances selon le type d'interaction possible avec l'environnement 89
4.6.3.1 Avec le mode Réactif 89
4.6.3.2 Avec le mode Pro-actif 90
4.6.4 Capacité de la simulation avant ralentissement 90
4.7 Conclusion sur l'évaluation des performances 91

5 Évaluation Subjective **93**
5.1 Objectif . . . 93
5.2 Hypothèses Générales . . . 94
5.2.1 Crédibilité dans les choix . . . 95
5.2.2 Crédibilité des comportements des agents . . . 95
5.2.3 Crédibilité des interactions avec l'environnement . . . 95
5.3 Protocole . . . 95
5.3.1 Principe . . . 96
5.3.1.1 Évaluation selon le contexte . . . 96
5.3.1.2 Évaluation selon les modes d'interaction . . . 97
5.3.2 Configuration de notre modèle pour générer les vidéos . . . 97
5.3.3 Description du questionnaire d'évaluation . . . 98
5.3.3.1 Page d'accueil . . . 98
5.3.3.2 Évaluation du paramètre "Difficulté" . . . 99
5.3.3.3 Évaluation du paramètre "Proximité" . . . 100
5.3.3.4 Évaluation du mode "Réactif" . . . 101
5.3.3.5 Évaluation du mode "Proactif" . . . 102
5.3.4 Réalisation des scénarios d'évaluation . . . 103
5.4 Résultats . . . 110
5.4.1 Hypothèses sur la crédibilité dans les choix . . . 110
5.4.2 Hypothèses sur les paramètres de calcul de la QoS . . . 112
5.4.2.1 Paramètre de « Difficulté » . . . 112
5.4.2.2 Paramètre de « Proximité » . . . 113
5.4.3 Hypothèses sur la crédibilité des comportements des agents . . . 115
5.4.4 Hypothèses sur la crédibilité des interactions avec l'environnement . . . 118
5.5 Discussion . . . 119
5.6 Conclusion . . . 122

6 Conclusion **125**

A Questionnaire de l'évaluation subjective **133**

Table des figures **141**

Liste des tableaux **143**

Bibliographie **145**

Chapitre 1

Introduction

1.1 Contexte Général

Les environnements virtuels permettent de mettre en situation des scénarios pour différents objectifs. Ils peuvent être utilisés pour l'apprentissage [Buche et al., 2010], l'entrainement [Barot et al., 2013], le divertissement [Forbus and Wright, 2001], etc. Leur réalisme et leur qualité graphique dépendent du type d'application auquel ils sont dédiés. Cependant, les environnements virtuels doivent être crédibles, où leur crédibilité repose sur la crédibilité comportementale des agents qu'ils mettent en scène [Loyall, 1997]. Nous qualifions de comportement crédible tout comportement qu'un être humain pourrait réaliser s'il se trouvait dans la même situation que l'agent, ce qui lui donnerait une *illusion de vie* [Barbosa et al., 2010].

Le projet Terra Dynamica dans lequel s'inscrit cette thèse a pour but de peupler une ville virtuelle avec des agents qui simulent les piétons et les véhicules (représenter la vie dans la ville). Ces derniers doivent avoir des comportements crédibles dans le sens où ils doivent reproduire des activités habituelles des habitants tout en étant capables de réagir aux événements survenant dans leur environnement. Ce projet a plusieurs cadres d'application dans les domaines de sécurité, d'urbanisme, de transport et des jeux vidéo. Par exemple, dans le domaine de la sécurité, la préfecture de police peut simuler une manifestation sur un parcours donné pour étudier l'impact sur la circulation, les risques de sécurité et les endroits stratégiques pour placer leurs agents. La simulation d'une telle situation dans un environnement virtuel (EV) permet d'observer le mouvement de foule en faisant varier des paramètres comme le nombre de manifestants, leur agressivité, le type de parcours et sa longueur, etc.

Les agents de l'environnement ne peuvent avoir de comportements crédibles que s'ils peuvent capter sa sémantique qui va le rendre compréhensible, ainsi ils pourront interagir d'une manière adaptée avec lui [Bates, 1994]. Dans notre travail, nous cherchons donc à représenter la sémantique de l'environnement utile pour les personnages qui le peuplent afin de leur permettre d'adopter des comportements crédibles.

Les premiers travaux qui ont été proposés pour la modélisation sémantique des environnements virtuels gardent toujours un lien de dépendance avec la représentation graphique de l'environnement. Ces différentes approches se basent généralement sur une représentation de l'EV pré-existante (par exemple, en utilisant des systèmes d'information géographiques pour construire des EV urbains [Linturi et al., 2000]), ou bien en ajoutant des annotations sémantiques [Maass and Dollner, 2006]. Cependant, le type d'information sur l'environnement virtuel représenté dans ce genre d'approches est le plus souvent utilisé

pour faire du *path finding* [Donikian, 1997]. Ainsi, ce type d'information sémantique est inexploitable par les agents pour faire des procédures de raisonnement de haut niveau.

Le développement d'un nouveau type d'environnement appelé Environnement Virtuel Sémantique (EVS) [Otto, 2005] va alors prendre son essor, et donner lieu à une nouvelle approche de conception des environnements virtuels [Pellens et al., 2005]. Les EVS proposent des informations sémantiques et symboliques aux agents afin qu'ils puissent avoir des interactions plus intelligentes avec leur environnement. En effet, les EVS fournissent une représentation cohérente (i) du monde simulé et du comportement de ses entités, (ii) des interactions et des tâches que les utilisateurs et les agents peuvent effectuer dans l'environnement, et (iii) des éléments de connaissances que les agents peuvent utiliser pour prendre des décisions [Stanney, 2002].

Le déplacement de la sémantique au niveau de l'environnement pose alors un difficile compromis entre l'autonomie des agents, c'est-à-dire que les agents ont le total contrôle de leur état interne tout en étant capables de prendre des décision et faire de la planification d'actions [Perlin and Goldberg, 1996], et le fait que les agents soient guidés par leur environnement [Abaci et al., 2005]. A cela s'ajoute le problème de la crédibilité des comportements des agents qui évoluent dans l'environnement. En effet, les agents doivent choisir dans l'environnement les objets avec lesquels ils interagissent de manière à donner un sentiment de réalisme à l'observateur de la simulation. Doyle [Doyle, 2002] considère que les connaissances de l'environnement ont une grande part dans la crédibilité des comportements des agents pour plusieurs raisons. La première est de permettre aux agents d'avoir une bonne représentation de leur environnement. Ça leur permet par la suite d'utiliser les bons éléments de l'environnement pour atteindre leur but. En effet, l'interprétation du contenu est fourni par le concepteur de l'environnement et donc, les objets de l'environnement ont forcément un sens et permettent d'atteindre un but connu par le concepteur de l'environnement. Les agents devraient alors les utiliser selon le but pour lequel ils ont été conçus dans l'environnement. C'est-à-dire qu'en représentant clairement le sens et le but du contenu aux agents, ils pourront donc atteindre leurs objectifs. La deuxième raison est de permettre aux agents d'interpréter leurs perceptions. En effet, les perceptions de l'agent ne déterminent pas si oui ou non l'action qu'il va réaliser avec un objet de l'environnement est celle à laquelle l'objet est destiné de faire. En fournissant des connaissances sur l'environnement, nous permettons aux agents de se comporter de manière conforme à la fonction des objets de l'environnement.

Notre travail se situe dans l'objectif de permettre aux agents de l'environnement d'avoir des comportements crédibles à travers un modèle de représentation sémantique de l'environnement, dans le cadre des applications du projet Terra Dynamica. Le modèle que nous proposons doit être suffisamment générique pour être appliqué aux entités hétérogènes qui composent l'environnement, et utilisable dans une simulation contenant un grand nombre d'agents (passage à l'échelle).

1.2 Définition de la problématique

La représentation sémantique des environnements virtuels pour que les personnages virtuels [Kalra et al., 1998] qui le peuplent aient des comportements crédibles soulèvent des problématiques de recherche générales, communes à une grande partie des modèles de représentation des EV et auxquelles nous ferons face dans cette thèse.

Rendre les EVs plus intelligents

Les environnements virtuels intelligents (EVI) [Aylett and Luck, 2000] interagissent avec les utilisateurs d'une manière intelligente, c'est-à-dire, qu'ils ont la capacité d'aider les utilisateurs à résoudre des problèmes spécifiques, plutôt que de compter uniquement sur les connaissances et les compétences des utilisateurs. Ils sont peuplés *d'agents virtuels intelligents* (ou cognitifs) possédant des *comportements intelligents*. Un comportement intelligent permet à un agent situé de survivre au mieux dans son environnement [Steels, 1994]. Les agents intelligents ont des capacités de représentation symbolique et de raisonnement sur leur propre comportement et leurs propres objectifs, les rendant ainsi capables de reproduire les comportements habituels des humains [Aylett and Luck, 2000]. Cette capacité de raisonnement n'est possible qu'à travers un modèle de représentation sémantique de leur environnement qui leur fournit des informations sur les objets virtuels, comme leur type, leur fonction, leur structure, leur état, leurs relations avec les autres, et les opérations qu'ils peuvent appliquer sur eux. Ces informations sémantiques sur l'environnement leur permettent de faire des raisonnements de haut-niveau comme pour la planification et l'anticipation des conséquences de leurs actions [Aylett and Cavazza, 2001]. En effet, une partie de *« l'intelligence »* des agents peut être déportée au niveau de l'environnement [Magnin, 1996]. Cela veut dire qu'une partie des calculs menant à la réalisation de l'action soit gérée par l'environnement [Ferber, 1995]. L'environnement centralise donc une partie des connaissances qui sont partagées par tous les agents. Cette centralisation facilite le calcul de l'état de l'environnement par rapport aux différentes demandes d'action des agents, et assure la cohérence des comportements observés des agents par rapport aux réactions attendues dans l'environnement.

Nous souhaitons simuler dans le projet Terra Dynamica des personnages virtuels dotés d'une intelligence artificielle qui fait participer plusieurs composants (parmi ces composants se trouvent un système de perception, de décision et d'action) leur permettant d'accomplir leur rôle de manière à pouvoir reproduire les activités habituelles des habitants d'une ville. Les défis technologiques du projet Terra Dynamica sont nombreux, engendrant plusieurs contraintes auxquelles notre modèle se doit de répondre et qui sont définies comme suit :

Agents autonomes avec des comportements crédibles

Le besoin de simuler des comportements humains dans des environnements virtuels peuplés de personnages animés a donné naissance à ce qu'on appelle des agents crédibles (*Believable Agents* [Bates, 1994]). Pour que l'agent ait des comportements crédibles il doit regrouper plusieurs caractéristiques dont l'autonomie, qui veut dire que l'agent a un total contrôle de son état interne et il est capable de faire de la planification d'actions et de prendre des décisions [Perlin and Goldberg, 1996]. L'agent doit aussi être réactif, c'est-à-dire, qu'il doit réagir en un temps raisonnable aux événements qui opèrent dans son environnement tout en étant capable de s'adapter aux changements qui peuvent y survenir. Et enfin, l'agent doit avoir des **choix d'action pertinents**, c'est-à-dire que toutes les actions réalisées par l'agent doivent aboutir à un ou plusieurs objectifs, tout en ayant un sens aux regards des observateurs [Perlin and Goldberg, 1996]. La compréhension de l'environnement joue un rôle important dans la crédibilité des comportements des agents. Cela leur permet d'intégrer les connaissances sur leur environnement dans leur processus de décision. Ainsi, ils pourront planifier leurs actions tout en ayant la capacité d'anticiper leurs conséquences [Aylett and Cavazza, 2001]. Nous devons donc proposer un modèle

qui permet de représenter les informations sémantiques de l'environnement et déduire les informations utiles pour l'aide à la décision des agents, afin qu'ils puissent faire des choix d'objets crédibles tout en étant en adéquation avec leur situation.

Passage à l'échelle

Le projet Terra Dynamica exige un traitement en temps-réel [Benyahia, 1993], [Ingrand et al., 1992] pour que les actions des agents suivent le cours de la simulation et pour que les utilisateurs puissent se sentir immergés. La simulation d'une ville virtuelle implique forcément la contrainte de pouvoir simuler un grand nombre d'agents en temps-réel, dans un environnement complexe composé d'un grand nombre d'objets, parmi lesquels des agents actifs qui vont le modifier. Par conséquent, l'environnement sera modifié par de nombreux agents, et les personnages devront percevoir ces changements. L'environnement doit alors fournir aux agents le sens à associer à leurs perceptions [Kallmann, 2001] afin de les intégrer dans leur le processus de décision et de planification. De plus, en rejoignant la caractéristique de réactivité des agents crédibles [Perlin and Goldberg, 1996], notre système de simulation doit avoir un temps de réponse rapide aux événements et retourner une réponse au bon moment afin de permettre aux agents d'êtres réactifs.

Généricité

Le projet Terra Dynamica a plusieurs domaines d'application allant des jeux vidéos en passant par l'urbanisme et la sécurité, etc, dont le principale reste la simulation urbaine. Nous devons alors proposer un modèle **générique**. Par générique, nous voulons dire adaptable aux différents domaines d'application mais aussi applicable à tous les types d'entités qui composent l'environnement (objets hétérogènes proposant des services différents). Cependant, il est impossible de modéliser tout et n'importe quoi, dans n'importe quel domaine d'application. Il n'est néanmoins pas impossible de réutiliser le principe de représentation de l'environnement virtuel et de l'adapter à un autre domaine d'application.

Validation

Notre modèle sémantique de l'environnement doit être capable de proposer les meilleurs objets pour réaliser les actions des agents afin que ces derniers puissent exhiber des comportements que des observateurs humains peuvent juger comme crédibles. Cette affirmation ne semble jamais avoir été vérifiée par un protocole d'évaluation pour des modèles de représentation de l'environnement afin de vérifier qu'ils permettent bien de rendre l'environnement compréhensible par les agents pour qu'ils puissent avoir des comportements crédibles. Nous devons donc mettre en place un protocole d'évaluation dédié à cette tâche.

1.3 Proposition

Nous proposons un modèle de sélection des services de l'environnement afin de permettre à chaque agent de choisir le meilleur objet qui propose le service répondant le mieux à leur action de telle sorte qu'ils aient des comportements crédibles.

Afin que les agents soient intelligents, ils ont besoin d'informations sémantiques qui leur permettent de comprendre le sens des objets, les relations entre eux et les interactions qu'ils peuvent avoir avec ces objets [Aylett and Cavazza, 2001]. Pour cela, nous proposons de modéliser cette information sémantique de l'environnement et déduire des informations

pour faciliter le processus de planification et l'aide à la décision des agents pour accomplir d'une manière crédible leurs actions. Nous avons utilisé le formalisme ontologique qui permet la spécification d'un vocabulaire de représentation pour un domaine partagé définissant les classes, les relations, les fonctions, et d'autres objets [Gruber, 1993]. Les ontologies soutiennent le partage et la réutilisation des connaissances formellement représentées dans les systèmes d'IA. Elles permettent de définir un vocabulaire commun dans lequel les connaissances partagées sont représentées.

Les techniques de modélisation sémantique ont connu un grand essor dans le domaine des systèmes d'informations, et le Web Sémantique [Berners-Lee et al., 2001] est un bon exemple d'utilisation d'ontologies. La représentation sémantique des services web permet leur découverte et leur comparaison grâce au calcul d'une qualité de service QoS. Ainsi, il permet aux informations dénuées de sens stockées dans des pages Web à être traitées par des algorithmes et recherchées en fonction de leur contenu.

Notre modèle de représentation sémantique de l'environnement s'inspire des techniques utilisées dans le domaine du Web Sémantique. Les services représentent les fonctions des objets de l'environnement et la QoS représente la qualité de service proposée par un objet pour exécuter une action donnée. Cette qualité de service nous permet de trier les objets qui permettent de répondre à une même action et d'aider les agents à choisir le meilleur d'entre eux pour exécuter leurs actions.

Ce formalisme de représentation des services de l'environnement nous permet d'appliquer le même algorithme de calcul de la QoS pour les différents services proposés par des objets hétérogènes, ce qui nous permet de satisfaire l'objectif de généricité de notre approche.

La partie du traitement pour réaliser une action consiste à interpréter les perceptions des agents en focalisant leur attention sur les objets qui les intéressent. Ceci est fait en calculant la qualité du service recherché pour les types d'objet qui peuvent répondre à l'action désirée.

Nous proposons donc de fournir aux agents des données abstraites possédant déjà une signification. Ainsi, nous éviterons le coût en terme de temps de traitement que peut engendrer la vision artificielle [Farenc et al., 1999b], ce qui nous permet de satisfaire notre objectif de simuler un grand nombre d'agents.

Afin de vérifier que notre modèle permet bien de rendre l'environnement compréhensible par les agents pour qu'ils puissent avoir des comportements crédibles, nous proposons un protocole de validation où nous proposons à des observateurs d'évaluer le caractère crédible des comportements produits par le simulateur à partir de notre modèle.

Nous proposons donc dans cette thèse un modèle de représentation de la sémantique de l'environnement pouvant être utilisé pour la définition sémantique d'objets variés pour différents scénarios d'utilisation, pour produire des données d'aide à la décision qui permettrons aux agents d'adopter des comportements crédibles.

1.4 Organisation de ce mémoire

Cette thèse est organisée comme suit. Dans le prochain chapitre, nous présenterons en détail l'état de l'art des différents domaines de recherche abordés dans cette thèse. Tout d'abord nous parlerons des environnements virtuels et des différentes approches de modélisation de leur sémantique. Nous parlerons par la suite des techniques de modélisation de la sémantique dans le web sémantique dont s'inspire notre approche. Nous terminerons en abordant la crédibilité comportementale et les méthodes de son évaluation dans les environnements virtuels.

Le troisième chapitre est focalisée sur la description des contributions de cette thèse. Nous commencerons par présenter notre modèle qui consiste à représenter dans une ontologie de services les différents services proposés par les objets qui composent l'environnement. Cette ontologie de services nous permettra par la suite de calculer la QoS qui nous servira pour trier les objets proposant le service recherché pour exécuter une action. Nous détaillerons aussi les algorithmes des différents modes d'interactions des agents avec leur environnement.

Dans le chapitre quatre, nous donnerons les détails concernant la mise en œuvre de notre modèle, et son évaluation objective. Il s'agira principalement de donner des informations sur le contexte de notre travail, c'est-à-dire, le logiciel développé dans le cadre du projet Terra Dynamica dans lequel nous avons intégré notre modèle. Nous présenterons les différentes interfaces que nous avons développées permettant de comprendre la manière dont fonctionne le système. L'évaluation objective de notre modèle consiste à mesurer expérimentalement des valeurs caractéristiques du modèle et de vérifier de manière objective leur signification. Nous vérifions ainsi la contrainte liée au passage à l'échelle après l'ajout de chaque mode d'interaction proposé.

Le chapitre cinq sera consacré à l'évaluation subjective afin de valider le modèle développé au cours de cette thèse. Cette évaluation consiste à faire appel à des observateurs externes du système, à qui nous demanderons de donner leur propres appréciations sur les comportements des agents qu'ils vont observer. Le but de cette évaluation subjective est de valider l'apport en crédibilité que va apporter notre modèle à la simulation de l'EV.

Pour conclure ce document, nous ferons une synthèse de l'ensemble du travail réalisé dans le cadre de cette thèse, et nous donnerons les perspectives majeures qui peuvent être explorées dans la continuité de nos travaux de recherche.

Chapitre 2

État de l'art

2.1 Introduction

Ce chapitre a pour objectif de présenter les différents travaux liés aux champs de recherche de cette thèse. Celui-ci est organisé de la manière suivante.

Dans un premier temps nous aborderons les différentes approches de la représentation de l'information sémantique dans les environnements virtuels. Nous nous attarderons sur les approches basées sur les ontologies qui nous intéressent plus particulièrement. Puis, nous verrons comment est représentée la sémantique des entités de l'environnement qui permettent leur sélection. Enfin, nous donnerons les architectures de représentation de la qualité de service dans le domaine du web sémantique dont nous nous sommes inspiré.

En second lieu, nous allons nous intéresser à la crédibilité dans les environnements virtuels qui est liée à la crédibilité des comportements des agents dans un système de simulation. Celle-ci représente l'un des objectifs à atteindre avec le modèle proposé dans cette thèse, et nous terminerons par une conclusion.

2.2 Représentation des Environnements Virtuels

La représentation des connaissances pour les environnements virtuels (EV) est un problème qui a pris de l'importance ces dernières années [Aylett and Cavazza, 2001]. En effet, les agents évoluant dans un EV ont besoin d'information sémantique sur celui-ci pour adapter leurs comportements aux éléments identifiables de cet environnement et faire des choix d'action [Tutenel et al., 2008].

Les Environnements Virtuels Sémantiques (EVS) [Otto, 2005] ont été proposés pour palier ce manque d'informations sémantiques (connaissances sur l'environnement) nécessaires pour rendre possible une interaction plus intelligente entre les agents et leur environnement.

Nous nous intéressons dans cet état de l'art à l'information sémantique dans les environnements virtuels et les différentes méthodes qui ont été adoptées par les équipes de recherche pour les représenter. Comme cela à été souligné par Ibáñez et son équipe [Ibáñez and Delgado-Mata, 2011], il n'y a pas de standard de modélisation de la sémantique des EVs, chaque équipe a développé son propre modèle de représentation. Néanmoins, dans [Ibáñez and Delgado-Mata, 2011], ils ont identifié trois principales approches :

– **Modélisation de l'EV avec son modèle sémantique :** dans cette approche, la génération du modèle sémantique de l'environnement est indépendante de la re-

présentation de ses entités [Soto and Allongue, 1997], [Doyle, 2002], [Yong et al., 2006].
Soto & Allongue proposent une approche dans ce sens, pour accroître l'interopérabilité dans les environnements virtuels (Projet *NOVAE* visant à la définition des composants d'une architecture ouverte pour soutenir l'interopérabilité des mondes virtuels [Soto and Allongue, 1997,Soto and Allongue, 1998]). Celle-ci repose sur une ontologie pour définir les connaissances communes dans les mondes virtuels, et un modèle *influence/réaction* pour décrire les comportements des entités virtuelles et les réactions du monde virtuel.
Les auteurs étendent le modèle *influence/réaction* [Ferber and Muller, 1996] (développé dans les systèmes multi-agents) à la notion de ressource. Une ressource représente une caractéristique sémantique forte du monde virtuel et peut être utilisée par le concepteur afin d'exprimer une contrainte ou une propriété commune à de nombreuses entités virtuelles. Cette approche a été reprise et décrite plus en détail dans [Soto and Allongue, 2002].

– **Construction de l'EV en se basant sur un niveau sémantique pré-existant :** il s'agit de compléter une représentation de l'EV existante et de tirer profit de tout ce qui existe déjà comme information sémantique. Par exemple, exploiter un SIG (Système d'Information Géographique) qui a été utilisé pour la reproduction d'une ville virtuelle et le compléter par de l'information sémantique relative aux connaissances de la ville pour développer une ville digitale [Linturi et al., 2000], [Dokonal et al., 2004].

– **Ajouter de l'annotation sémantique à un modèle existant de l'EV :** le principe est de compléter la représentation avec des annotations sémantiques qui n'ont de sens que pour les utilisateurs, mais qui restent incomprises par le système. Par exemple, dans [Loughlin and Hughes, 1994], ils proposent d'annoter les modèles de fluides 3D en utilisant le principe des post-it. Ils incorporent des marqueurs associés au texte dans un environnement virtuel qui seront accessibles pour d'autres utilisateurs.
Un autre type d'annotation est proposé dans [Harmon et al., 1996], qui proposent une annotation vocale dans un projet de livre virtuel.

Dans le cas de notre projet, le niveau sémantique pré-existant (SIG et les données géométriques et cartographiques qui nous ont permis de générer la représentation 3D de la ville) n'est pas assez riche sémantiquement pour qu'il soit exploité dans notre représentation sémantique de l'environnement.

Notre but n'est pas de rendre l'environnement compréhensible par les utilisateurs du système, mais de le rendre compréhensible par les agents du système pour qu'ils puissent exploiter cette information sémantique qui rendra leurs comportements crédibles.

Nous n'allons donc pas nous intéresser aux deux derniers types d'approches, mais plutôt à la première approche de modélisation de l'EV. Nous proposons alors de générer un modèle de représentation de l'environnement indépendant de la représentation physique de ses objets.

2.2.1 Modélisation de l'EV avec son modèle sémantique

2.2.1.1 Environnement Virtuel Informé

Parmi les premiers travaux qui ont introduits la sémantique dans la représentation des environnements virtuels, nous retrouvons les travaux de Thalmann et son équipe [Farenc et al., 1999b]. Les environnements virtuels informés (IVE) *Informed Virtual Environent* offrent toutes les informations sémantiques et géométriques nécessaires pour la simulation d'humains virtuels ayant des comportements cohérents, notamment pour la navigation et la planification d'actions. L'approche proposée dans [Farenc et al., 1999b] est basée sur une décomposition hiérarchique d'une scène urbaine dans ce qu'ils appellent *Environment Entities* (ex., passage pour piétons, immeuble, etc). Ces entités de l'environnement fournissent des informations géométriques ainsi que des notions sémantiques (signification des couleurs d'un feu tricolore). Ces informations sont contenues dans une base de données, qui va permettre par la suite d'alimenter une méthode de construction des scènes virtuelles. Ce modèle de données de l'environnement virtuel informé a été utilisé dans une application web permettant l'animation d'humanoïdes virtuels, comme c'est décrit dans [Farenc et al., 1999a].

Les solutions proposées dans cette approche de modélisation de l'EV avec son modèle sémantique peuvent êtres classées dans trois autres catégories qui dépendent du type, du niveau et de l'utilisation de l'information sémantique à représenter.

2.2.1.2 Représentation selon le domaine d'application

Quelques travaux de la littérature qui ont modélisé la sémantique de l'environnement ont été dédiés à des applications spécifiques, ce qui a conduit à des représentation sémantiques dépendantes du domaine d'application [Farenc et al., 1999b]. Dans VUEMS [Donikian, 1997], ils proposent une approche basée sur les IVE en connectant différents niveaux de représentation de l'environnement (ville virtuelle), et en assemblant des données géométriques, topologiques et sémantiques pour la simulation du trafic de véhicules.

Le seul avantage de ces représentations est la bonne maitrise du langage de modélisation par les concepteurs. Le principal inconvénient est que ces modèles sémantiques dépendent fortement de leur domaine d'application donc de leur plateforme de conception ce qui les rend difficiles à étendre.

2.2.1.3 Représentation selon le niveau sémantique

[Chevaillier et al., 2011] considèrent qu'il existe deux catégories de représentations sémantiques pour les EVs correspondant à deux niveaux de sémantique.

– **La représentation orientée-contenu** vise à faire une représentation explicite du contenu de l'environnement afin de reproduire les scènes. De ce fait, nous avons un accès direct aux définitions des entités de l'environnement. Ces représentations concernent essentiellement l'information géométrique et topologique de l'environnement. L'information sémantique concerne les types, les propriétés, les structures et les relations des entités de la scène. Plusieurs approches ont été proposées pour faire cette représentation [Gutierrez et al., 2005]. La plupart de ces solutions proposent une représentation ontologique contenant différents types d'informations sur l'environnement pour différents types d'application [Grimaldo et al., 2006a]. Ce type

d'approches restent indépendants du domaine d'application.

– **La représentation orientée-système** est motivée par les questions relatives à l'ingénierie des systèmes. Ce type de représentation vise à faire le lien entre les différentes composantes de l'application (contenu graphique et le système d'exécution) tout en gardant l'indépendance des unes des autres [Tutenel et al., 2009].
Bowman et son équipe [Bowman et al., 2003] proposent d'intégrer les informations sémantiques abstraites dans la représentation de l'EV. Ce type d'environnements sont appelés *Information-rich virtual environments (IRVE)*. Ils combinent les techniques de visualisation de l'information et les EVs, augmentant ainsi l'EV avec des informations supplémentaires abstraites. Dans cette approche, les auteurs proposent d'encapsuler dans des nœuds de graphe réutilisables appelés *Objets Sémantiques*, les informations abstraites de l'EV sous la forme de texte, d'images, de ressources audio, vidéos, ou des liens hypertextes. L'information abstraite de l'EV est décrite en utilisant VRML et les nœuds X3D PROTOTYPE. Lors de l'exécution, l'accès aux objets sémantiques est obtenu au moyen d'API (Application Program Interfaces).

2.2.1.4 Utilisation de la sémantique dans la phase de conception de l'EV

Cela consiste à introduire de l'information sémantique dans la phase de modélisation conceptuelle des EVs, avec l'objectif d'éviter de nombreux obstacles qui empêchent une propagation rapide de ce type d'applications [De Troyer et al., 2003].

Cette abstraction des questions de mise en œuvre en impliquant les experts de domaine réduit la complexité de développement et n'exige aucune connaissance approfondie des EVs. Elle permet aussi de détecter d'une manière précoce les erreurs de conception en ayant une communication plus facile avec les parties prenantes au développement de l'EV.

[Vanacken et al., 2007] soutiennent cette approche, et considèrent que l'introduction de l'information sémantique en ajoutant des annotations sémantiques à la représentation des EVs existante était une tâche fastidieuse très couteuse en temps. De plus, cela impliquait une dépendance avec le systèmes sous-jacent de l'EV.

Comme le modèle conceptuel de l'EV est indépendant du système, cette approche permet d'avoir un découplage entre la couche système et la couche sémantique de l'EV.

Notre objectif est d'avoir un modèle de représentation sémantique indépendant du domaine d'application et de la couche système. Cet objectif est lié à la contrainte du projet qui considère plusieurs types d'applications (jeu vidéo, apprentissage, sécurité, etc). Nous avons donc choisi une approche qui nous permet de représenter l'environnement tout en ayant une couche sémantique indépendante de la couche système de l'EV. Ce type de représentation nous permet d'enrichir la représentation avec les connaissances du domaine sans pour autant toucher à la couche système de l'environnement.

Nous parlerons dans la suite de ce cette section des différents travaux qui abordent cette problématique en suivant l'approche de modélisation de l'EV avec son modèle sémantique.

2.2.2 Méthodologie de développement d'un EV sémantique

Chevaillier et son équipe [Chevaillier et al., 2009] proposent un méta-modèle d'environnement virtuel informé et structuré défini comme une extension d'UML 2.1. Cette représentation des connaissances de l'environnement a pour but de rendre les connaissances

de l'environnement explicites pour les agents et les utilisateurs du système. Ils proposent donc une méthodologie pour développer ce type d'EV sémantiquement riche.

Chevaillier et son équipe proposent dans le projet MASCARET [1] un profile UML adapté appelé VEHA (Virtual Environment supporting Human Activities) . Il permet de concevoir des modèles d'environnements métiers indépendants des plates-formes, pouvant ensuite être exécutés en temps réel sur différentes plates-formes de réalité virtuelle.

VEHA couvre la modélisation des propriétés sémantiques, structurelles, géométriques et topologiques des entités de l'environnement virtuel ainsi que leurs comportements réactifs. Chevaillier et al. considèrent que l'intérêt de proposer une approche basée sur UML est que son méta-modèle couvre différents points de vue de la modélisation sémantique de l'EV : l'ontologie, la structure, les comportements, les interactions, les actions. UML est également un langage extensible qui peut être spécialisé pour fournir une sémantique opérationnelle formelle.

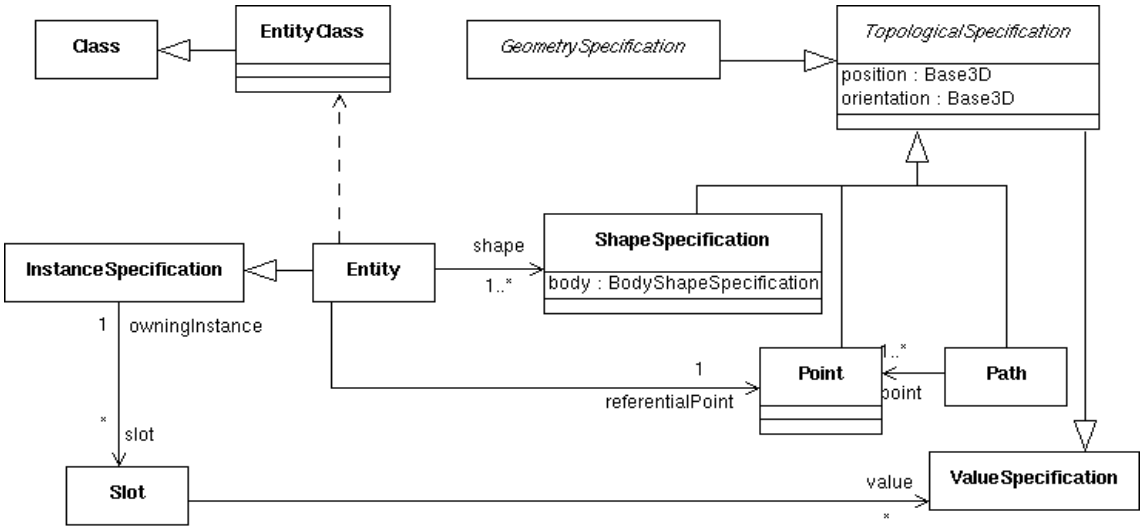


Figure 2.1 – Méta-modèle pour la modélisation des concepts du domaine dans Mascaret [Chevaillier et al., 2009]

Nous pouvons voir dans la figure 2.1 une vue conceptuelle d'une partie du méta-modèle du projet Mascaret représentant les concepts du domaine. EntityClass est une méta-classe, dérivée de la classe Classe du méta-modèle d'UML. Elle représente une famille de concepts du domaine. À son tour, un concept d'un domaine particulier est appelé Entity. Une entité est une instance de la méta-classe EntityClass, dérivée de la classe InstanceSpecification du méta-modèle d'UML.

Le framework de MASCARET a été utilisé pour développer des applications riches en contenu interactif qui peuvent être déployées sur diverses plates-formes d'EV (tels que les simulations multi-agents [Septseault, 2007,Marion et al., 2007] ou des applications portant sur le patrimoine culturel [Barange et al., 2011]).

1. MASCARET est un système multi-agents pour les environnements collaboratifs, adaptatifs et réalistes dédiés à la formation. Initialement, il a été consacré au développement d'environnements virtuels pour l'apprentissage humain [Querrec, 2002, Chevaillier, 2006, Querrec, 2010]

Conclusion

L'utilisation d'une extension d'UML dans le projet MASCARET permet grâce à son méta-modèle de couvrir les différents points de vue de la modélisation sémantique de l'environnement virtuel. Ainsi, ils ont modélisé l'environnement, les agents et leurs actions en utilisant le même langage.

Cette représentation de la réalité virtuelle va au-delà du but de rendre l'environnement compréhensible par les agents et inclue les utilisateurs du système. D'un autre côté, cette représentation permet de modéliser en plus des propriétés sémantiques de l'environnement, les propriétés structurelles, géométriques et topologiques des entités de l'environnement virtuel ainsi que leurs comportements réactifs. De ce fait, cette représentation de l'environnement couvre des connaissances adaptées aux applications riches en contenu interactif.

Dans notre modèle, nous souhaitons rendre le contenu de l'environnement compréhensible par les agents, pour cela, nous avons considéré que l'utilisation du formalisme ontologique était adapté pour définir les différents concepts du monde et leurs relations pour nous permettre d'appliquer des raisonnements.

2.2.3 Les Objets sémantiques

L'ensemble des travaux présentés ci-après proposent d'enrichir la description des objets par de l'information sémantique afin de faciliter les interactions entre les agents et leur environnement.

2.2.3.1 Smart Object

Les Environnements virtuels informés utilisent les **Smarts Objects** [Kallmann and Thalmann, 1999] pour interagir avec les entités de l'environnement.

Kallmann & Thalmann proposent l'idée d'inclure dans la description de l'objet toutes les informations nécessaires pour décrire la façon d'interagir avec lui. Ainsi, nous pouvons distinguer des informations sur : (i) les propriétés de l'objet (sémantiques et physiques), (ii) comment interagir avec l'objet (actions, positions, gestes), (iii) le comportement de l'objet en réponse à une action et (iv) le comportement des agents pour réaliser l'interaction. Par exemple, la poignée d'une porte sert pour ouvrir ou fermer la porte, cette partie de la porte a une indication sur l'endroit où l'utilisateur doit mettre la main afin d'interagir avec cette partie. Cette indication définie la forme de la main et l'emplacement prévu afin de procéder à une action. Malheureusement, les smarts objects, tel qu'ils sont présentés par Kallmann, sont strictement destinés à l'animation et ne permettent pas de faire du raisonnement de haut-niveau tel que pour allouer aux agents la capacité d'adapter leurs comportements aux conséquences de leurs actions.

Les Smarts Objects ont été principalement utilisés pour l'animation du comportement des agents. Abaci et al. [Abaci et al., 2005] ont proposé d'étendre cette solution en permettant aux agents de faire un raisonnement de haut-niveau sur la description sémantique des objets (obtenue des smarts objects) afin de planifier leur actions en tenant compte de leurs conséquences.

Les informations sémantiques ajoutées aux smarts objects sont décrites d'une manière formelle. Elles sont représentées par des règles en utilisant la logique du premier ordre. Les interactions sont décrites par les conditions nécessaires pour leur application et leurs effets sur l'état de l'agent et/ou de l'objet, si l'action est effectuée.

Exemple :

La formule suivante, décrit une opération *«Preparepush»*, lorsque l'agent X se prépare à déplacer un objet Y, si les conditions indiquées sont remplies dans un certain état du monde.

$$\forall X \forall Y (\exists P place(P) \wedge at(X,P) \wedge at(Y,P) \wedge agent(X) \wedge pushing(X,Y)) \Rightarrow preparepush(X,Y)$$

2.2.3.2 Star Fish

Dans [Badawi and Donikian, 2004], les auteurs proposent des objets synoptiques **STAR FISH** en suivent la même philosophie que celle des *Smarts Objects*. L'idée est de définir un ensemble minimal d'actions primitives de base qui sont utilisées pour construire des actions complexes. Ces actions sont associées à une interface interactive qui est la partie géométrique de l'objet concernée par l'action. Les agents n'auront plus qu'à interagir avec ces interfaces pour récupérer l'information relative à l'objet qu'ils veulent utiliser et adapter leur comportement en conséquence.

2.2.3.3 Conclusion

La combinaison des informations de haut-niveau ainsi que les données d'animation dans les smarts objects, permet de faire une planification plus efficace. Cette combinaison a l'avantage de simplifier le processus de conception. Ainsi, les opérateurs de planification doivent être conçus en même temps que les scripts d'interaction, ce qui contribue à assurer la cohérence entre la sémantique formelle de l'action et sa mise en œuvre réelle. Les *Smarts object* sont utilisés pour l'animation ils sont très intéressants pour des applications comme les jeux vidéos où ils ont été utilisés dans le jeu The Sims^{TM} [Forbus and Wright, 2001].

L'idée derrière ces approches est de réduire la complexité des tâches d'animation de l'agent par un transfert d'informations orienté vers l'environnement virtuel (les entités qui le composent). Par conséquent, l'interaction agent-objet est gérée par l'environnement, puisque toutes les informations relatives à l'information sont stockées dans les objets eux-mêmes.

Dans le cas de notre projet, les modules d'IA de haut niveau ont besoin d'informations plus riches sémantiquement qui vont au delà de l'information sur l'animation adaptée pour l'utilisation d'un objet. Ces informations vont leur permettre d'exécuter une action du plan en conformité avec sa description sémantique tandis qu'ils n'auront aucune information qui va leur permettre de comparer et de choisir le meilleur objet de l'environnement.

Nous allons donc nous intéresser aux modèles de représentation qui permettent de modéliser des informations sémantiques plus complexes sur l'environnement. Ce qui nous amène aux approches de représentation utilisant des ontologies, qui feront l'objet de la section suivante.

2.2.4 Les solutions avec des ontologies

Le formalisme qui s'est imposé pour associer du sens aux composants est celui des ontologies [Gruber, 1993], [Guarino and Giaretta, 1995]. En effet, elles permettent d'enlever les ambiguïtés sur la définition des concepts et de leur associer de la sémantique tout en utilisant un langage formel.

Nous présentons dans cette section les différents types d'utilisation du formalisme ontologique pour exprimer et représenter la sémantique dans les EVs.

2.2.4.1 Utilisation des ontologies dans la phase de conception

Conceptualisation des scènes de l'EV

Troyer et al. [De Troyer et al., 2003] proposent une approche qui permet de générer un monde virtuel à partir d'une ontologie de domaine. Le principe est de faire la description du monde virtuel avec un langage de haut niveau sous la forme d'une ontologie.

L'idée sous-entendue par cette approche est que le domaine des objets et leurs propriétés nécessaires dans le monde virtuel sont définis et décrits dans un domaine d'ontologie. A partir de cette ontologie, le monde virtuel peut être généré. Cependant, l'ontologie ne contient pas toutes les informations nécessaires pour générer le code pour le monde virtuel, elle décrit les objets, leurs caractéristiques et leurs relations qui existent dans le domaine considéré. De cette façon, un expert du domaine peut préciser le monde virtuel à un niveau plus élevé et la plupart du code pour le monde virtuel peut être généré [Bille et al., 2004].

Pour faire la transformation des objets de l'ontologie du domaine en objets de développement du monde virtuel, ils ont mis les différents objets qui peuvent être décrits avec le langage *DAML+OIL* et ceux avec *VORTEX* (leur outil de développement 3D) dans deux ontologies différentes. La première ontologie, contient les types d'objets décrits avec le langage *DAML+OIL* (*DAML Ontology*).

La deuxième ontologie contient les objets qui peuvent être décrits par *VORTEX* (*Vortex Ontology*). Celle-ci donnera toute la description des types d'objets disponibles dans le moteur *Vortex*. Elle contient des informations comme les types d'objets primitifs qui peuvent être affichés en utilisant le moteur 3D et les différentes contraintes et les connexions qui sont possibles.

Et enfin, une troisième ontologie est alors définie pour faire le lien entre les deux ontologies précédentes (*Mapping Ontology*) (voir la figure 2.2). Elle décrit la correspondance entre les deux formalismes utilisés, à savoir entre les concepts *DAML+OIL* et les concepts *VORTEX*. Cette ontologie ne contient qu'une seule classe décrivant les propriétés de cette correspondance.

Pour simplifier la phase de spécification, une ontologie est utilisée pour représenter le schéma de mapping par défaut qui peut être utilisé pour tous les types d'objets (*DefaultMapping Ontology*). Cette ontologie peut être considérée comme une instanciation de l'ontologie *Mapping Ontology*.

L'ontologie *DefaultMapping Ontology* contient une représentation par défaut des objets. Elle peut être utilisée pour le prototypage rapide du monde virtuel souhaité. Le développeur n'a besoin que de spécifier des représentations pour les objets les plus importants, pour les autres objets une représentation par défaut sera utilisée.

Plus tard, le projet VR-Wise [Pellens et al., 2005] est venu apporter une extension à ce projet, ce dernier permet à un non spécialiste de la réalité virtuelle, de concevoir un environnement virtuel en utilisant un ensemble d'ontologies *DAML+OIL*. Par la suite, les concepts du monde seront traduits par la représentation graphique *VRML* [2] qui est utilisée pour produire la scène.

2. Virtual Reality Modeling Language ou Virtual Reality Markup Language est un langage de description d'univers virtuels en 3 dimensions. (http ://www.w3.org/MarkUp/VRML/)

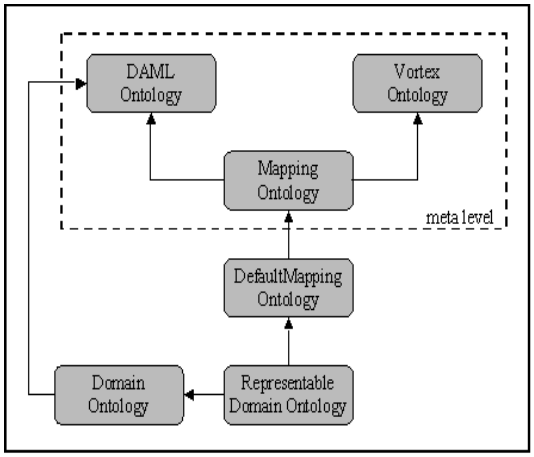


Figure 2.2 – Lien entre les différentes ontologies utilisées dans les travaux de [De Troyer et al., 2003]

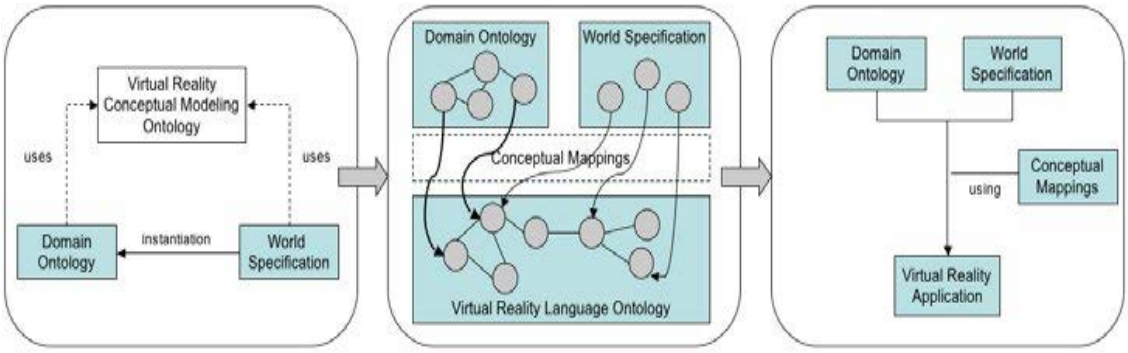


Figure 2.3 – Processus conceptuel VR-Wise proposé dans [Pellens et al., 2005] (de gauche à droite) : étape de spécification, étape de mapping et étape de génération

L'idée principale de cette approche est d'introduire une phase de conception dans le processus de design de l'EV. Ainsi, le processus de développement en VR-Wise se compose de trois étapes successives, à savoir l'étape de spécification, l'étape de mapping, et l'étape de génération (cf. figure 2.3) [Kleinermann et al., 2005].

L'étape de spécification permet de spécifier l'EV à un niveau conceptuel en utilisant les connaissances du domaine sans tenir compte de tous les détails d'implémentation. La phase de spécification se fait à deux niveaux : au niveau conceptuel et au niveau de l'instance. Au niveau conceptuel, les concepts de domaine sont mis en correspondance avec les concepts de l'ontologie de domaine. Les propriétés, les relations entre les concepts et le comportement des concepts peuvent également être spécifiées. Au niveau de l'instance, les instances représentent les objets concrets qui peuplent l'EV, tel que c'est spécifié par l'ontologie de domaine.

L'étape de mapping permet de relier les concepts du domaine à leur représentation géométrique, le but étant de préciser comment sont représentés les concepts du domaine dans le monde virtuel.

Et enfin, l'étape de génération va produire le code source de l'EV défini dans l'étape de spécification, en utilisant les correspondances définies dans l'étape de mapping.

Discussion

Toutes ces approches utilisent le formalisme des ontologies pour fournir une représentation conceptuelle du contenu de la scène. L'idée principale est de faire une correspondance directe entre le contenu graphique de la scène et leur sémantique dans les ontologies. Ces travaux proposent des alternatives intéressantes pour la création d'environnements virtuels 3D, en réduisant l'effort fourni par les développeurs et en faisant participer les experts du domaine dans la spécification de l'EV d'une manière plus intuitive, en utilisant leur expertises et leur propre terminologie. Cela permet de faire un prototypage rapide et peu coûteux en terme de temps de développement (en excluant le temps de développement des ontologies). Ce qui peut être intéressant pour des travaux de simulation et de test. Par exemple, pour des algorithmes de navigation dans un monde virtuel 3D simple.

Cependant, ces solutions manquent de généricité qui aurait permis de réutiliser d'une manière flexible les entités virtuelle d'un environnement qui ont déjà été développées pour un autre environnement et de les adapter à d'autres types d'application avec des contextes différents.

2.2.4.2 Représentation sémantique générique de l'EV

Nous présentons dans cette section les différentes approches qui proposent une représentation de l'environnement virtuel avec un niveau sémantique générique indépendant du domaine d'application permettant la réutilisation des entités virtuelles dans d'autres EV.

Réutilisabilité du modèle sémantique de l'EV

La plate-forme SeVEn [Otto, 2005] vise à développer un logiciel indépendant du système afin de créer des objets virtuels qui peuvent être réutilisés dans des environnements virtuels différents. L'auteur s'attaque au problème de réutilisabilité des environnements virtuels déjà développés. En effet, il est parti du constat qu'il n'est généralement pas possible de ré-utiliser un système d'environnement virtuel dans un autre EV, même s'ils appartiennent au même domaine d'application. Par exemple, l'agent guide d'un musée développé pour une application musée ne pourra pas être utilisé dans le système d'un autre musée car ils n'ont pas le même format du modèle ni le même système. L'objectif principal est donc de développer un modèle sémantique de l'EV qui contient des concepts communs et abstraites afin qu'ils puissent êtres réutilisés dans des EVs du même domaine d'application.

Pour ce faire, la plate-forme qui prend en charge la définition de l'EV repose sur le modèle de graphe RDF [3] [Lassila and Swick, 1999] et classe les objets en utilisant un champ de type explicite. Selon le domaine, des informations pertinentes sur les tâches d'un objet peuvent être extraites. L'utilisation des graphes RDF pour la représentation des scènes de l'environnement, permet d'avoir des accès uniformes aux entités hétérogènes de l'environnement. Cette représentation comprise par la machine permet aux agents d'utiliser différents graphes RDF d'une manière transparente.

Flexibilité de la représentation de l'environnement

Dans [Gutierrez et al., 2005], les auteurs s'attaquent au problème de **flexibilité** des entités virtuelles de l'environnement lorsqu'il s'agit de les réutiliser dans des applications

3. RDF : Resource Description Framework (http ://www.w3.org/RDF/)

et des contextes différents de ceux pour lequel ils ont été conçus au départ. Il s'agit principalement de la nécessité pour les entités virtuelles qui composent l'environnement de s'adapter à de nouvelles applications ou de nouveaux contextes du point de vue géométrique et d'interface.

Pour ce faire, les auteurs proposent un modèle sémantique générique *XML-based ontology* pour la représentation de l'environnement virtuel. Le principe est de faire un modèle sémantique qui tourne autour du concept qu'ils appellent *«item digital»*, qui correspond à une entité de l'environnement virtuel comme par exemple, un acteur ou un objet de l'environnement (voir figure 2.4).

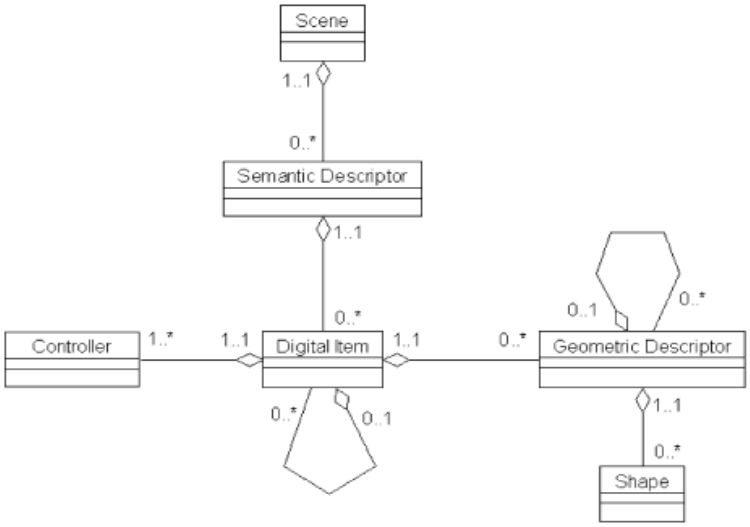


Figure 2.4 – Modèle sémantique XML-based de l'environnement virtuel autour de l'*Item Digital*, proposé dans [Gutierrez et al., 2005]

L'organisation des items digitaux est récursive, c'est à dire, un item digital peut contenir d'autres items digitaux. Un item digital a différentes représentations géométriques définies dans la classe *Geometric Descriptor* qui spécifie le type de forme associé à l'item digital. La classe *Controller* définit la façon dont l'item digital se comporte, par exemple, par des animations autonomes ou en fonction des commandes interactives envoyées par les utilisateurs du système. Enfin, un item digital est décrit par la classe *Semantic Descriptor* qui fournit des informations compréhensibles par l'homme et la machine sur un item digital particulier.

Les descripteurs sémantiques sont extraits de la norme MPEG-7 [4] sous la forme de descripteurs XML [Salembier and Sikora, 2002].

En se basant sur la description sémantique de l'item digital décrite dans *Semantic Descriptor*, il est possible de choisir la bonne représentation géométrique et la bonne interface pour représenter l'item digital (visualisation) selon le dispositif de sortie (grand écran, tablette, etc). Ce modèle de représentation de la sémantique des EVs peut être utilisé pour l'animation, l'interaction et la visualisation de l'environnement virtuel. Ces principaux avantages sont la flexibilité, l'évolutivité, la réutilisation et l'interopérabilité [Gutierrez et al., 2005].

4. MPEG-7 est une norme de description dont le but est de faciliter l'indexation et la recherche de documents multimédia.

Ontologie noyau - Concept abstrait génériques

Grimaldo et son équipe [Grimaldo et al., 2006a], [Grimaldo et al., 2006b] ont proposé une approche qui permet d'une part, de définir des objets complexes et d'introduire des niveaux sémantiques de détail qui aident les agents lors de la détection de scènes complexes. D'autre part, ils proposent une ontologie contenant la taxonomie des objets qui est utilisée pour définir les activités générales et réutilisables de la simulation.

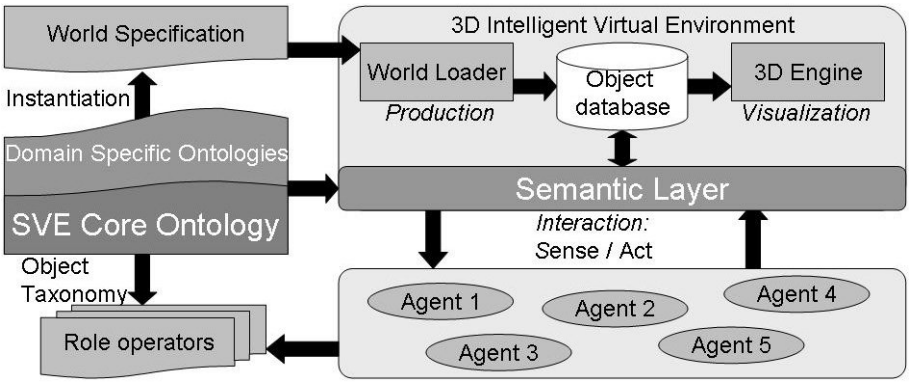


FIGURE 2.5 – Architecture de l'EV sémantique basée sur les ontologies proposée dans [Grimaldo et al., 2006a]

Grimaldo et al. [Grimaldo et al., 2008] présentent une approche des EVS qui utilise les ontologies pour animer des environnements virtuels avec des agents intelligents. L'architecture de ce framework multi-agent est constituée d'une couche sémantique qui joue le rôle d'interface entre l'agent et son environnement (le gestionnaire d'actions). Elle utilise les ontologies pour bien gérer les interactions agent-objet pendant la simulation en levant les ambigüités lorsque les agents détectent des objets complexes. En outre, cette couche est chargée d'exécuter les actions demandées par les agents et de maintenir la cohérence sémantique de leurs actions. Elle adapte les conséquences des actions des agents sur l'environnement virtuel (par exemple diminuer la distance qui reste à parcourir pour un agent lorsqu'il se déplace d'un endroit à un autre au fur et à mesure qu'il marche).

Les ontologies définissent la base de connaissances du monde. Ici nous distinguons deux niveaux de représentation, *the SVE Core ontology* qui est une ontologie de base, unique et adaptée à tous les environnements virtuels, et une ontologies spécifiques (propres à un domaine donné) afin de modéliser les connaissances propres à un domaine d'application (cf. figure 2.5).

L'ontologie noyau (*Core ontology*) définit des *concepts abstraits*, ainsi que leurs propriétés et les relations entre eux. Les concepts abstraits sont génériques et indépendants du domaine d'application. Ils peuvent représenter des agents ou bien des objets. Les objets sont classés suivant deux critères (objets mobiles / non mobiles, ou objets graphiques / non-graphiques). Chaque concept abstrait est associé à une liste de propriétés. Ces propriétés peuvent être quantitatives (l'emplacement, la hauteur) ou qualitatives (vide, plein). Les relations entre les concepts abstraits sont organisés dans un ordre hiérarchique. Autrement dit, un concept abstrait peut contenir un autre concept abstrait (cf. figure 2.6).

L'ontologie du domaine (*Domain Specific Ontology*) définit des concepts concrets d'un domaine particulier. Les concepts spécifiques au domaine sont étendues à partir des concepts abstraits prédéfinis dans l'ontologie noyau (Core ontology). Un concept de domaine spécifique hérite des propriétés et des relations de son concept abstrait. Ainsi, l'ontologie *Domain Specific Ontology* représente des connaissances sur un domaine donné.

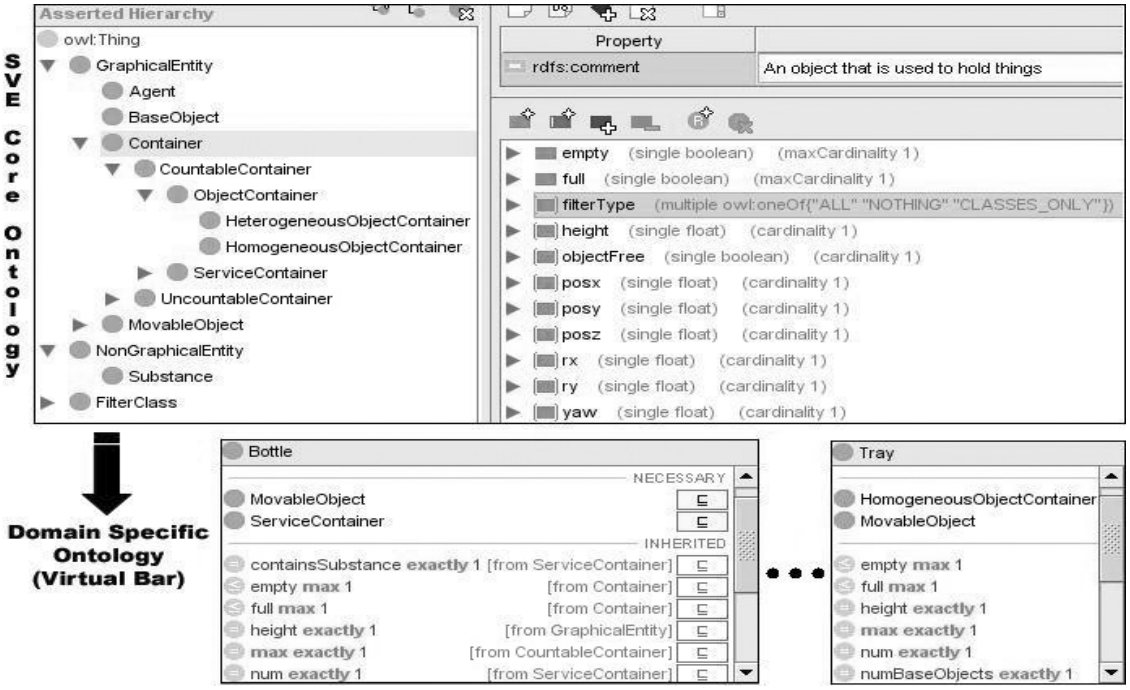


Figure 2.6 – Exemple de l'ontologie de domaine utilisée dans [Grimaldo et al., 2008]

Dans l'approche présentée dans cet article, les auteurs ont utilisé la taxonomie des objets pour définir les opérateurs généraux et réutilisables. Cette méthode permet de définir des opérateurs communs pour une famille d'objets. Cela veut dire que tous les objets instanciés d'une même classe de l'ontologie auront les même opérateurs. En outre, la taxonomie des concepts est facilement déductible de la structure hiérarchique de l'ontologie. Les agents seront en mesure de gérer n'importe quel objet du monde à condition qu'il soit une instance d'une classe définie dans l'ontologie. Ce qui évite la définition d'un opérateur pour chaque objet. Par exemple, si un serveur sait comment servir un verre d'eau, il saura très bien faire la même chose avec un verre de vin. Car l'opérateur utilisé pour les deux actions est le même. L'eau et le vin sont le contenu de bouteilles et ils seront tous les deux versés dans un contenant qui est le verre.

Ontologie de base indépendante du domaine d'application

Ibañez et Delgado-Mata [Ibáñez and Delgado-Mata, 2006] proposent un niveau sémantique de l'environnement indépendant du domaine d'application avec le but de constituer un niveau de représentation sémantique commun à divers EVs et pour tous les types d'application. Il ont réduit le nombre de composants hétérogènes dont la majorité d'entre eux peuvent être automatiquement annotés. Les auteurs considèrent qu'il y a un écart entre la représentation de bas niveau de l'EV (qui permet de visualiser la géométrie des composantes de l'environnement) et le type d'information symbolique que les agents intelligents emploient pour faire du raisonnement sur leur environnement. Ainsi, ils proposent d'intégrer un modèle de représentation de haut niveau (l'information sémantique qui permet de décrire les objets de l'environnement) à un niveau plus abstrait afin de rendre les interactions entre les agents et leur environnement plus riches en leur permettant de raisonner sur lui. Pour éviter d'avoir à définir chaque propriété de chaque objet dans le monde virtuel, les auteurs proposent un système qui définit les données des objets. Ils ont

introduit une architecture qui offre un niveau général commun sur lequel sera construite la représentation sémantique des objets de l'environnement.

Les données situées au niveau commun (niveau abstrait contenant des données de bas niveau, tels que largeur, la hauteur, l'emplacement et l'orientation automatique, qui peuvent êtres enrichies par des données plus spécifiques en fonction de l'application) regroupe des informations qui sont utiles dans toutes les applications, tels que la position et les dimensions d'un objet. L'avantage de cette approche réside dans le fait que certaines des propriétés des objets qui sont indépendantes du domaine d'application sont calculées automatiquement (par exemple, les dimensions d'un objet sur la base du modèle d'objet). Ceci diminue le temps des annotations manuelles.

Leur modèle repose sur dix concepts : le type d'objet, le réseau de navigation, l'identifiant de l'objet, l'emplacement, l'orientation, la largeur, la hauteur, la profondeur, les relations spatiales, et les plus courts chemins. Le principal avantage de ce modèle est que seuls les deux premiers éléments doivent être manuellement annotés. Les huit autres éléments peuvent être calculés. Le type de l'objet se rapporte au type d'un objet en fonction d'une ontologie particulière. Chaque objet est associé à un point tridimensionnel, appelé nœud accessible. Un ensemble de nœuds accessibles compose un réseau de navigation dans l'EV. L'avantage de cette solution est de réduire la quantité des annotations manuelles ou l'ajustement des propriétés de l'objet.

Conclusion

Le but d'utiliser les ontologies du domaine pour décrire la sémantique des environnements virtuels est de trouver une base indépendante du logiciel de description de l'environnement afin de créer des objets virtuels qui peuvent être réutilisés sur plusieurs environnements virtuels différents. La réutilisation des objets dans des environnements virtuels différents est possible en ajoutant des informations pertinentes sur les tâches d'un objet selon le domaine de l'environnement virtuel à créer.

Les solutions d'annotation automatique ou semi-automatiques permettent de réduire la quantité des annotations manuelles ou l'ajustement des propriétés de l'objet. Cela permet donc d'ajouter des données sémantiques sans influencer le temps ni l'effort de conception du monde virtuel. Mais ces données sémantiques restent insuffisante pour permettre aux agents de faire du raisonnement plus complexe sur leur EV. En effet, même si on arrive à déduire des informations sémantiques (par exemple, la définition d'un matériaux nous permet de déduire automatiquement des propriétés physiques (poids, texture, etc)) cela reste insuffisant pour le système de déduire des informations sémantiques plus riches (par exemple, la fonction ou la relation avec un autre objet) afin de faire un raisonnement plus complexe qui permet aux agents de faire de la planification, tout cela sans augmenter d'une manière considérable le temps de conception du système.

Nous aborderons dans la section suivante d'autres approches qui proposent de raisonner sur le contenu de l'environnement en donnant un accès directe à sa sémantique.

2.2.4.3 Raisonnement sur le contenu de l'environnement

Les approches présentées ci-après proposent une couche sémantique reliée à la représentation graphique des objets afin de déduire de nouvelles informations sur l'interaction avec les objets de l'environnement.

Utilisation des ontologies pour l'aide à la planification dans les EVs

Dans [Kalogerakis et al., 2006], les auteurs proposent de coupler la représentation ontologique avec la représentation du contenu graphique de l'EV. Ils proposent de mettre en correspondance les entités définies dans des fichiers VRML et X3D [5] avec une ontologie appelée *« X3DOntology »* (représentée par le Web Ontology Language (OWL)) (cf. figure 2.7)) qui fournit une représentation sémantique de la scène afin de déduire son contenu. Ainsi, chaque classe dans l'ontologie *OntologyX3D* est l'abstraction d'une ressource graphique.

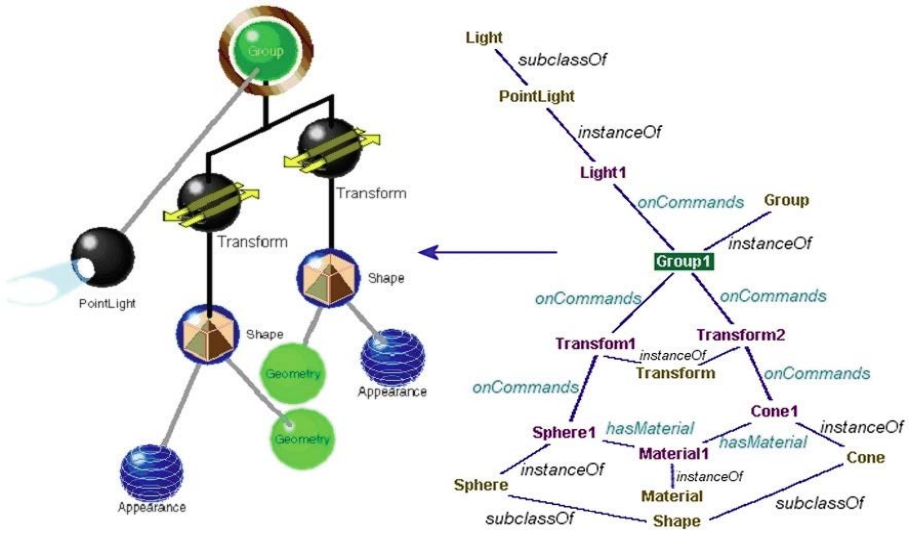


Figure 2.7 – Graphe OWL décrivant une scène dans [Kalogerakis et al., 2006]

Le couplage de la représentation graphique de la scène avec une ontologie offre aux agents la possibilité d'effectuer un raisonnement automatisé sur le contenu graphique. En utilisant un moteur d'inférence à base de règles écrite en *Semantic Web Rule Language* (SWRL) [6] sur cette représentation ontologique de l'environnement, ils peuvent répondre aux requêtes portant sur le contenu (Par exemple, pour répondre à la question " quelle est la fonctionnalité de cet objet ? ", le système effectue dans un premier temps une première correspondance entre l'objet visé dans la scène et la classe mappée dans l'ontologie, puis il va chercher la relation "hasFunction" pour cette classe.

Nous pouvons retrouver partiellement la même approche dans [Vanacken et al., 2008] dans la phase de conceptualisation, en faisant la correspondance entre les objets de l'environnement virtuel et les concepts de l'ontologie.

Procédures d'animation basées sur la sémantique de l'EV

Chu et al. [Chu and Li, 2008] cherchent à utiliser l'ontologie pour décrire la sémantique des objets dans l'environnement virtuel de telle sorte que les utilisateurs peuvent concevoir leurs propres procédures d'animation basées sur les informations sémantiques. En outre,

5. X3D est un format de fichier graphique et multimédia orienté 3D. Il a été créé par le consortium Web3D dans le but de succéder à VRML.

6. http ://www.w3.org/Submission/SWRL/

ils ont également conçu spécifiquement une ontologie pour l'échange d'informations entre les acteurs de l'environnement. L'objectif de concevoir des ontologies pour l'environnement virtuel dans ce travail est double. Tout d'abord, ils cherchent à conserver les informations qui existent dans un projet antérieur qu'ils ont réalisé IMNET [Li et al., 2005], telle que la géométrie des objets. Deuxièmement, ils veulent utiliser des informations plus sémantiques sur les objets virtuels pour faciliter le calcul des procédures de raisonnement comme la planification du chemin.

L'ontologie de l'environnement virtuel contient les informations relatives au monde virtuel (tous les objets virtuels du monde). Les informations non sémantiques sur les objets du monde virtuel (comme la géométrie des objets) sont stockées dans des fichiers, qui sont référencés dans l'ontologie. Les informations sémantiques sur les éventuelles transformations que peut subir un objet sont aussi représentées dans l'ontologie du monde virtuel. Chaque objet a des attributs additionnels comme un nom, un tag, etc... L'attribut *tag* sera défini par l'utilisateur pour indiquer des propriétés particulières d'application sur les objets virtuels. Par exemple, dans la planification de chemin, ils peuvent taguer certains objets comme des trottoirs et des passages pour piétons, de telle sorte que ces régions peuvent être traitées de façon appropriée par le planificateur de chemin, en fonction de leur signification dans le monde. Chaque objet peut également avoir une représentation 2D approximative (enveloppe extérieur) qui est un polygone qui peut être utilisé pour définir le rapprochement 2D des obstacles dans l'environnement pour le planificateur de chemin. Toutes ces informations sont représentées dans l'ontologie.

Dans ces travaux, les auteurs ont séparé la représentation du monde (la géométrie des objets) de sa sémantique tout en gardant un lien entre les deux en utilisant des références aux fichiers qui contiennent la description géométrique des objets dans l'ontologie du monde virtuel. Sauf que leur représentation du monde ne suffit pas pour faire des procédures de raisonnement autre que le path finding [Donikian, 1997].

Interface de communication entre les agents et leur EV

Chang et al. [Chang et al., 2005] proposent une architecture qui porte sur la même idée que l'approche précédente, c-à-d séparer la représentation des informations graphiques des informations sémantiques. Cette approche propose une couche cognitive intermédiaire basée sur une ontologie entre les agents et leur environnement afin de gérer les concepts sémantiques. Celle-ci permet aux agents de comprendre leur environnement à travers les relations qui existent entre les concepts, ce qui leur permet d'inférer de nouvelles relations entre les instances des concepts. La couche sémantique représente également des actions au moyen de règles de causalité dont l'effet est de transformer un objet en instance d'un autre concept.

Comme nous pouvons le voir sur la figure 2.8, les auteurs proposent un framework composé de trois couches :

- La couche **Reality** qui se trouve dans le plus bas niveau de la représentation de l'environnement. Elle contient les caractéristiques physiques de l'environnement. Elle utilise les approximations de la loi physique pour modéliser la variation de ces propriétés physiques. Ces propriétés sont reliées par un ensemble de fonctions aux concepts de l'environnement définis dans la couche Concept le plus souvent sous une forme numérique. Cependant, certains objets artificiels n'ont pas de propriétés physiques simples. Ainsi, ils sont directement reliés à un concept sans avoir à passer

par des calculs numériques.

- La couche **Concept** qui est au cœur du framework. Cette couche contient une copie du modèle conceptuel qui spécifie les relations entre les concepts physiques, ou entre les concepts physiques et les actions. Les liens entre les concepts se divisent en deux types : les relations ontologiques et les règles de causalité. Les relations ontologiques comprennent à la fois la hiérarchie des concepts et les relations de cause à effet entre des instances de concepts. La hiérarchie des concepts définit les relations de sous-concepts entre les concepts, alors que les relations causales décrivent l'effet provoqué d'une instance spécifique sur une autre instance.
- La couche **Mind** utilise la couche Concept qui contient tous ce que l'agent peut percevoir dans son environnement et comment il peut agir sur lui, tandis que, les émotions, les intentions, la planification et la capacité de prise de décision appartient à l'agent. Dans certains cas les instances de concept déclenchent des émotions chez l'agent et dans d'autres cas elles deviennent une partie du plan de l'agent.

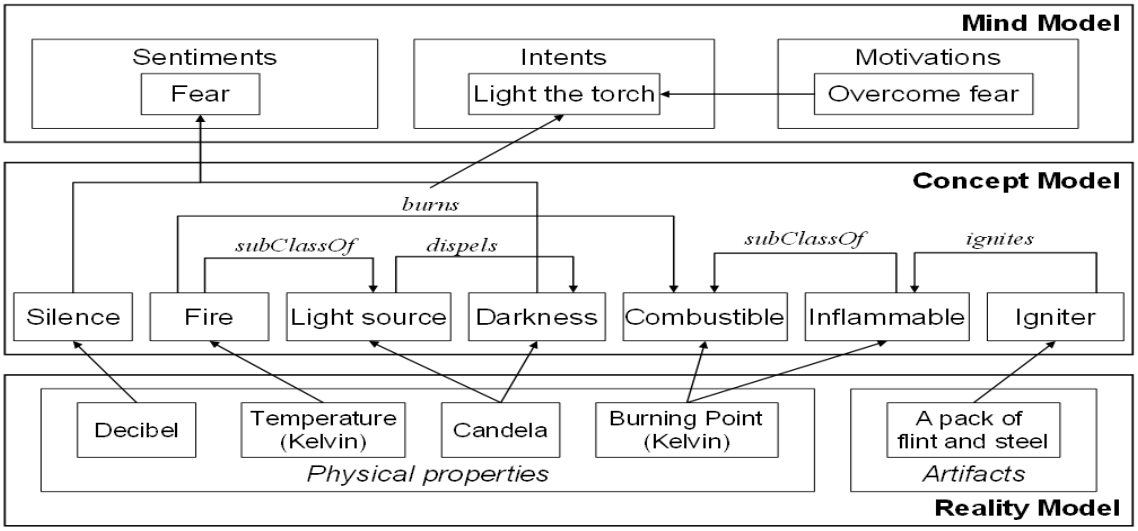


Figure 2.8 – Architecture du framework à trois couches proposé dans [Chang et al., 2005]

Les auteurs proposent une couche intermédiaire cognitive entre la cognition de l'agent et l'environnement basée sur des ontologies. Ces ontologies vont permettre aux agents d'interpréter les scènes comme des instances de concepts reliés entre eux, leur permettant par la suite d'inférer les relations entre les objets . Les actions possibles sur les entités de l'environnement sont représentées au moyen de règles de causalité dont l'effet retourne un objet dans une instance d'un autre concept. Cette approche est utilisée pour la modélisation du module cognitif d'un robot assistant d'un laboratoire de chimie.

Combinaison des techniques du web sémantique et l'IA

Kapahnke et son équipe [Kapahnke et al., 2010] ont considéré le problème de l'incapacité de compréhension de la sémantique de l'environnement virtuel par les avatars (agents). Cette incompréhension est due en raison de l'absence d'annotations sémantiques des scènes 3D. Ainsi, il n'est pas possible de raisonner dessus, ni d'exploiter la sémantique de la scène 3D pour la planification d'actions intelligentes.

Les auteurs proposent une plate-forme 3DI [7] (Projet ISReal [Tansu et al., 2007]) pour la simulation d'un monde virtuel 3D fondée sur les techniques du Web sémantique et composé de cinq composants :l'interface utilisateur, la sémantique globale, les agents intelligents, un environnement de vérification et les graphiques 3D (voir figre 2.9).

Les avatars de la simulation sont couplés avec des agents intelligents qui comprennent la sémantique de leur environnement 3D grâce aux annotations sémantiques des objets de l'environnement rajoutées, ce qui permet aux agents d'effectuer la planification de leurs actions en utilisant cette sémantique pour satisfaire leurs objectifs (requêtes). Un agent ISReal peut répondre aux requêtes sémantiques sur l'ontologie globale du monde simulé. En outre, il peut satisfaire les objectifs d'action comme "Allez à la pièce suivante." ou des objectifs déclaratifs comme "Montrez-moi comment produire Y avec la machine X ?" en utilisant son planificateur BDI et la composition de services sémantique définis dans l'ontologie.

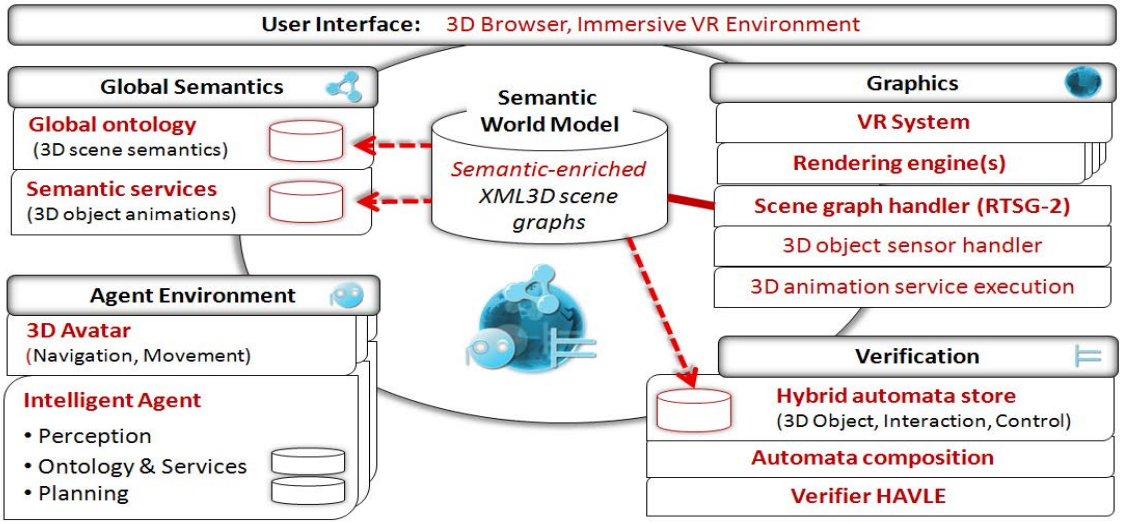


FIGURE 2.9 – Composants de la plateforme ISReal proposée dans [Kapahnke et al., 2010]

Le modèle du monde sémantique de la plateforme ISReal est l'ensemble de tous les graphes de scène XML3D [8] sémantiquement annotés pour le monde virtuel simulé. Tout objet 3D de la scène est représenté comme un nœud de la scène graphique XML3D qui décrit graphiquement le monde virtuel. La sémantique d'un objet 3D de la scène est décrite par l'annotation de son nœud XML3D du graphe de scène en utilisant le standard RDFa [9] reliée à (a) l'objet annoté dans l'ontologie de la scène définie avec OWL2 [10]. Celle-ci représente les connaissances conceptuelles et assertionnelles de la scène et du domaine d'application, (b) les services sémantiques (OWL-S [Martin et al., 2004]) qui décrivent la fonctionnalité opérationnelle de l'objet de la scène fondé sur son script d'animation 3D, et (c) les automates hybrides qui décrivent les propriétés de l'objet (cf. figure 2.10).

La représentation sémantique de l'environnement est composé de deux parties (cf. figure 2.11), qui sont une ontologie pour la gestion du système (Ontology Management System OMS) et un gestionnaire de services sémantique (semantic service handler SemSH). L'OMS maintient un ensemble d'ontologies décrivant les connaissances conceptuelles (TBox)

7. 3D Internet est l'ensemble des mondes de réalité virtuelle 3D sur Internet que les utilisateurs peuvent utiliser et partager avec d'autres pour diverses applications.
8. http ://xml3d.org/
9. http ://www.w3.org/TR/xhtml-rdfa-primer/
10. http ://www.w3.org/TR/owl2-overview/

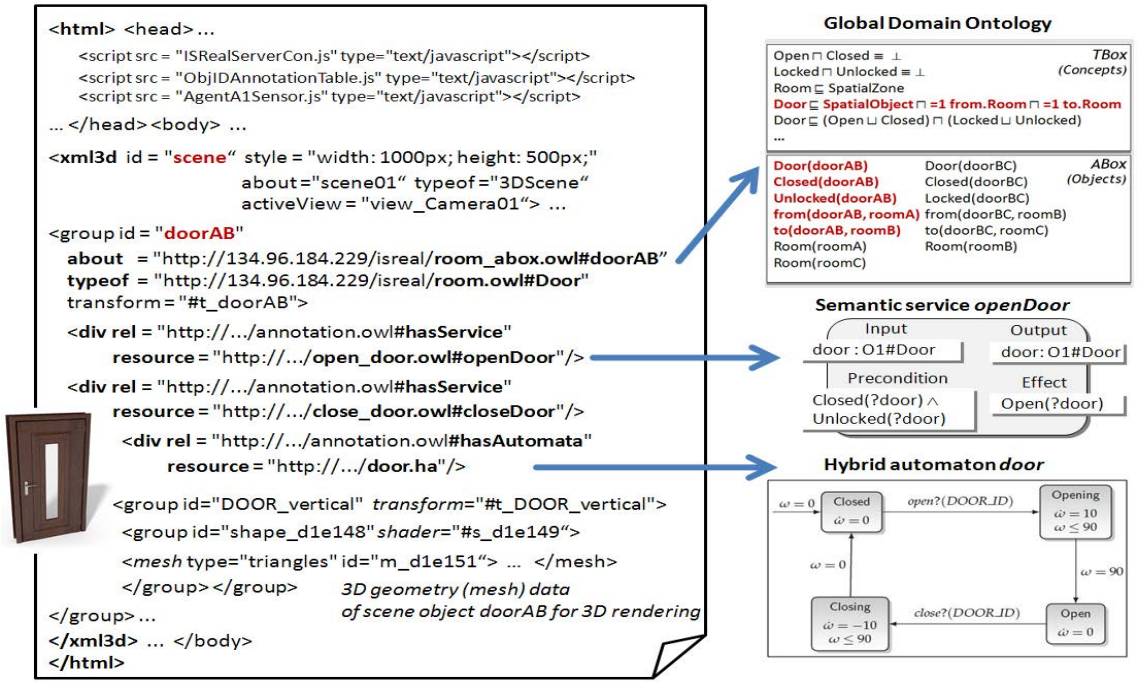


Figure 2.10 – Exemple d'annotation sémantique d'un objet graphique XML3D avec RDFa.

et factuelles (ABox) base de faits du monde virtuel simulé. Il gère le traitement des différents types de requêtes sémantiques émises par l'utilisateur ou les agents. Le SemSH gère le référentiel de service sémantique contenant tous les services avec OWL-S qui sont liés à l'ontologie globale de l'environnement.

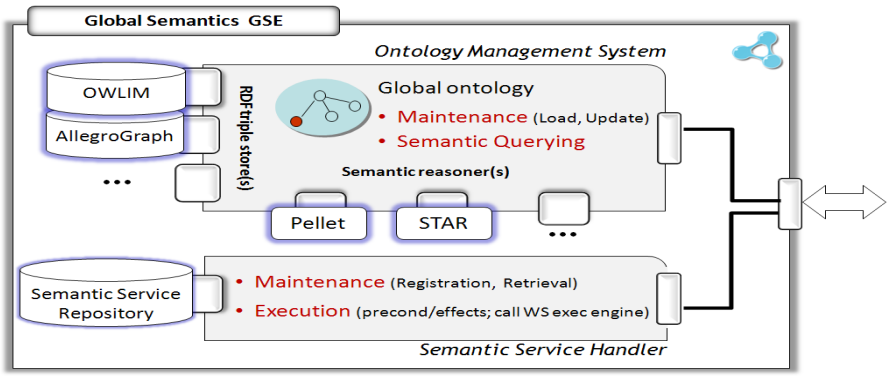


Figure 2.11 – Architecture du modèle sémantique de l'environnement virtuel de la plateforme ISReal proposée dans [Kapahnke et al., 2010]

La planification des actions des agents se fait avec OWLS-Xplan [Klusch et al., 2005]. Pour cela, OWLS-xPlan convertit d'abord l'ontologie de domaine et les descriptions de services en OWL et OWL-S, respectivement en problèmes PDDL2,1 [Fox and Long, 2003] et en descriptions du domaine. La description du domaine contient la définition de tous les types, prédicats et actions, alors que la description du problème inclut tous les objets, l'état initial et l'état but. Les deux descriptions sont ensuite utilisées par l'IA du planifi-

cateur pour créer un plan (cela consiste à faire la composition de services afin d'arriver à l'état but désiré).

Cette approche utilise les techniques du web sémantique pour représenter la sémantique de l'environnement. Elle utilise le concept des services web représentés dans une ontologie OWL-S [Martin et al., 2004] pour décrire la fonction des objets de l'environnement et leurs propriétés. Ainsi, la planification des actions des agents se fait par la combinaison de services en se basant sur un état initial et des pré-conditions pour arriver à l'état but. Cependant, la sélection de services pendant la planification ne repose pas sur une qualité de service QoS, et donc il n'y a pas de comparaison entre les services équivalents. Ce qui ne conduit pas forcément à la solution la plus optimale pour atteindre l'état but. Ce type de panification peut être intéressant pour un nombre réduit de services hétérogènes. Dans le cas de notre environnement de simulation, il comporte des objets hétérogènes qui peuvent proposer le même service avec des qualités différentes. Pour cela, nous avons considéré qu'il était important de trier les objets selon la QoS qu'ils proposent afin que le module de décision de l'agent puisse choisir la meilleure solution pour l'action échéante.

Conclusion

Les approches présentées ici proposent une représentation sémantique intermédiaire qui joue le rôle d'interface entre les agents et leur environnent virtuel. Cette représentation sémantique est construite à partir d'un ensemble d'ontologies. Ce type d'architecture permet de séparer la représentation graphique de la représentation sémantique de l'environnement virtuel, ce qui est tout à fait adapté aux environnements virtuels sémantiques. Ainsi, les agents peuvent interagir directement avec la sémantique de leur environnement afin d'appliquer un raisonnement direct dessus. Ce type d'architecture est réutilisable dans différents types d'application. En effet, il suffit juste d'utiliser les ontologies propres au domaine d'application de l'EV et de définir les raisonnements appropriés au domaine qui seront employés par les agents.

2.2.5 Synthèse

Afin de faire la synthèse des différentes approches que nous avons présentées dans cet état de l'art, nous avons défini les différentes caractéristiques qui nous permettent de les classer.

Caractéristiques des différentes approches de représentation sémantique des EVs

Les différentes approches citées précédemment dans notre état de l'art concernant la modélisation de la sémantique des EVs, qui peuvent êtres caractérisées selon quatre critères que nous jugeons êtres importants pour notre approche. Ces critères sont définis comme suit :

- **C1 (Modèle Orienté-Contenu) :** qui représente les connaissances du contenu de l'EV,
- **C2 (Généricité) :** qui définit si le modèle sémantique de l'environnement est générique ou dépend-t-il du domaine d'application,
- **C3 (Découplage de la représentation graphique) :** qui représente les connaissances du contenu de l'EV indépendamment de sa représentation graphique,

– **C4 (Méthodologie conceptuelle) :** qui propose une méthode de modélisation de l'EV, ainsi le modèle sémantique de l'EV résulte de la phase de modélisation conceptuelle.

Selon les critères que nous avons définis, nous pouvons résumer les différentes approches liées à la modélisation sémantique des EVs dans le tableau 2.1 :

Approches / Critères	C1	C2	C3	C4
Environnement Virtuel Informé [Farenc et al., 1999a]	✓			
VUEMS [Donikian, 1997]	✓			
SeVEn [Otto, 2005]	✓	✓		
OntologyX3D [Kalogerakis et al., 2006]	✓	✓		
Noyau et Ontologie spécifique [Grimaldo et al., 2006a]	✓	✓		
Ontologie basée sur du XML [Gutierrez et al., 2005]	✓	✓		
Niveau Sémantique de base [Ibáñez and Delgado-Mata, 2006]	✓	✓	✓	
Information sémantique pour l'animation [Chu and Li, 2008]	✓	✓	✓	
ISReal [Kapahnke et al., 2010]	✓	✓	✓	
VR-WISE [Pellens et al., 2005]	✓	✓	✓	✓
NiMMiT [Vanacken et al., 2007]	✓	✓	✓	✓
MASCARET [Chevaillier et al., 2009]	✓	✓	✓	✓

Table 2.1 – Récapitulatif des différentes approches de modélisation de la sémantique des EVs dans quatre critères : représentation de la sémantique du contenu de l'EV, indépendance du domaine d'application, découplage de la représentation graphique et résultant d'une phase de modélisation conceptuelle.

Les approches de modélisations de la sémantique de l'environnement qui dépendent du domaine d'application [Farenc et al., 1999a], [Donikian, 1997] sont limitées pour qu'elles soient uniquement utilisées dans le domaine pour lequel elles ont été définies. De ce fait le critère de généricité n'est pas vérifié.

L'utilisation d'une couche de représentation sémantique de l'environnement permet de dissocier la représentation graphique de l'environnement de sa sémantique. Les agents peuvent alors interagir directement avec la couche sémantique pour l'inclure dans leur processus de raisonnement sur l'environnement. Ce qui la rend réutilisable pour différents types d'applications.

La plupart des approches de représentation du contenu de l'EV [Otto, 2005], [Grimaldo et al., 2006a] utilisent les ontologies pour coupler le contenu de l'EV avec sa représentation sémantique contenue dans une ou plusieurs ontologies. Cependant, comme les concepts de l'ontologie sont reliés directement à des primitives géométriques de bas niveau, ce couplage ne peut être fait que d'une manière ad-hoc et de façon manuelle.

En revanche, ce couplage manuelle de la sémantique des entités de l'environnement est évité dans les approches qui incluent la sémantique de l'EV dans la phase de modélisation conceptuelle [Pellens et al., 2005, Vanacken et al., 2007] en utilisant un modèle conceptuel explicite. Toutefois, la limitation majeure de ces approches résident dans le fait que seuls les aspects statiques de l'EV sont représentés dans le modèle conceptuel. De ce fait, nous ne pouvons pas observer lors de l'exécution les changements des attributs des objets et

leurs relations.

L'objectif principal de la représentation des connaissances sémantiques de notre EV est de le rendre compréhensible par les agents de la simulation pour qu'ils puissent choisir les meilleurs objets de l'EV pour réaliser leurs actions. Ce qui rendrait leurs comportements plus crédibles.

Nous voulons donc modéliser la sémantique des objets de l'environnement relative à leurs fonctions (services proposés par les objets) et les critères de qualité qui vont nous permettre de choisir les meilleurs objets qui proposent les services les plus adaptés pour répondre aux actions des agents.

Pour ce faire, la modélisation sémantique de notre EV s'inspire des approches de modélisation orientée-contenu. En outre, elle doit être générique, basée sur un méta-modèle représentant les fonctionnalités opérationnelles des entités de l'environnement, comme l'ontologie de service dans ISReal [Kapahnke et al., 2010] qui utilise l'instanciation de l'ontologie OWL-S. Et enfin, notre modèle de représentation doit être indépendant de la plateforme d'application de l'EV, générée pendant la phase de modélisation conceptuelle du processus de développement [Pellens et al., 2005], [Chevaillier et al., 2009].

2.2.6 Conclusion

L'information sémantique des EVs joue un rôle important pour améliorer la capacité des agents à interagir avec leur environnement [Cavazza, 1998]. Nous avons constaté à travers les différents travaux cités dans la littérature qu'il n'y avait pas un standard de modélisation de cette sémantique.

L'une des premières méthodes de représentation de la sémantique des EVs consistait à l'annotation des méta-données associées aux éléments contenus dans l'EV. C'est une étape utile et nécessaire pour établir une couche d'informations sémantiques sur l'environnement. Cependant, l'annotation sémantique de l'environnement reste une tâche fastidieuse pour les concepteurs du système, il est donc plus intéressant d'utiliser des techniques d'annotation semi-automatiques ou automatiques lorsque c'est possible car ça demande moins d'effort.

Parallèlement, des équipes de recherches se sont penchées sur la question de la définition de la sémantique de l'environnement virtuel. Ils ont considéré qu'elle doit suivre une stratégie et une méthodologie qui facilite et formalise le processus de conception de l'environnement virtuel. Comme nous avons pu le voir dans *VR-WISE* [Pellens et al., 2005] qui définit les premières étapes pour la définition complète d'une méthodologie afin de permettre à l'utilisateur final de définir l'EV à un niveau conceptuel en utilisant la terminologie du domaine.

Beaucoup de travaux ont proposé des approches basées sur des ontologies qui sont appropriées pour la représentation de la sémantique des EVs. L'utilisation des ontologies et les bases de connaissances du domaine constituent un niveau sémantique dans l'architecture des EVs. Ce type d'approche constitue une couche de représentation sémantique de l'environnement qui permet de dissocier la représentation graphique de l'environnement de sa sémantique. Ainsi, les agents peuvent interagir directement avec la couche sémantique pour l'inclure dans leur processus de raisonnement sur l'environnement. Ce qui la rend réutilisable pour différents types d'applications. Cependant, les raisonnements sur l'environnement que proposent ces approches ne permettent pas aux agents de *comparer* entre les différentes solutions qui s'offrent à eux pour réaliser leurs actions. Ce qui représente un manque majeur pour la crédibilité des comportements des agents. En effet, les agents

peuvent être confrontés à choisir entre deux entités de l'environnement qui leur permettent de réaliser la même action. En faisant l'analogie avec le domaine du sémantique web où le problème de classement des services web a été largement exploré. Les services web représentent les fonctions rendues par les entités de l'environnement (objets). Différentes approches de représentation pour la déduction d'une *QoS* permettant de trier les services entre eux ont été proposées. Nous allons donc nous intéresser à ces différentes approches dans ce qui suit.

2.3 Sélection des services de l'environnement

La modélisation sémantique a été étudiée et appliquée avec succès dans de nombreux domaines scientifiques, notamment dans le web sémantique dont s'inspire notre modèle de représentation. Dans la littérature, diverses approches ont été proposées pour permettre la modélisation de **Services Web Sémantiques**. Nous nous intéressons plus particulièrement à l'ontologie **OWL-S**(Ontology Web Language for Services) qui a souvent été utilisée pour définir des modèles de représentation de la qualité des services.

Dans la suite de cette section nous allons définir l'ontologie OWL-S. Nous présenterons par la suite quelques travaux qui ont été réalisés dans le domaine du sémantique web pour le classement des services en proposant différentes représentations sémantiques pour la déduction de la *QoS*.

2.3.1 Représentation des Services Web Sémantiques avec OWL-S

OWL-S [Martin et al., 2004] est basé sur le langage OWL. Cette ontologie a pour objectif de décrire de façon non ambigüe les Services Web de telle sorte qu'un agent logiciel puisse exploiter automatiquement ses informations. OWL-S permet la découverte automatique, la composition et l'interopérabilité de services Web et la surveillance automatique de leur exécution.

OWL-S décrit un service à l'aide de trois classes principales :

- *ServiceProfile* : définit le service Web,
- *ServiceModel* : elle décrit le fonctionnement du service Web. Ceci est fait en exprimant les transformations faites par le service Web sur les données (input et output), et la transformation d'état (pré-conditions et effets). Les services Web sont modélisés avec OWL-S en tant que processus .
- *ServiceGrounding* : définit les détails techniques permettant d'accéder au service Web.

Nous nous intéressant à la description des services web utilisée dans OWL-S (*ServiceProfile*) (voir figure 2.12).

ServiceProfile

Cette classe permet de décrire le service Web. La classe *ServiceProfile* spécifie :

- le nom du service, les contacts et la description textuelle du service : le nom du service est utilisé comme identifiant du service, tandis que les informations contacts et la description textuelle sont destinées aux utilisateurs humains,
- la description fonctionnelle du service : elle spécifie les paramètres, les pré-conditions et les effets du service, les conditions d'entrées (*Inputs*) attendues et les résultats produits en sortie (*Outputs*).

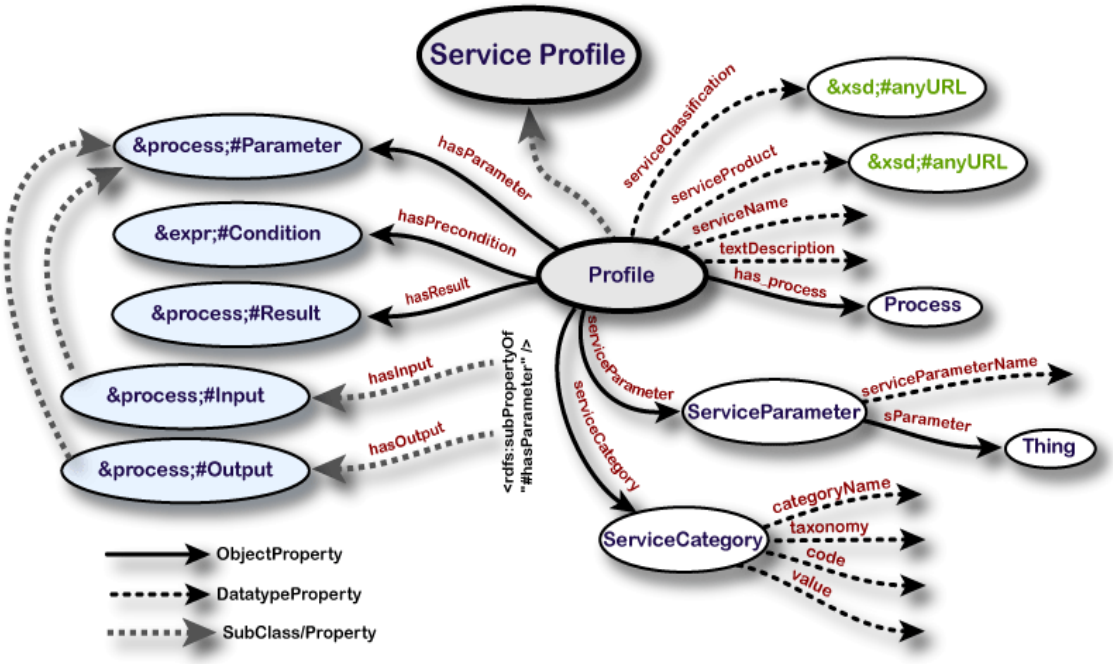


FIGURE 2.12 – Classe ServiceProfile de l'ontologie OWL-S

- les attributs du profil du service : ils comprennent les garanties de qualité qui sont fournis par le service, la classification possible du service, et les paramètres supplémentaires que le service peut vouloir préciser.
- la classification taxinomique : la classe *ServiceCategory* décrit les catégories de services sur la base de certaines classifications qui peuvent être non définies dans OWL-S.
- la spécification du type de service et du produit : les deux propriétés *serviceClassification* et *serviceProduct*, sont utilisées pour spécifier le type du service fourni et les produits qui sont manipulés par le service. Les valeurs des deux propriétés sont des instances de classes indiquées dans l'ontologie de services et de produits.

La représentation des services web dans l'ontologie OWL-S est complète pour la définition et la découverte des services web, cependant elle ne contient pas assez d'informations pour déduire la qualité des services qui permet de les comparer entre eux. Nous verrons dans ce qui suit quelques approches qui ont proposé de déduire la *QoS* en étendant ou en complétant l'ontologie OWL-S.

2.3.2 Différentes approches de représentation de la *QoS*

Il existe deux types d'approche pour la représentation de la *QoS* pour les services web, les approches Basée-Composant et les approches Basée-Service. Les approches basée-composant décrivent les catégories de contraintes dans une ou plusieurs dimensions. Chaque dimension a une métrique qui peut être soit numérique pour les caractéristiques quantitatives ou ordinales pour les caractéristiques qualitatives. Cependant, ils ne fournissent pas de comparaison entre les valeurs des propriétés. Ainsi, ces approches n'ont pas d'expressivité sémantique ni de mécanismes de classement [Bart et al., 2006], [Zhang et al., 2012], [Yessad and Boufaida, 2011].

Les approches basée-service pallient ce manque d'expressivité sémantique et proposent des mécanismes de classement entre les services. La plupart de ces approches proposent une extension de l'ontologie OWL-S car la définition des services comme ils sont décrits dans OWL-S n'offre pas la possibilité de faire la comparaison entre les services en calculant leur qualité par rapport à la demande du client.

L'ontologie **DAML-QoS** [Zhou et al., 2004], propose une extension de *DAML-S* (devenu ensuite *OWL-S*) pour la *QoS*. Elle détermine les contraintes du domaine et le rang des propriétés de la *QoS* qui sont utilisés dans le mécanisme de matching de *QoS*. Elle permet aussi de définir des métriques atomiques et complexes. Par exemple, le concept *QoSPrecondition* permet de définir les conditions qui doivent être remplies par le consommateur de service web pour obtenir la qualité de service spécifiée par le fournisseur. Cependant, cette ontologie ne fournit pas de description concrète des propriétés de qualité.

L'ontologie **OWL-Q** [Plexousakis and Kritikos, 2007] est une extension de *OWL-S*, elle permet l'association des métriques aux propriétés de qualité. Ce système complet et extensible permet en effet de représenter des métriques simples, complexes (dérivées à partir d'autres métriques) et même d'ajouter de nouvelles métriques. De plus, il fournit des algorithmes complexes pour comparer ces métriques, mais aussi de distinguer les attributs de qualité dont les valeurs sont statiques de celles dont les valeurs sont dynamiques (changement au cours du temps). Cependant, il ne permet pas de définir une hiérarchie entre ces attributs ni d'autres relations que l'on peut établir et qui sont définies dans les standards de qualité. Il n'est pas possible de construire un profile de *QoS* à partir d'un ensemble d'attributs *QoS*, c.à.d, de caractéristiques de *QoS*.

L'ontologie **WSMO-QoS** [Vu, 2003] ne permet pas elle aussi de spécifier un profile *QoS* à partir d'un ensemble de caractéristiques *QoS*.

L'ontologie **QoSOnt** [Dobson et al., 2005] contient différentes ontologies interconnectées permettant de spécifier les besoins de qualité de service. Le problème principal qui est traité est celui d'avoir une même métrique associée à des unités différentes. *QoSOnt* comprend une ontologie qui permet de faire le lien avec l'ontologie *OWL-S*, c'est à dire d'associer des attributs et des métriques aux services *OWL-S*. Cependant, *QoSOnt* ne permet pas de définir plusieurs niveaux d'abstractions d'attributs de qualité ni d'identifier les attributs de qualité qui représentent le même concept même s'ils sont définis par des termes différents.

QOS-MO [Tondello and Siqueira, 2008] permet de représenter la *QoS* des services web décrits en *OWL-S*. Elle permet de décrire l'interaction entre les fournisseurs et les consommateurs de services web. Les caractéristiques de qualité sont définies dans une classe *QoSCharacteristic*. Cette classe ne propose que peu d'éléments pour définir les types de données et les unités de mesure de ces caractéristiques. De même, elle ne permet pas de définir des relations entre ces caractéristiques, ni d'établir de lien avec les standards de qualité.

L'ontologie **onQoS** [Zimeo and Giallonardo, 2007] définit des éléments permettant d'associer des concepts de *QoS* aux profils de services web dans *OWL-S*. Cette ontologie propose un puissant système de types de données pour décrire les valeurs des propriétés de

qualité. L'évaluation de ces valeurs peut ensuite être faite à partir de métriques, échelles et règles de calcul. Cette approche ne fournit aucun élément supplémentaire à *OWL-S* pour décrire les caractéristiques de qualité. L'ontologie *onQoS* contient trois niveaux d'ontologie (haut, moyen et bas). Le niveau haut de l'ontologie permet de faire le lien entre l'ontologie *OWL-S* et le sens de spécifier les mesures de la *QoS*. Le niveau moyen de l'ontologie définit le vocabulaire standard en relation avec le *QoS* comme les paramètres, les métriques et les échelles. Et enfin, le dernier niveau (bas) de l'ontologie définit les concepts, les propriétés et les contraintes d'un domaine spécifique. Il mesure aussi les différents types de métriques comme les métriques internes ou externes.

FQoS [Jeong et al., 2009] définit un ensemble d'attributs fonctionnels (catégorie, nom, nom d'opération, données d'entrée/sortie, annotation) en spécifiant leur métrique ainsi que les opérations nécessaires à leur utilisation pour le processus de sélection. Elle permet également de définir des attributs de qualité (coût d'invocation, performance, disponibilité, sécurité, etc.). Ces caractéristiques fonctionnelles et non fonctionnelles sont ensuite utilisées dans le processus d'appariement entre demande et offre de services. Cependant, ces travaux ne considèrent ni la possibilité d'enrichir la description de service avec des ontologies, ni le problème de terminologie des attributs de qualité dans le processus de sélection.

SL-Ontology [Bleul and Weise, 2005] est une ontologie construite en couches. Elle s'appuie sur 4 autres ontologies, que ça soit pour décrire les différentes caractéristiques de qualité du service, d'associer les caractéristique de qualité à une métrique, ou de relier entre les différentes unités et de définir leur règles de transformation. Cette approche utilise des taxonomies pour régler le problème de la variabilité de terminologie pour les noms de caractéristiques de qualité, Ainsi l'équivalence entre caractéristiques de qualité repose uniquement sur les noms donnés.

Les différentes solutions proposées pour la représentation de la *QoS* dans le domaine du web sémantique utilisent une extension de OWL-S. En effet, l'ontologie OWL-S a été proposée initialement pour la représentation des services web afin de faciliter leur découverte. Elle ne permet pas donc de calculer une *QoS* en se basant sur les concepts définis pour la représentation des services. Les différentes approches proposées ont donc complété cette représentation des services par différentes caractéristiques portant essentiellement sur l'aspect non fonctionnel de la qualité de service, ainsi que des métriques de mesure.

Nous nous sommes intéressés à la représentation des services web dans OWL-S pour définir les services de l'environnement dans une ontologie de services partageant le même principe de représentation des caractères fonctionnels d'un service, c'est-à-dire, son nom, le type du service, l'action qu'il permet de réaliser, le type d'objet qui le propose, etc. Cette représentation sera enrichie par la définition des caractéristiques non fonctionnelles des services de l'environnement afin de pouvoir calculer une *QoS*, ces caractéristiques comprennent les critères permettant d'évaluer les services.

Le fait qu'un service permet de réaliser une action ne veut pas dire forcément que le service sélectionné répond aux attentes d'un agent, En effet, l'agent peut avoir des préférences pour réaliser une action.

Le problème de prise en compte des préférences des utilisateurs dans le domaine du Web sémantique a fait l'objet de nombreux travaux. C'est ce que nous aborderons dans la section suivante.

2.3.3 Prise en compte des préférences dans la sélection

L'ordre établi sur l'ensemble des résultats retournés avec le calcul de la *QoS* vient trier les objets sélectionnés en fonction de leur pertinence par rapport à l'action que l'agent va réaliser. La sélection des objets de l'environnement basée sur la fonction qu'ils offrent reste insuffisante pour la pertinence des résultats par rapport aux préférences des agents (l'action exacte qu'ils souhaitent réaliser, leur profil et leur contexte). Il est donc intéressant de tenir compte des préférences des agents et de leur contexte afin de répondre au mieux à leur requête. Le premier problème posé est comment intégrer les préférences des agents dans le processus de recherche d'objets dans l'ontologie de représentation de l'environnement.

Les préférences ont leur origine dans la théorie de la décision, comme un moyen de soutenir les processus de décision complexes, multifactorielles [Fishburn, 1970]. Plusieurs techniques d'interprétation des préférences ont déjà été mises au point , comme par exemple dans [Blum et al., 2004, Sai et al., 2001], celles-ci peuvent êtres utilisées dans des applications de Web sémantique. Car en effet, les utilisateurs n'expriment pas souvent leurs préférences implicitement, il faut donc des techniques qui permettent de déduire de nouvelles informations qui vont aider dans la recherche à partir du profil de l'utilisateur ou d'autres informations de contexte.

Plus récemment, dans le domaine du Web sémantique, les chercheurs se sont penchés sur des méthodes de découverte de services web en se basant sur les préférences des utilisateurs. Le principe est de déduire les préférences des utilisateurs et de les intégrer d'une manière implicite dans la recherche et la proposition des services web. Une première ébauche de solutions ont émergées pour pouvoir proposer "la meilleure" réponses aux utilisateurs. Deux approches ont été étudiées, la première consiste à introduire des propriétés spécifiques dans les ontologies, attachées à des classes. La seconde consiste à utiliser les attributs spécifiques du modèle d'ontologie comme la note, ou une définition pour coder la préférence.

Parmi les travaux qui ont ébauchés des solutions dans ce sens dans le domaine du Web Sémantique, nous avons [Muller et al., 2004, Haase et al., 2004]. Par exemple, dans [Haase et al., 2004] ils cherchent des publications par thème et classent les résultats en fonction de leur similarité avec le thème demandé. Cependant, les préférences utilisées dans ces systèmes s'appliquent généralement à des propriétés spécifiques, et sont codés en dur, et la fonction de classement est spécifique à leur domaine d'application ce qui la rend inutilisable dans une autre application.

Cependant, ces solutions ne sont pas facilement applicables. En général, elles proposent d'intégrer de l'information structurée tel que la taxonomie, sauf que les requêtes des utilisateurs sont constituées de mots clés qui doivent êtres traduits dans la requête en information structurée ce qui n'est pas une tâche facile à faire.

Plaçons nous dans le cas d'un système avec des agents qui n'expriment pas des préférences d'une manière explicite pour réaliser leurs actions, mais qui seront dotés d'un profil (rôle) et situés dans un contexte de simulation. Les agents auront à exécuter un plan qui leur est dicté par le module de décision. Ils vont donc enchainer l'exécution de plusieurs actions consécutives afin d'atteindre un but. Chaque action sera précise et contribuera pour atteindre le but final de l'agent. Les actions des agents doivent alors être interprétées afin qu'ils puissent réaliser au mieux leurs actions et atteindre au final leurs buts.

Par exemple, un agent qui n'a pas le temps de faire une longue pause déjeuner avant de retourner au travail, cherchera donc à casser la croute rapidement. De ce fait, la requête

doit être interprétée pour déduire qu'il souhaite manger rapidement. Dans ce cas, le restaurant n'est pas adapté à son besoin puisqu'il n'a pas le temps. L'agent va donc préférer plutôt un fast-food où il pourra manger sur le pouce, ou une boulangerie qui proposent des sandwiches à emporter.

De plus, si le système doit simuler un grand nombre d'agents en temps réel, le temps de réponse doit être rapide et la réponse efficace. Par conséquent, il n'est pas possible de demander aux agents d'exprimer leurs préférences sur tous les services recherchés vu le nombre de services hétérogènes et les types d'objets qui les proposent qui sont tout aussi différents et hétérogènes (comme c'est le cas dans notre projet). Il se doit donc de tenir compte du temps de traitement dans le processus de recherche, qui sera optimisé s'il existait un langage commun et des définitions communes qui permettent de déduire quel est le service qui répond à une action précise.

Le principe est donc de définir d'une manière formelle dans une ontologie les différents services qui sont proposés par les objets de l'environnement qui correspondent à des actions précises qui nous permet d'interpréter le besoin de l'agent et donc de répondre au mieux à sa requête.

2.4 Crédibilité

La crédibilité d'une simulation mettant en scène plusieurs agents dans un environnement virtuel repose sur la crédibilité de ce dernier et de la manière dont il évolue [Loyall, 1997].

La crédibilité telle qu'elle a été définie par Burkhardt, est le degré de confiance ou de véracité des attitudes et des comportements que les utilisateurs vont accorder aux agents dans leur environnement [Burkhardt, 2007]. Cependant, un environnement virtuel peut être perçu d'une manière différente par deux utilisateurs. Cela est en partie due à la cohérence entre l'observation et la représentation mentale de chaque utilisateur (l'idée qu'il se fait de ce qu'il voit). De ce fait, les utilisateurs doivent mettre de côté leur *incrédulité* (*suspension of disbelief*) pendant leur interaction avec l'environnement virtuel [Bates, 1994] afin qu'ils se sentent complètement immergés dans l'environnement de manière à mieux interagir avec lui.

Un environnement virtuel ne peut être crédible sans que les agents qui le peuplent le soient aussi. Les travaux qui ont été réalisés pour avoir des agents crédibles sont basés soit sur *la logique des comportements* qui utilise les techniques d'intelligence artificielle [Laird and Duchi, 2000], [Mac Namee, 2004], soit sur *le réalisme graphique* [Brogni et al., 2007].

Certains travaux se focalisent sur le réalisme des personnages et considèrent qu'une représentation réaliste peut être utile lorsqu'il ne s'agit pas d'applications de divertissement [De Rosis et al., 2004]. D'un autre côté, si les personnages sont d'apparence réalistes mais qu'ils se comportent d'une manière non naturelle, alors l'illusion de réalité est perdue [Metoyer and Hodgins, 2000].

Pour Sengers [Sengers, 1999] le réalisme ou bien l'illusion de la vie ne sont pas forcément nécessaires pour que les agents soient crédibles. En effet, les personnages de dessins animés sont tout à fait crédibles même s'ils sont loin d'être réalistes [Neal Reilly, 1996] (par exemple, les personnages de Disney [Thomas and Johnston, 1981]). Sengers définit les agents crédibles par des agents ayant des comportements interprétables et compréhensibles par les utilisateurs en étant en accord avec le contexte de leur environnement [Loyall, 1997], [Neal Reilly, 1996], [Blumberg, 1996], [Lester and Stone, 1997].

Pour Ortony [Ortony, 2003], le comportement d'un agent doit être *consistant*, c'est à dire, que s'il se retrouve devant des situations similaires, son comportement doit être le même (avec une certaine variabilité), mais aussi *cohérent* en agençant les émotions, les motivations, et les actions des personnages virtuels aux changements de situation. Par exemple, un personnage qui souhaite s'asseoir va se diriger vers un banc public, mais s'il n'en trouve pas, il peut s'adapter en allant s'asseoir dans une terrasse de café.

Un comportement émotionnel d'un agent permet à l'utilisateur de mieux s'idenifier à un personnage. Laaksolahti et son équipe [Laaksolahti et al., 2001] ont montré que la simulation d'expressions émotionnelles cohérentes avec le contexte et la situation de l'agent le rendait plus compréhensible et plus crédible au regard de l'utilisateur.

Les travaux sur les Agents Conversationnels Animés (ACA) ou Embodied Conversational Agents (ECA) ([Cassell, 2000], [Egges et al., 2003] et [Pelachaud and Bilvi, 2003]) sont axés sur les différents moyens de communication non verbaux comme la gestuelle, les expressions faciales et les intonations de voix, afin d'améliorer la compréhension des personnages par l'utilisateur et donc leur empathie et leur crédibilité. Tous ces travaux ont souvent un lien étroit avec les émotions, leurs expressions et leur compréhension.

Il est aussi important d'observer des interactions sociales crédibles entre les agents car ces deniers évoluent rarement seuls dans un environnement virtuel (tel est le cas dans notre projet de simulation Terra Dynamica). En effet, selon Guye-Vuillème et Thalmann [Guye-Vuilléme and Thalmann, 2000] les personnages doivent être dotés d'une identité sociale en plus d'être dirigés par des buts ou des émotions. Ainsi les interactions entre les agents deviennent plus crédibles et réalistes.

2.4.1 Évaluation de la crédibilité

La crédibilité est une notion subjective difficile à évaluer. Cette évaluation peut se résumer à demander aux utilisateurs si les comportements des agents virtuels sont crédibles. Malheureusement, cette question est trop *subjective* et il est inapproprié de fonder l'évaluation de la crédibilité d'un système sur une seule question. En effet, les réponses des utilisateurs peuvent être influencées par leur connaissance (ou méconnaissance) ce qui peut fausser les résultats [Mac Namee, 2004].

Dans la plupart des travaux, l'évaluation de la crédibilité se fait en testant les applications avec un certain nombre d'utilisateurs. Ces derniers vont alors regarder l'apparence des personnages et leur comportements. Ils vont essayer de comprendre le comportement des personnages en s'identifiant à eux et en se mettant dans la même situation. La crédibilité sera alors mesurée selon le degré d'empathie que l'utilisateur aura développé envers les personnages.

Laaksolahti [Laaksolahti et al., 2001] propose une méthode qualitative pour mesurer la crédibilité. Il se base sur les descriptions des personnages que donnent les utilisateurs. Si ces derniers décrivent les personnages et leur vie en utilisant des termes riches émotionnellement, cela veut dire que les utilisateurs ont éprouvé de l'empathie envers les personnages et qu'ils les ont trouvé crédibles. Dans le cas contraire, si les utilisateurs hésitent dans leur description et trouvent que les personnages sont étranges, alors ils n'ont pas ressenti de l'empathie envers eux et ne les trouvent pas crédibles.

Certains auteurs considèrent que le bon déroulement de l'apprentissage dans un environnement virtuel de formation été un gage de crédibilité [Ganier et al., 2013].

Catégorie	Critère de satisfaction
Crédibilité	est-ce que les personnages donnent l'illusion de la vie pour les joueurs ?
Aptitudes sociales	est-ce que les agents sont capables d'avoir des interactions sociales cohérentes, crédibles et intéressantes ?
Domaine d'application	est-ce que l'utilisation du système crée une expérience satisfaisante et immersive ?
Problèmes de calcul	est-ce que le système s'exécute avec succès selon les critères de rapidité et d'efficacité ?
Production	quel est le degré de facilité pour les concepteurs de créer des agents pour les intégrer dans le système ?

TABLE 2.2 – Les différentes catégories d'évaluation et leur critères de satisfaction [Isbister and Doyle, 2002]

La crédibilité dépend aussi du type d'application (pédagogique, ludique). En effet, un personnage peut être considéré comme crédible dans une application de divertissement (comme pour les personnages de dessins animés), sans qu'ils le soient dans une autre application d'apprentissage (simulation de formation professionnelle) [Ortony, 2003].

Isbister et Doyle [Isbister and Doyle, 2002] ont proposé une méthode d'évaluation permettant d'évaluer des agents conversationnels pédagogiques dans un environnement virtuel dédié à l'apprentissage. Ils proposent de séparer le problème d'évaluation en catégories distinctes afin de pouvoir les évaluer individuellement. Le tableau 2.2 donne les différentes catégories d'évaluation avec les critères qui permettent d'avoir une évaluation réussie pour chaque catégorie.

La principale catégorie de ce modèle d'évaluation est *la crédibilité* des comportements des humains virtuels. L'évaluation du système est considéré comme réussie si les personnages se comportent en permanence d'une manière cohérente et crédible (qui est la définition des auteurs pour désigner des personnages donnant *l'illusion de la vie*).

Ce modèle d'évaluation a inspiré de nombreux travaux visant à faire une évaluation rigoureuse [Abbattista et al., 2002], [Ruttkay et al., 2002], [Hook, 2005], [Christoph, 2005], [Isbister and Höök, 2005], et sert de modèle pour le domaine plus général de la réalité virtuelle.

2.4.2 Conclusion

Un environnement virtuel n'est crédible que si les agents qui le peuplent ont un comportement crédible suscitant l'empathie des utilisateurs. Pour cela, les agents doivent avoir une bonne compréhension de leur environnement.

Dans le cas de l'évaluation d'un système de simulation d'une ville virtuelle par rapport à un modèle de représentation sémantique de l'environnement, l'approche présentée dans [Isbister and Doyle, 2002] est la plus appropriée. Nous devons évaluer la crédibilité des comportements des agents par rapport aux choix des objets qu'ils ont utilisé pour exécuter leurs plans et leur contexte. Nous devons aussi évaluer la crédibilité des interactions des agents avec leur environnement. Pour ce faire nous proposons un protocole d'évaluation dans lequel des observateurs doivent évaluer la crédibilité des agents en utilisant une méthode quantitative basée sur un questionnaire [Christoph, 2005]. Les participants

auront le choix entre des réponses ouvertes où ils seront libres de s'exprimer en utilisant leur propres termes pour qualifier ce qu'ils ont vu, et des questions fermées où ils devront répondre selon l'échelle de Likert de 1 à 5.

2.5 Conclusion

La prise en compte des modèles sémantiques connaît un grand essor au sein des environnements virtuels. L'introduction de la sémantique dans les environnements virtuels a donné naissance aux environnements virtuels sémantiques où les informations utilisées pour le développement et la description des objets de l'environnement vont au delà des données géométriques. Ces informations sémantiques rendent les objets virtuels plus compréhensibles par les agents qui les considéraient comme de simples formes géométriques, et où ils devaient passer par une phase d'interprétation pour déduire leur sens. Cette bonne compréhension de l'environnement permet aux agents d'avoir des comportements crédibles [Bates, 1994].

L'utilisation de l'information sémantique des EVs permet de définir une base de connaissances complexe sur l'environnement tels que les comportements, les aspects dynamiques, etc. Nous trouvons dans la littératures différents travaux qui abordent des solutions pour la création d'approches générales permettant de modéliser ce type d'information sémantique. Cependant, nous sommes loin d'une solution complète pour modéliser des environnements virtuels [Ibáñez and Delgado-Mata, 2011]. La plupart des solutions proposées utilisent des ontologies pour modéliser l'information sémantique parce qu'elles permettent de modéliser des relations très complexes qui ne peuvent êtres représentées avec d'autres types de représentation (comme les taxonomies ou les thésaurus) [Huhns and Singh, 1997]. Pourtant, même si ces solutions proposent aux agents les informations sémantiques nécessaires pour choisir les objets adéquats pour réaliser leurs actions, elles proposent des informations insuffisantes pour leur permettre de choisir *les meilleurs objets* pour réaliser leurs actions tout en étant adaptés à leur contexte. Cela aurait été possible s'il y avait un tri de tous les objets possibles proposant de réaliser une action donnée. En général ces solutions permettent de lever l'ambiguïté sur les entités de l'environnement et d'éviter des aberrations dans le comportement des agents.

Nous proposons dans notre approche de modéliser l'information sémantique qui nous permet de faire du raisonnement sur celle-ci afin de fournir des informations d'aide à la décision afin de faciliter le calcul des procédures de raisonnement, qui permettraient aux agents de faire le meilleur choix d'objet pour réaliser leurs actions. Nous proposons donc de représenter les informations sémantiques liées à la définition de l'utilisation des objets de l'environnement indépendamment de leurs informations géométriques ou permettant leur génération. Ainsi, nous avons représenté les fonctions des objets dans une ontologie de services de l'environnement. Notre représentation tire profit des différentes techniques utilisées dans le domaine du web sémantique. Notre objectif est donc de modéliser les concepts sémantiques de l'environnement qui concernent les services proposés par ses entités, ainsi que leur critères d'évaluation afin de pouvoir par la suite inférer une qualité de service *QoS* d'une manière générique applicable sur des entités et des services hétérogènes.

Notre modèle ontologique nous permet de définir les concepts sémantiques de notre environnement. Ainsi, cette représentation sera utilisée pour formater et interpréter l'information sensorielle des agents (perceptions sémantiques). L'avantage d'utiliser une ontologie pour représenter la sémantique de l'environnement est que ce modèle de représentation

soit tout à fait indépendante de l'environnement virtuel à formaliser, mais aussi de la technologie utilisée pour créer l'environnement virtuel. Ce qui nous permet d'avoir une représentation indépendante du système de développement de l'environnement virtuel.

Le formalisme basé sur le langage de description proposé par les ontologies et son expressivité reste la meilleure solution pour modéliser l'information sémantique. Cependant, l'utilisation des langages d'ontologie (par exemple, OWL ou RDF) permet de modéliser et de définir les concepts du domaine qui composent l'aspect statique de l'EV. La limitation de ce type d'approches basées sur les ontologies réside dans leur incapacité de modéliser la dynamique des concepts de l'EV (par exemple, les interactions ou les comportements des objets de l'EV). Souvent, des langues textuelle de script sont intégrés aux langages d'ontologie pour permettre la description de la dynamique des EVs. Nous n'allons donc pas utiliser l'ontologie pour modéliser la dynamique des concepts de l'EV qui sera définie séparément de notre modèle de représentation ontologique (voir chapitre Implémentation et mise en œuvre).

Dans des travaux similaires [Kapahnke et al., 2010, Van Oijen and Dignum, 2011], les auteurs démontrent que les modèles utilisant des ontologies sont bien adaptés pour rendre l'environnement plus compréhensible pour les agents et sont assez rapide pour les petits mondes virtuels avec une ontologie de taille moyenne (nombre réduit de triplets) et avec quelques dizaines d'agents. Nous devons donc proposer une architecture de notre modèle qui puisse passer à l'échelle car nous nous situons dans un environnement urbain peuplé d'un très grand nombre d'acteurs (piétons et véhicules).

Chapitre 3

Modèle de représentation sémantique d'un environnement virtuel urbain

3.1 Introduction

Ce chapitre est consacré à la définition et à la formalisation de notre modèle sémantique de l'environnement virtuel qui consiste à faire un modèle formel de représentation permettant de calculer une QoS d'une manière générique s'appliquant à tous les types de services. Nous proposons aussi des algorithmes de sélection d'objets de l'environnement triés selon cette QoS avant de les proposer au module d'IA de haut niveau.

Notre objectif est de faire une représentation de l'environnement proposant les éléments d'informations utiles et nécessaires aux modules d'IA des agents pour leur permettre d'avoir des comportements crédibles, grâce à la proposition du meilleur objet sélectionné avec la QoS la plus élevée qui permet aux agents de réaliser leurs actions en cohérence avec l'état sémantique de leur environnement (contexte). L'idée est de représenter les fonctions proposés par les objets de l'environnement par des services définis dans une ontologie de services, et de déduire quel est l'objet disponible qui présente la meilleure qualité de service à un moment donné de la simulation.

Notre modèle se classe parmi les modèles de représentation de l'environnement orienté-contenu [Ibáñez and Delgado-Mata, 2006], c-à-d, que le modèle sémantique de l'environnement permet la représentation de la sémantique des entités qui le composent (la fonction, la relation entre les objets, leur type, etc.). Ce qui permet d'avoir un accès direct au contenu sémantique du monde virtuel.

Notre modèle se veut être générique, c-à-d, qu'il soit indépendant du domaine d'application. Nous proposons un méta-modèle pour la représentation des fonctionnalités hétérogènes des objets de l'environnement en s'inspirant de l'ontologie OWL-S [Martin et al., 2004] utilisée dans le domaine du Web Sémantique, dont nous avons expliqué les fondements dans le chapitre 2. Nous avons rajouté dans l'ontologie de services les paramètres qui nous permettent d'évaluer les services et de calculer leurs qualités (QoS).

Notre modèle de représentation est indépendant de la plateforme d'application de l'environnement virtuel, générée pendant la phase de modélisation conceptuelle du processus

de développement du projet. Nous avons aussi séparé la représentation sémantique de notre environnement de sa représentation graphique pour que les agents puissent interagir directement avec la sémantique de leur environnement.

Nous aborderons dans ce chapitre le principe général de notre modèle et nos motivations qui nous ont poussé à développer une telle approche (section 3.2). Nous verrons le rôle de notre module dans l'architecture du projet Terra Dynamica et comment il s'articule avec les autres modules du projet. Nous préciserons par la suite notre modèle et le formalisme que nous avons utilisé (section 3.3). Nous commencerons par présenter l'ontologie de services (section 3.3.1), pour laquelle nous avons défini les concepts d'objet, service d'objet et les critères d'évaluation d'un service. Nous verrons par la suite comment nous avons défini la qualité de service QoS, qui représentent respectivement la fonction rendue par un objet de l'environnement avec une certaine qualité de service estimée par rapport à un nombre de critères à satisfaire (section 3.3.2) et nous avons défini chaque paramètre de calcul de la *QoS*. Nous avons précisé les algorithmes d'interaction entre les agents et leur environnement à travers les 3 modes d'interaction que nous avons défini (section 3.4).

3.2 Principe général du modèle

L'objectif de notre modèle est de représenter l'information sémantique de l'environnement afin de l'inclure dans le processus de raisonnement du module de décision des agents. Plus précisément, nous voulons aider les agents dans leurs choix d'objets qui leur permettent d'accomplir leurs actions de manière crédible (crédibilité dans l'exécution de l'action de l'agent).

L'architecture du projet Terra Dynamica s'articule autour de plusieurs modules interagissant entre eux : le module sémantique de l'environnement, les modules de haut niveau (comme le module décisionnel [Reynaud et al., 2012]) et les modules de bas niveau (comme le module de navigation [Simo Kanmeugne and Beynier, 2014]).

Le module sémantique de l'environnement se trouve au cœur de cette architecture. Il centralise toute l'information sémantique de l'environnement nécessaire pour le bon déroulement des plans des agents et de leurs comportements. Il représente une couche sémantique qui joue le rôle d'intermédiaire entre les agents et leur environnement.

Comme nous pouvons le voir sur la figure 3.1, le module sémantique est sollicité par le module de décision des agents à travers des requêtes. Le module de décision interroge le module sémantique afin qu'il lui propose les entités de l'environnement lui permettant de définir un plan d'action pour les agents ou bien de choisir le meilleur élément pour réaliser un plan d'action. Par ce dernier cas, nous distinguons deux situations :

- le cas où les agents veulent réaliser une action immédiate (jugée importante) et,
- le cas où les agents souhaitent réaliser une tâche de leur plan d'une manière opportuniste (lorsque l'occasion se présente à eux).

Le module sémantique de l'environnement renvoie au module de décision un ensemble d'objet qui permettent de répondre aux actions des agents. Ces derniers doivent êtres triés selon leur pertinence par rapport aux actions que les agents s'apprêtent à réaliser. Pour ce faire, nous avons proposé de calculer une qualité de service *QoS* proposée par les objets sélectionnés offrant le service recherché. Le module de décision choisira par la suite entre les meilleurs objets les mieux classés renvoyés par le module sémantique.

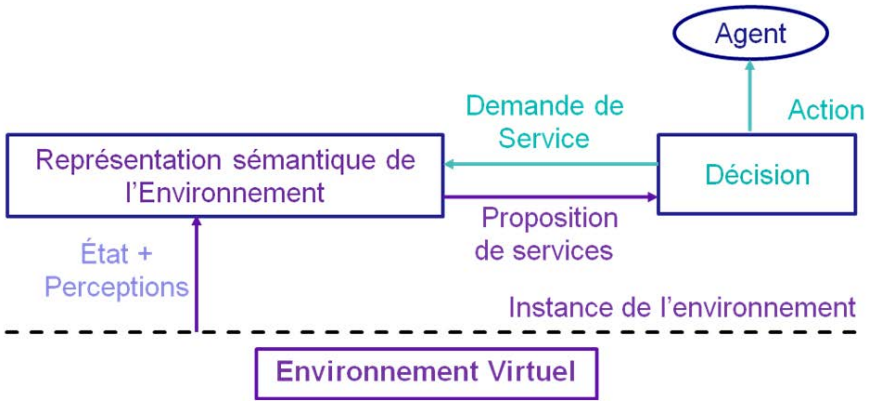


Figure 3.1 – Interactions du module sémantique de l'EV avec le module décisionnel des agents : le module décisionnel envoie des requêtes pour trouver un objet qui propose le service recherché, et le module sémantique lui renvoie une liste d'objets triés selon la *QoS* qu'ils proposent.

Nous précisons que le module décisionnel des agents tient compte d'autres informations sur l'environnement et l'état de la simulation en plus de la *QoS* pour prendre une décision (données géométriques et topologiques extraites de la Bdd, distance par rapport à un objet, présence d'une file d'attente, etc).

Nous distinguons trois types de requêtes envoyées par le module décisionnel des agents :
- Requête pour l'obtention d'un service donné (pour le mode interrogatif), qui sont les requêtes les plus fréquentes,
- Requête sur la cohérence d'utiliser un élément de l'environnement pour l'obtention d'un service donné (pour le mode réactif),
- Requête d'abonnement pour recevoir des notifications sur la proximité d'un objet proposant un service donné (pour le mode proactif).

Nous verrons dans le chapitre 4 comment sont implémentées ces requêtes.

Nous avons présenté dans le chapitre 2 les différents travaux qui abordent des solutions générales permettant de modéliser les informations sémantiques de l'environnement. Nous avons constaté que la meilleure façon de modéliser l'information sémantique de l'environnement était d'utiliser les ontologies. En effet, la plupart des solutions proposées dans la littérature utilisent des ontologies parce qu'elles permettent de modéliser des relations très complexes qui ne peuvent êtres définies avec d'autres types de représentation (comme les taxonomies ou les thésaurus) [Huhns and Singh, 1997]. Le formalisme du langage ontologique offre une flexibilité en étant indépendant de la plateforme de conception du système, mais aussi une adaptabilité de la solution en étant indépendant du domaine d'application.

Pour toutes ces raisons, nous avons utilisé une ontologie pour la représentation de la sémantique de l'environnement. L'ontologie de description de l'environnement que nous proposons offre un formalisme et une description explicite commune des entités de l'environnement. En effet, afin que nous puissions faire le même traitement de sélection des objets de l'environnement, ceux-ci doivent partager la même description formelle.

L'ontologie que nous proposons contient la définition de chaque type d'objet et la description de sa fonctionnalité que nous avons défini sous la forme d'un **service**, par

example, *"une chaise"* propose le service *"s'asseoir"*. Les services d'objet sont évalués par une qualité de service (*QoS*) quantitative qui nous permet de trier les objets entre eux.

Pour pouvoir calculer cette *QoS*, nous avons défini dans notre ontologie pour chaque service proposé par un objet de l'environnement des critères qui permettent d'évaluer le service. Par exemple, le critère *"confort"* permet d'évaluer le service *"s'asseoir"*. Les différents critères d'évaluation seront par la suite utilisés pour calculer les paramètre de calcul de la *QoS* que nous avons définis. En effet, nous avons défini trois paramètres généraux s'appliquant à tous les types de services de l'environnement qui nous permettent de calculer notre *QoS* pour chaque type d'objet. Nous avons ainsi défini le paramètre *"efficacité"* qui estime la pertinence du service proposé par un type d'objet par rapport à l'action. Nous avons aussi défini le paramètre *"difficulté"* qui nous permet d'évaluer les contraintes supplémentaires que l'agent devra prendre en compte pour obtenir son service et exécuter son action. Et enfin, nous avons défini le paramètre *"proximité"* qui permet d'estimer le degré de facilité pour avoir un service donné selon l'abondance des objets qui le proposent dans l'environnement.

Nous avons considéré qu'il était important que notre modèle de représentation soit *extensible*, c-à-d, que nous pouvons l'appliquer sur n'importe quel service en respectant un modèle de base que nous avons décrit dans un *méta-modèle*. Ainsi, nous pouvons appliqué notre modèle de description donné dans le méta-modèle à n'importe quel service quelque soit le domaine d'application.

Le méta-modèle que nous avons défini s'inspire de l'ontologie OWL-S. Pour les raisons que nous avons données dans le chapitre précédent, cette représentation est loin de pouvoir estimer la qualité d'un service. Nous avons donc inclus dans notre méta-modèle la description des paramètres *non fonctionnels* des services afin de pouvoir calculer leur qualité *QoS* d'une manière générique.

Les agents interagissent à différents moments de leurs vie avec leur environnement afin d'avoir les informations qui les intéressent aux moments opportuns. Nous avons donc défini trois modes d'interactions entre les agents et leur environnement. Le premier leur permet d'avoir l'information sémantique de l'environnement lors de la planification de leurs actions (*"mode Interrogatif"*). Le deuxième mode (*"mode réactif"*) leur permet de mettre à jour les informations obtenues lors de leur planification au moment où ils s'apprêtent à exécuter leurs actions (car il y a un temps entre le moment de la planification des actions et leur exécution). Et enfin, le troisième mode leur permet de recevoir des notifications de leur environnement afin de les prévenir de la proximité des objets qui proposent les services qui les intéressent afin de réaliser certaines actions d'une manière opportuniste (*"mode proactif"*).

Nous verrons dans ce qui suit (section 3.3) la définition formelle de notre modèle de représentation sémantique de l'environnement, comprenant notre ontologie de services (section 3.3.1), la définition des paramètres de calcul de la *QoS* et son calcul (section 3.3.2) et enfin les trois modes d'interaction des agents avec leur environnement (section 3.4).

3.3 Formalisme du modèle de représentation de l'environnement

Notre modèle de représentation de l'environnement repose sur une ontologie à deux niveaux qui sert de support à un mécanisme générique de raisonnement sur la dynamique

et le fonctionnement de l'environnement simulé (voir figure 3.2).

Nous proposons de représenter notre environnement dans une **Ontologie de services**. Celle-ci se compose de deux couches superposées :

- (i) une couche contenant la définition des services proposés dans l'environnement et les relations entre eux, ainsi que les différents types d'objet qui les proposent,
- (ii) une couche méta contenant la définition des concepts généraux de l'environnement et les relations qui existent entre eux, mais aussi les axiomes et les contraintes qui nous permettent de faire un traitement unifié sur les éléments de l'environnement.

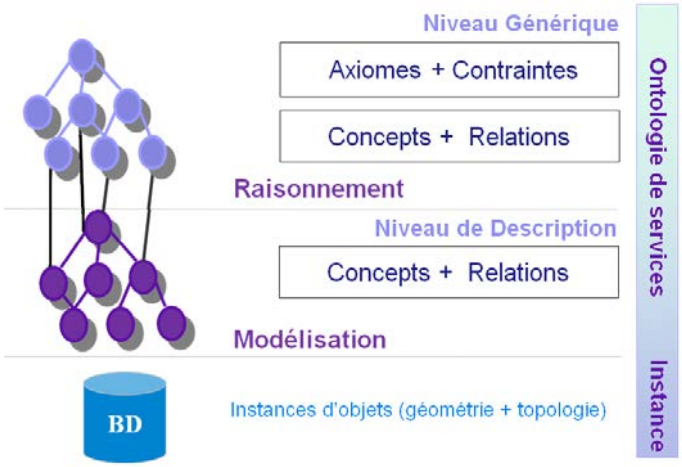


Figure 3.2 – Ontologie de services : une couche méta contenant les concepts génériques du modèle, et une couche sémantique contenant les concepts de l'environnement et les relations qui existent entre eux.

Notre modèle de représentation du monde ne dépend pas de la définition des instances du monde mais de celles des concepts définis dans notre ontologie. En effet, l'ontologie de services décrit d'une manière statique l'environnement et ne reflète pas l'état de ses instances à un instant donné (par exemple, l'instance d'objet *"distributeur de billets"* est occupé). Cela veut dire que les relations que nous avons définies dans notre ontologie ne sont pas entre les instances de l'environnement (entité concrète de l'environnement) ce qui la rendrait dynamique, mais entre les concepts de l'ontologie que nous avons définie.

Même si nous avons une représentation statique de notre ontologie de l'environnement, elle reste néanmoins utilisée pour faire un raisonnement sur la dynamique de l'environnement. En effet, elle permet de calculer une qualité de service proposé par un élément de l'environnement selon un contexte donné de la simulation.

3.3.1 Ontologie de Services

L'ontologie de services modélise tous les services de l'environnement ainsi que les relations qui existent entre eux. Cette représentation fait le lien entre les services du monde et les objets qui leur sont associés (l'objet qui propose le service) et les relations qui existent entre les objets du monde (hiérarchiques, dépendances, etc).

Dans notre modèle, une ontologie est une paire $\mathcal{O} =< \mathcal{C}, \mathcal{R} >$, où $\mathcal{C}$ est l'ensemble des concepts et $\mathcal{R}$ est l'ensemble des relations binaires.

Chaque relation $r \in \mathcal{R}$ est caractérisée par un ensemble de paires de concepts. Ainsi, nous notons $(c_i, c_j) \in r$ avec $r \in \mathcal{R}$ quand les deux concepts c_i et c_j sont reliés par la relation r dans l'ontologie.

L'ontologie de services n'est pas limitée à des relations de hiérarchie (isa) ou de méronymie ($partof$), mais nous avons défini aussi des relations spécifiques à notre objectif de représentation.

3.3.1.1 Couche Méta (méta-modèle)

Notre but est de proposer un modèle générique pouvant s'adapter à n'importe quel environnement. Nous avons donc proposé un modèle de représentation des différents concepts de l'environnement servant de standard pour définir un type d'objet et le(s) service(s) qu'il propose. Cette couche servira de modèle pour définir n'importe quel environnement de différents domaines d'application. Ainsi, les concepts liés à un environnement vont hériter des concepts généraux définis dans notre couche méta du modèle de représentation de l'environnement.

Le méta-modèle de notre représentation contient les concepts généraux de l'environnement $\mathcal{C}$ (voir figure 3.3), composé de trois sous-ensembles $\mathcal{C} = S \cup O \cup C_r$ où :
- S l'ensemble de tous les services,
- O l'ensemble de tous les types d'objet de l'environnement,
- C_r rassemble les critères d'évaluation de la qualité de service rendu par un type d'objet.

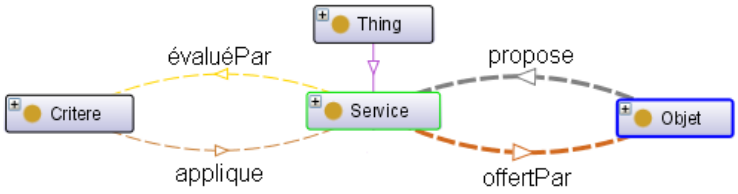


Figure 3.3 – Concepts généraux de l'ontologie de services «Niveau méta».

3.3.1.2 Définition des Types d'objet

Les objets de l'environnement ont un type que nous avons défini dans l'ensemble $\mathcal{O}$. Un type d'objet propose un/ou plusieurs services. Chaque type d'objet propose un service avec une certaine qualité (QoS). En effet, plusieurs types d'objet peuvent proposer le même service et seront différenciés par la qualité du service pour chaque type d'objet.

Par exemple, l'objet $o_1 =$*"chaise"* et $o_2 =$*"banc public"* proposent le même service $s =$*"s'asseoir"* sauf que leur qualité de service diffère.

3.3.1.3 Définition des services

Nous avons représenté la fonctionnalité d'un objet par des **services d'objet**. Un service est un tuple $\langle nom, C_r(s) \subset C_r, O(s) \subset O \rangle$ tel que nom est le nom du service, $C_r(s)$

l'ensemble des critères d'évaluation du service s et $O(s)$ l'ensemble des types d'objets proposant ce service.

Nous avons défini la relation $propose(o, s) \in O \times S$ entre les services et les types d'objet. Elle permet de dire qu'un type d'objet o propose un service $s \in S$ tel que $o \in O(s)$.

Notons que les types d'objets proposent uniquement des services d'actions $\forall o \in O, \forall s \in S, o \in O(s) \Rightarrow s \in S_a$.

Par exemple, le service *Manger* peut être proposé par plusieurs types d'objets $O(Manger) =$ {*Restaurant, Fast-Food, Cantine, etc*}, ce service sera évalué selon les critères $C_r(Manger) =$ {*Coût, Temps, Quantité, Qualité, Ambiance*}.

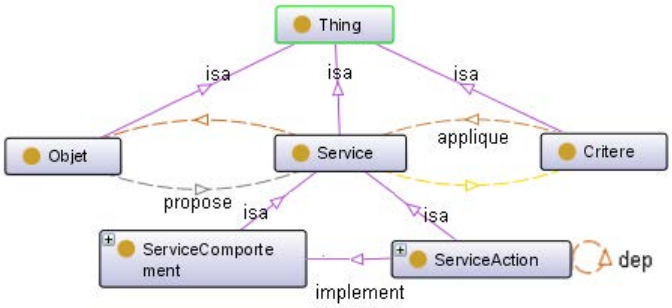


Figure 3.4 – Hiérarchie de Services : Services de comportement S_c et Services d'action S_a.

Nous avons défini deux niveaux de hiérarchie de services S_c et S_a, comme nous pouvons le voir sur la figure 3.4.

- S_c est l'ensemble des services qui permettent d'avoir un comportement et de répondre à une motivation de l'agent,
- S_a l'ensemble des services permettant de faire une action.

Avec $S = S_c \cup S_a$ et $S_c \cap S_a = \phi$.

Les services d'action S_a et de comportement S_c sont reliés par la relation $implement(s, s') \in S_a \times S_c$, qui veut dire qu'un service de comportement peut être obtenu (implémenté) par un ou plusieurs services d'action.

Par exemple le service $s = Manger \in S_c$ est un service de comportement qui peut être obtenu par différentes manières (implémenté avec des services d'action différents) $s_1 = CasserLaCroute$, $s_2 = Grignoter$, $s_3 = SeRestaurer$, etc, avec $s_1, s_2, s_3 \in S_a$.

Nous avons pour chaque service de comportement au moins un service d'action qui l'implémente $\forall s \in S_c, \exists s' \in S_a$ tel que $implement(s', s)$, sachant qu'un service d'action est relié à un seul et unique service de comportement $\forall s'' \in S_a, \exists! s \in S_c$ tel que $implement(s'', s)$.

Les services sont *non comparables* entre eux c'est à dire qu'un service d'action (ou de comportement) ne peut pas être implémenté par un autre service d'action (ou de comportement)

Les agents n'expriment pas leurs préférences explicitement, ils peuvent donc demander un service de comportement en exprimant une action motivée par un besoin comme par exemple *la faim*. Dans ce cas il va avoir le comportement de $Manger \in S_c$. Or, il existe différentes façons *de manger* comme *{grignoter, casser la croute , savourer}* $\in S_a$, etc. Nous

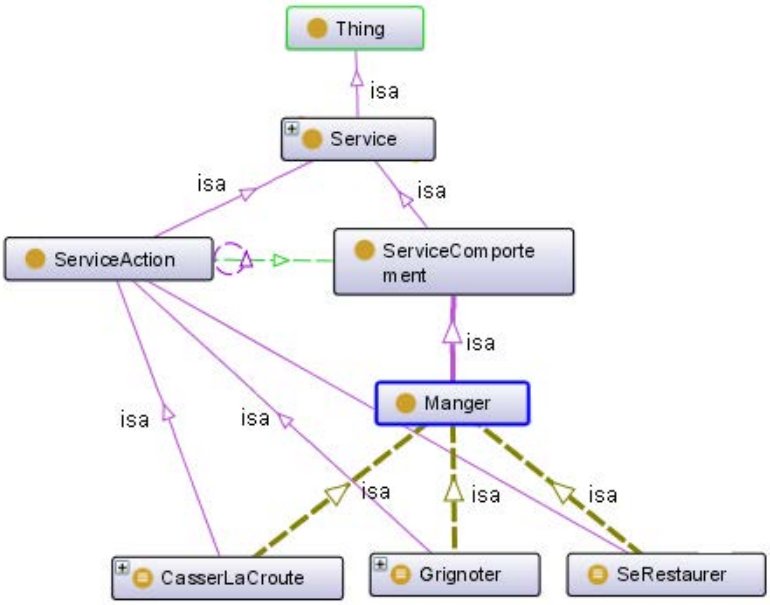


Figure 3.5 – Ontologie de services : exemple du service «Manger»

pouvons donc proposer les différents types d'objets qui permettent d'avoir ces différentes déclinaisons du service *Manger* (voir figure 3.5).

Un agent peut demander un service d'action $s \in S_a$ telle que *Grignoter*. Dans ce cas nous devrons lui proposer des types d'objets offrant ce service particulier, par exemple une *Boulangerie* ou un *Distributeur*.

3.3.1.4 Dépendance entre les services

Nous avons défini une relation non-hiérarchique de *dépendance* $dep(s, s')$ entre deux services s et s' proposés par le même type d'objet tel que, $dep(s, s') \in 2^S$, avec $o \in O(s)$ et $o \in O(s')$. Par exemple, le service *Aller aux toilettes* proposé par l'objet *Restaurant* dépend du service *se Restaurer* proposé par le même objet *Restaurant* car nous ne pouvons pas utiliser le service *Aller aux toilettes* du restaurant si nous n'avons pas l'intention de manger. Nous utiliserons la relation de dépendance entre les services dans le calcul de la qualité de service (section 3.3.2).

La relation *dep* est réflexive ($\forall s \in S_a, dep(s, s) = vrai$), transitive ($\forall s_1, s_2, s_3 \in S_a, \exists o \in O(s_1) \wedge o \in O(s_2) \wedge o \in O(s_3), dep(s_1, s_2) \wedge dep(s_2, s_3) \Rightarrow dep(s_1, s_3)$) et anti-symétrique ($\forall s_1, s_2 \in S_a, \exists o \in O(s_1) \wedge o \in O(s_2), dep(s_1, s_2) \wedge dep(s_2, s_1) \Rightarrow (s_1 = s_2)$).

3.3.1.5 Définition des critères d'évaluation des services C_r

Les services sont évalués selon un ensemble de critères. Un critère peut être partagé par plusieurs services comme par exemple *le Coût*, ou il peut être propre à un service, par exemple *Qualité des plats* qui est réservé aux services de restauration.

Soit C_r l'ensemble des critères. Chaque service $s \in S$ sera évalué selon un sous ensemble de C_r que nous notons $C_r(s) \subset C_r$.

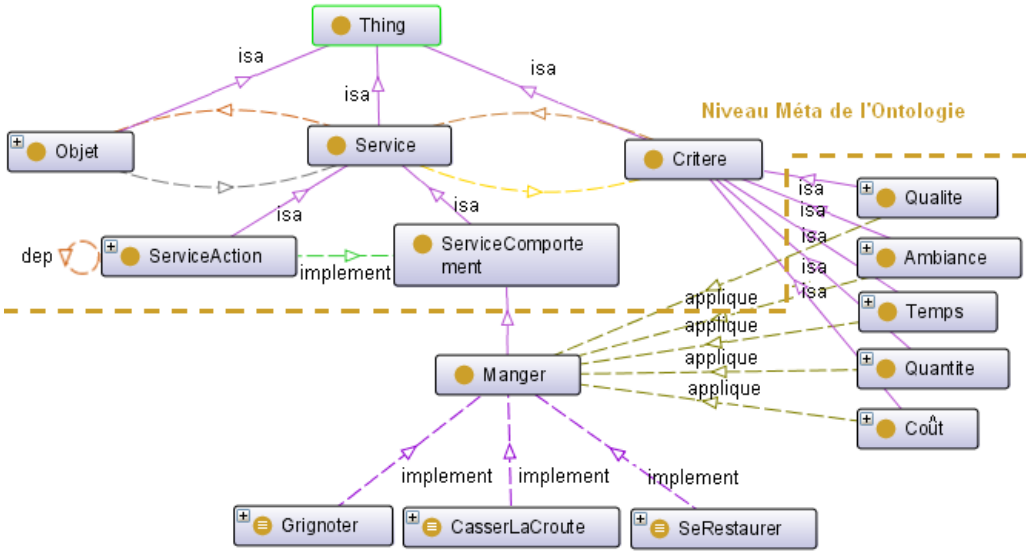


Figure 3.6 – Ontologie de services : Les différents critères d'évaluation appliqués au service «Manger»

L'ensemble des critères $C_r(s)$ d'un service d'action $s \in S_a$ est un sous ensemble des critères $C_r(s')$ du service de comportement $s' \in S_c$ qu'il implémente ($\forall s \in S_a, s' \in S_c, implement(s, s') \Rightarrow C_r(s) \subseteq C_r(s')$).

Nous avons défini une relation non-hiérarchique *applique* entre les services et les critères, que nous notons $applique(c, s) \in C_r \times S$, qui veut dire que le critère c est *appliqué* pour évaluer le service s (voir figure 3.6), donc $c \in C_r(s)$.

Nous avons défini pour chaque critère $c \in C_r$ un ensemble de valeurs noté $V(c)$. Cet ensemble contient toutes les valeurs possibles d'un critère c donné. Nous aurons par exemple les valeurs $V(coût) = \{cher, pas\ Cher, gratuit\}$.

3.3.2 Qualité de Service *QoS*

La difficulté de proposer un modèle de représentation unifié pour un environnement hétérogène est de pouvoir faire un calcul qui s'applique à n'importe quel service de l'environnement tout en gardant la pertinence du résultat. Pour ce faire, nous avons identifié les informations sur lesquelles vont se baser les agents pour prendre leur décision en les transposant sur les différents services existant dans l'environnement pour ne retenir que les plus importants pour les agents. Ces données vont nous permettre de calculer la qualité de service *QoS* par type d'objet (élément de l'environnement), ce qui nous permet de trier les types d'objet proposant le même service pour présenter le plus intéressant à l'agent.

La Qualité de Service *QoS* est calculée en fonction des paramètres suivants :
- **L'efficacité**, permet d'évaluer si un élément de l'environnement répond au besoin de l'agent et lui permet de réaliser sa tâche.
- **La difficulté**, permet d'estimer les contraintes supplémentaires à rajouter pour accéder au service souhaité en utilisant un élément de l'environnement.
- **La proximité**, correspond à la proximité d'un service en fonction de l'abondance des éléments de l'environnement le proposant.

Nous avons défini les concepts qui nous permettent de calculer les paramètres de la *QoS* dans le niveau méta de notre ontologie. Ces calculs tiennent compte du contexte de l'environnement, ce qui nous permet de trier les éléments de l'environnement (proposant un service recherché par l'agent) selon leur pertinence. Par exemple, un agent cherchant le service *Casser la croute* et se trouvant à proximité d'une *Boulangerie* et d'un *Distributeur*, choisira plutôt la *Boulangerie*. Par contre, s'il n'y a qu'un *Distributeur* à proximité il n'aura pas le choix et devra utiliser le *Distributeur*.

Nous verrons dans la section 3.3.2 comment nous avons combiné ces paramètres pour calculer la *QoS* pour un couple (type d'objet, service).

Notons que la *QoS* est indépendante des préférences des agents (nous avons défini un seul type de calcul pour n'importe quel type d'objet de l'environnement ou de service).

3.3.2.1 Paramètres de calcul de la *QoS*

a. Efficacité

Le paramètre d'efficacité que nous avons défini ici a pour objectif de nous permettre d'estimer si le service est pertinent par rapport à l'action souhaitée à travers la concordance des valeurs des critères d'évaluation. Par exemple, l'obtention rapide d'une barre de chocolat du distributeur augmente *l'efficacité* de l'objet *distributeur* pour le service *grignoter* grâce au critère *temps*. Le paramètre d'efficacité permet ainsi de faire la différence entre deux objets proposant le même service. Par exemple, si le but de l'agent est de *casser la croûte*, nous lui proposerons plutôt un *fast-food* qu'un *distributeur* qui est désavantagé en terme de *quantité* et de *qualité* de la nourriture par rapport au *fast-food*.

L'efficacité d'un type d'objet proposant un service $s \in S_a$ recherché par un agent est le rapport entre le nombre de critères satisfaits par le service proposé par le type d'objet sur le nombre total des critères d'évaluation du service s.

Pour ce faire, nous avons défini les ensembles suivants :

Critères de qualité $QC(o, s)$

Soit $QC(o, s) \subseteq C_r(s)$ l'ensemble des critères qui s'appliquent pour évaluer le service s proposé par un type d'objet $o \in O$ tel que $s \in S_a, o \in O(s)$ et $propose(o, s)$.

Par exemple, le service $s =$ *casserLaCroute* proposé par le type d'objet $o_1 =$ *Distributeur* aura comme critères l'ensemble $QC(o_1, s) = \{$*Coût, Temps, Qualité, Quantité*$\}$ et le type d'objet $o_2 =$ *Fast-Food* proposant le même service s aura comme critères l'ensemble $QC(o_2, s) = \{$*Coût, Temps, Qualité, Quantité, Ambiance*$\}$.

Valeurs de qualité d'un service $V_q(s, c)$

Nous avons défini un ensemble de valeurs sur les critères des services que nous avons noté $V_q(s, c) \subseteq V(c)$ tel que $c \in C_r(s)$. L'ensemble $V_q(s, c)$ contient toutes les valeurs possible que peut prendre un service pour l'un de ses critères. Par exemple, le critère $c =$ *Quantité* du service $s =$ *casserLaCroute*, aura une valeur dans l'ensemble $\{$*Moyenne, Petite*$\}$.

Estimation de la qualité d'un service $EQ(s)$

Nous avons défini un ensemble $EQ(s)$ *(Estimated Quality)* contenant toutes les combinaisons des couples (critère, valeur) possibles pour un service s, avec $EQ(s) = \{(c, v) \in C_r(s) \times V_q(s, c)\}$. En reprenant l'exemple précédent, $EQ(casserLaCroute)$ contient l'ensemble {*(Quantité, Petite), (Quantité, Moyenne)*}.

Effets d'un service par type d'objet $EQ(o, s)$

Soit $EQ(o, s)$ l'ensemble des effets attendus par l'utilisation d'un type d'objet $o \in O$ pour l'obtention d'un service donné $s \in S_a$, avec $EQ(o, s) = \{(c, v)/c \in QC(o, s) \wedge v \in V_q(s, c)\}$.

Par exemple, le service $s =$*casserLaCroute*, proposé par l'objet $o =$ *Distributeur* aura comme résultat $EQ(o, s) =$ { *(Coût, Pas Cher), (Temps, Rapide), (Quantité, Petite)*}.

L'efficacité $Eff(o, s)$ d'un type d'objet o par rapport au service s recherché est donc calculée comme suit :

$$Eff(o, s) = \frac{|EQ(s) \cap EQ(o, s)|}{|QC(o, s)|}$$

Remarque : Dans cette formule, comme dans celles qui suivent, nous noterons par $|X|$ le nombre d'éléments d'un ensemble X.

b. Difficulté

Le paramètre de difficulté permet d'éviter aux agents d'ajouter une action supplémentaire à leur plan en évitant de leur proposer un service dépendant d'un autre service proposé par le même objet. Par exemple, le paramètre *difficulté* d'une *chaise d'un café* diminuera la *QoS* du service *s'asseoir* car ce service dépend d'un autre service qui obligera l'agent de consommer pour pouvoir s'y installer. De ce fait, cette nouvelle action devra être prise en compte pour replanifier le plan de l'agent sinon cette nouvelle action mettra en péril la crédibilité du comportement de l'agent.

Le paramètre de difficulté est calculé en fonction des conditions supplémentaires à rajouter pour obtenir un service. Nous considérons qu'un type d'objet propose un service avec une difficulté lorsque celui-ci dépend d'un autre service proposé par le même type d'objet.

La difficulté $Diff(o, s)$ du service s proposé par le type d'objet o et ayant une relation de dépendance ($dep(s, s')$) avec un autre service s', est le taux de critères en plus à considérer (critères de s') pour obtenir le service s.

$$Diff(o, s) = \frac{|C_r(s')| - |QC(o, s) \cap QC(o, s')|}{|C_r(s')|}$$

La difficulté n'est pas prise en compte dans le cas où l'agent avait prévu une action correspondant au service s' dans son plan initial. Dans ce cas, la solution proposée permettant d'avoir les deux services sera considérée comme une solution optimisée et le paramètre de difficulté sera égal à *nul*. Par exemple, un touriste qui souhaite profiter de la vue en voulant *s'asseoir* confortablement quelque part et profiter des spécialités locales sera intéressé de *s'asseoir* sur la terrasse d'une *brasserie* et *manger*. Nous verrons dans la description des modes d'interaction (section 3.4) comment nous avons sélectionné un objet permettant de répondre à deux services pour proposer une solution optimisée.

c. Proximité

Nous avons mesuré la proximité d'un service en fonction du taux de présence des types d'objets qui le proposent.

Notre but est de permettre aux agents de faire la différence entre un type d'objet qu'ils peuvent trouver très souvent dans l'environnement ce qui le rend plus accessible et moins difficile à trouver qu'un type d'objet pas très fréquent dans l'environnement et qui peut être le seul moyen pour avoir un service donné.

Le choix entre plusieurs types d'objets proposant le même service tiendra compte du fait qu'un agent puisse choisir entre les instances d'un type d'objet parce qu'il peut en trouver partout dans son environnement (si il y a une file d'attente il se dirigera vers un autre objet du même type se trouvant pas très loin). Ou bien il n'a pas le choix et devra chercher une instance d'un type d'objet qui est le seul à pouvoir lui fournir le service avec la meilleure qualité.

La notion d'abondance ou de rareté nous permet d'estimer si l'agent a un grand choix ou pas par rapport à son contexte.

Par exemple, si un agent est intéressé par le service *se Restaurer* et que le meilleur type d'objet proposant ce service est un *Restaurant*, mais que celui-ci n'a pas un très grand taux de présence à l'endroit où se trouve l'agent, alors le type d'objet *Restaurant* sera considéré quand même comme une solution pertinente pour satisfaire son besoin.

L'abondance d'un type d'objet calculée concerne les objets du même type et proposant le même service .

Soit un type d'objet $o \in O$ proposant le service recherché s. Nous avons défini l'ensemble $O_p(o)$ contenant les instances d'objets ayant le type d'objet o. Soit $O_s(s)$ l'ensemble des instances des types d'objets $o' \in O$ proposant le service s.

La proximité d'un type d'objet o pour un service s $Prox(o, s)$ sera donc calculée en fonction du nombre de ses instances sur le nombre d'instances des types d'objets qui proposent le même service s :

$$Prox(o, s) = \frac{|O_p(o)|}{|O_s(s)|}$$

Cela représente le nombre d'objets du même type que l'objet sélectionné, sur l'ensemble de tous les objets de la zone sélectionnée proposant un service similaire au service recherché proposé par le type d'objet o.

3.3.2.2 Calcul de la QoS

La qualité d'un service proposé par un type d'objet de l'environnement tient compte des trois paramètres que nous avons cité plus haut avec une variante sur le poids de leur importance. Par exemple, un agent peut préférer avoir un type d'objet qui lui propose un service avec une efficacité maximale peu importe le moyen de le faire (paramètre de difficulté) ou la difficulté de le trouver (paramètre de proximité).

Nous avons donc calculé la QoS en faisant une somme pondérée des trois paramètres comme suit :

$$QoS(o, s) = \frac{Eff(o, s) + Diff(o, s) + Prox_z(o, s)}{3}$$

Une variante de la QoS peut être envisagée en donnant un poids à chaque paramètre. Cela nous permet de mettre en avant un paramètre par rapport aux autres en lui donnant un poids plus important. Par exemple, un agent peut préférer d'avoir un type d'objet

qui lui propose un service avec une efficacité maximale peu importe le moyen de le faire (paramètre de difficulté) ou la difficulté de le trouver (paramètre de proximité).

$$QoS(o,s) = \frac{p_1 Eff(o,s) + p_2 Diff(o,s) + p_3 Prox(o,s)}{\sum_i^3 p_i}$$

Dans l'implémentation de notre modèle (chapitre 4), nous avons considéré que tous les paramètres avaient le même poids d'importance ($p_1 = p_2 = p_3 = 1$).

3.4 Les interactions du module sémantique avec les modules d'IA

3.4.1 Mode interrogatif

C'est le mode d'interaction le plus courant. Lors de la phase de planification le module sémantique est sollicité par le module de décision des agents pour lui fournir des informations sur l'environnement. Le module sémantique va répondre à ces requêtes en proposant les éléments de l'environnement permettant de réaliser chaque action du plan en cohérence avec le contexte de l'environnement. Pour chaque action du plan de l'agent, le module sémantique va proposer un ensemble d'objets offrant un service qui permet de la réaliser. Les objets proposés sont triés et ordonnés selon la *QoS* qu'ils proposent. Le module de décision devra alors choisir celui qui correspond le mieux au plan de l'agent.

La plupart du temps, c'est l'objet en haut de la liste qui est choisi par le module de décision. La proposition de plusieurs solutions au module de décision nous évite d'imposer un choix aux agents. Sachant que le module sémantique propose des informations d'aide à la décision mais ne prend pas de décisions pour les agents. En effet, le module sémantique n'est pas informé du plan global des agents ni de leurs objectifs. De ce fait, il est important de permettre au module de décision de choisir parmi la liste des objets proposés celui qui convient au plan global d'un agent. Par exemple, pour une requête du module de décision portant sur l'action de *"Faire du shopping"*, le module sémantique peut lui proposer un *"Centre commercial"*, un *"Magasin de vêtements"* ou un *"Magasin de chaussures"* etc.

Comme nous pouvons le voir dans l'algorithme 1, la première étape consiste à chercher dans l'ontologie les objets qui proposent le service qui correspond à l'action recherchée. Pour cela, nous devons d'abord interpréter la requête du module d'IA en cherchant le / les services qui correspondent à l'action souhaitée ***(TrouverService())***. Nous aurons deux possibilités, soit le service correspondant est un service de comportement. Dans ce cas, nous allons chercher quels sont les services d'action qui l'implémentent. Soit c'est un service d'action, dans ce cas, nous chercherons directement les types d'objet qui le proposent. Pour chaque type d'objet que nous avons trouvé, nous allons calculer sa *QoS* et ses paramètres ***(CalculEff(), CalculDiff(), CalculProx(), CalculQoS())***. Nous pouvons alors trier les types d'objet selon leur *QoS* ***(TrierParQoS())***. Les instances d'objets trouvés dans la BD de l'environnement ***(TrouverInstance())*** seront présentés dans le même ordre selon la *QoS* de leur type d'objet ***(TrierInstanceObject())***.

Dans le cas où le module d'IA souhaite avoir une solution optimisée, il va demander au module sémantique les objets qui permettent de réaliser plusieurs actions à la fois. Dans ce cas, le module d'IA va envoyer dans la même requête une liste d'actions. Le module

sémantique lui proposera alors les objets qui permettent de réaliser plus d'une action apparaissant dans la requête. Le calcul de la QoS des objets proposant plus d'un service souhaité, ne prendra pas en compte le paramètre de *difficulté* qui aura la valeur nulle s'il y a une dépendance entre les services qu'il propose au même temps et apparaissant dans la liste des services souhaités (voir Algorithme 2).

Algorithm 1: Recherche des objets en réponses aux demandes du module d'IA

Input:
a l'action qui doit être réalisée
Srv liste de services d'action correspondants aux actions à réaliser
SO listes des couples (service, type d'objet) sélectionnés
SOInst liste des instances d'objet qui permettent de réaliser les actions souhaitées
Res tableau des résultats

//Interprétation de l'action pour chercher le service correspondant
s = TrouverService(a)
//Au niveau de l'Ontologie de services :
//Le service recherché est un service de comportement
if $s \in S_c$ **then**
  //Chercher tous les services d'action qui implémentent s
  $Srv \leftarrow s_a \in S_a$ avec $implement(s_a, s)$
else
  //Le service recherché est un service d'action
  $Srv \leftarrow s$
end
//Chercher les types d'objet qui proposent le service recherché
foreach $s_i \in Srv$ **do**
  $SO \leftarrow (s_i, o)$ avec $propose(o, s_i)$ et $o \in O$
end
//Calculer la QoS pour chaque couple (s_i, o)
foreach $(s_i, o) \in SO$ **do**
  $CalculEff(s_i, o)$
  $CalculDiff(s_i, o)$
  $CalculProx(s_i, o)$
  $CalculQoS(s_i, o)$
end
$SO \leftarrow TrierParQoS(SO)$
//Au niveau de la Base de données de l'environnement :
//Chercher dans l'environnement les instances d'objets trouvés
foreach $(s_i, o) \in SO$ **do**
  $SOInst \leftarrow TrouverInstance(s_i, o)$
end
$Res \leftarrow TrierInstanceObjet(SOInst)$
return Res

Nous verrons les détails d'implémentation des trois modes dans le chapitre 4 (section 4.3.7).

Algorithm 2: Calcul de la QoS pour les types d'objets permettant d'exécuter plus d'une action souhaitée

Input:
Srv liste de services d'action correspondants aux actions à réaliser
SO listes des couples (service, type d'objet) sélectionnés
SOInst liste des instances d'objet qui permettent de réaliser les actons souhaitées
Res tableau des résultats

```
foreach (s_i, o_i) ∈ SO do
    foreach (s_j, o_j) ∈ SO do
        //Trouver les objets qui permettent d'avoir plusieurs services souhaités
        if (o_i == o_j) then
            if dep(s_i, s_j) then
                Diff(s_i, o_i) = 0
            else
                if dep(s_j, s_i) then
                    Diff(s_j, o_j) = 0
                else
                end
            end
        else
        end
    end
end
//Calculer la QoS pour chaque couple (s_i, o) en mettant à 0 le paramètre difficulté
pour les objets qui permettent d'avoir plus d'un service souhaité
foreach (s_i, o) do
    CalculQoS(s_i, o)
end
SO ← TrierParQoS(SO)
//Au niveau de la Base de données de l'environnement :
//Chercher dans l'environnement les instances d'objets trouvés
foreach (s_i, o) ∈ SO do
    SOInst ← TrouverInstance(s_i, o)
end
Res ← TrierInstanceObjet(SOInst)
return Res
```

3.4.2 Mode réactif

Le module de décision des agents peut solliciter le module sémantique en dehors du moment de planification. Dans ce cas, les objets qui permettent de réaliser les actions des agents sont identifiés lors de la phase de planification. Cependant, entre le moment de la planification et le moment où l'agent s'apprête à réaliser son action, il se peut que l'objet change d'état (*occupé* : objet sollicité par un autre agent au même temps). Ce mode est souvent utilisé pour les types d'objets pouvant être sujets à des filles d'attente (distributeurs de banque, guichet) ou bien à utilisation individuelle (chaise).

Dans ce mode, le module sémantique peut soit proposer une autre instance du même type d'objet se trouvent à proximité (changer de guichet là où il y a moins d'agents en

attente par exemple) ou bien de proposer une autre alternative de type d'objet qui propose à cet instant là *une meilleure QoS*.

3.4.3 Mode proactif

Ce mode fait une recherche active d'un service à la demande du module de décision pour un agent. Cela intervient après avoir exprimé une demande d'abonnement à un service donné. Ce mode d'interaction est utilisé lorsque le module de décision considère qu'une action est peu prioritaire et/ou que les objets qui lui permettent de la réaliser sont plus difficiles à atteindre que les autres objets de son plan. De ce fait, le module de décision va considérer que l'action peut être réalisée à n'importe quel moment du temps d'exécution du plan d'un agent.

Le module sémantique proposera à l'agent abonné une instance d'objet qui propose le service recherché dès que celui-ci se trouvera à proximité. L'agent recevra alors une notification l'informant de la présence d'un objet lui permettant de réaliser son action en instance d'attente. Il pourra alors adapter son comportement en prenant en compte l'existence de l'objet qu'il lui a été proposé.

Exemple : Un agent se dirige vers son lieu de travail après avoir terminé de manger son sandwiche en cours de chemin. Il souhaite alors jeter la serviette en papier qu'il a utilisé. Considérant que c'est une action peu importante, et pouvant être exécutée à n'importe quel moment en cours de son chemin, sans nécessiter de faire un détour. Il (le module décisionnel de l'agent) va donc s'abonner au service qui va lui permettre de jeter ses déchets. Le module sémantique va chercher activement une poubelle se trouvant à proximité pour la proposer à l'agent. Le module sémantique ne va pas proposer plusieurs fois la même instance d'objet à l'agent si celui-ci la refuse.

3.5 Conclusion

Nous avons présenté dans ce chapitre le principe général et la formalisation de notre modèle, conçu pour la représentation de l'information sémantique de l'environnement, qui est utilisée pour l'aide à la décision afin de produire des comportements crédibles.

Notre modèle est fondé sur le principe de représenter les objets de l'environnement par leurs fonctionnalités que nous avons définies sous la forme de *services* représentés dans une ontologie. Nous nous sommes inspirés des techniques utilisées dans le web sémantique pour la représentation des services. Nous avons défini une ontologie de services partitionnée en deux couches superposées. La couche supérieure représente le méta-modèle de notre représentation qui contient les concepts généraux de notre modèle dont les concepts de domaine vont hériter. La couche inférieure contient la sémantique de l'environnement relative à notre domaine d'application. Notre représentation des services de l'environnement nous permet de calculer une qualité de service (QoS), utilisée pour trier entre eux les objets qui proposent le même service.

Nous avons défini trois paramètres représentant les critères non fonctionnels des services (l'efficacité, la difficulté et la proximité). De part leur aspect non fonctionnel, ces paramètres sont applicables à tous les services de l'environnement.

Le module de décision des agents obtient les informations sémantique sur l'environnement à travers des requêtes envoyées au module sémantique. Celui-ci propose une liste d'objets proposant le service qui correspond à une action donnée triée selon la qualité de service.

Afin que les agents puissent avoir une crédibilité dans leurs comportements, nous avons veillé à ce qu'ils aient la bonne information sémantique sur leur environnement au moment opportun de la simulation. Pour cela, nous avons défini trois modes d'interaction entre les agents (leur module de décision) et leur environnement (le mode interrogatif, réactif et proactif).

Notre modèle de représentation de la sémantique de l'environnement propose quatre originalités par rapport aux modèles existants.

La première réside dans la représentation sémantique des éléments de l'environnement. Nous avons représenté les différents types d'objets de l'environnement par leurs fonctionnalités qui sont définies par des services. Ainsi, nous avons séparé la représentation graphique de l'environnement de sa sémantique. Car notre but n'est pas de faire un modèle pour la génération d'un environnement virtuel ni pour sa visualisation. Nous avons voulu que les agents puissent interagir directement avec la couche sémantique de leur environnement pour pouvoir réaliser leurs actions. De ce fait, nous proposons directement aux agents les objets de l'environnement qui coïncident avec leurs actions.

La deuxième originalité réside dans la procédure de sélection des objets de l'environnement proposés aux agents pour qu'ils puissent réaliser leurs actions. Notre sélection d'objets est basée sur la qualité de service proposée par les objets pour un service donné correspondant à une action. La sélection des objets pour répondre à une requête d'action en se basant sur la *QoS* est une tâche fastidieuse due à l'hétérogénéité des objets qui proposent le même service. Nous avons défini trois paramètres généraux qui permettent d'évaluer la qualité d'un service par rapport à une requête. Le premier paramètre nous offre la possibilité d'évaluer si le service sélectionné correspond à une action donnée (Efficacité). Le deuxième paramètre évalue s'il y a des conditions supplémentaires pour accéder au service pour pouvoir répondre à une action donnée (Difficulté). Et enfin, le troisième paramètre nous permet d'estimer si les agents peuvent trouver facilement des objets qui proposent le même service afin de dire s'il a le choix pour trouver plus facilement un autre type d'objet qui propose le même service (Proximité). La description de ces trois paramètres est donnée dans la section 3.3.2.

La troisième originalité est dans l'interprétation de la requête des agents afin d'interpréter leurs préférences. En effet, le module de décision des agents n'exprime pas explicitement leurs préférences pour réaliser leurs actions, ni le but final de leur plan. Le module sémantique devra quand même proposer les services qui sont en adéquation avec les objectifs des agents. Nous avons donc proposé un modèle de représentation des services de l'environnement et une méthode d'évaluation de leur qualité qui interprète les préférences des agents implicitement à travers les actions qu'ils souhaitent réaliser. Ainsi, nous pouvons déduire des contraintes sur le temps, le coût, etc (nous pouvons interpréter par l'action "casser la croute" que le facteur temps doit être important pour l'agent qui n'a pas le temps de manger dans un restaurant).

Enfin, la quatrième originalité est dans le type d'interaction des agents avec la couche sémantique de leur environnement avec la proposition de trois types d'interactions (mode

interrogatif, réactif et proactif) qui permettent aux agents d'avoir les informations sur leur environnement au moment opportun.

Dans les chapitres 4 et 5 qui suivent, nous allons vérifier si les choix de conception que nous avons utilisés pour définir notre modèle sont pertinents. Nous verrons si les propositions du module sémantique permettent d'avoir des comportements crédibles par des observateurs humains, et nous évaluerons les caractéristiques techniques de notre modèle concernant le temps de traitement et le problème du passage à l'échelle.

Chapitre 4

Mise en Œuvre et Évaluation Objective

4.1 Introduction

Ce chapitre a pour objectif de décrire la mise en œuvre de notre modèle décrit dans le chapitre 3. Nous présenterons le prototype que nous avons utilisé lors de la phase de conception de l'architecture du projet Terra Dynamica (section 4.2). Nous nous attarderons par la suite sur l'intégration du module sémantique de l'environnement ayant été utilisé lors de l'évaluation de notre modèle présentée dans le chapitre 5. Nous décrirons les interfaces développées et nous aborderons notamment les problèmes d'implémentation que nous avons rencontrés. Enfin nous terminerons par donner les résultats de l'évaluation des performances obtenus par notre module sémantique de l'environnement (section 4.6).

Implémentation du modèle

Avant d'intégrer notre module sémantique de l'environnement, nous avons implémenté un premier prototype qui a découlé des travaux présentés dans [Harkouken-Saiah et al., 2011]. Nous présentons dans ce qui suit le modèle initial qui nous a permis de tester notre solution sur un exemple simple d'application. Nous avons pu alors tester les propositions émanant du module sémantique et les différents modes d'interactions entre l'environnement et les autres modules du simulateur (Navigation, Décision, etc).

4.2 Prototype

4.2.1 Présentation générale

L'idée principale de notre prototype est de valider notre approche de représentation de l'environnement et de la communication entre les différents modules du simulateurs. Grâce à ce prototype, nous avons pu définir les différentes informations sur l'environnement qui peuvent êtres modélisées dans notre module et qui pourront être utilisées par les autres modules du projet Terra Dynamica. Par la suite, nous avons défini les différents modes de communication nécessaires pour une simulation temps réel.

Nous nous intéressons dans notre travail aux données qui permettent aux agents de comprendre quel objet de l'environnement utiliser pour réaliser une action donnée. Ces données sémantiques représentent les fonctions et le type d'utilisation des objets (services

proposés par les objets de l'environnement)(voir la figure 4.1). Nous sommes partis du principe que chaque objet de l'environnement proposait un ou plusieurs services avec un certain nombre de contraintes d'utilisation et une qualité. Notre objectif est de formaliser les connaissances de l'environnement pour inférer de nouvelles informations utiles pour le processus de décision des agents.

Notre module sémantique permet d'interpréter les perceptions des agents. Ces perceptions sont appelées *«Perceptions Sémantiques»*, c'est-à-dire la représentation sémantique permettant d'accéder à un service de l'environnement. Le module décisionnel des agents s'appuiera sur l'interprétation de ces perceptions afin de choisir les actions des agents via trois modes d'interaction.

Ainsi, nous avons défini les perceptions des agents :

- lorsqu'ils s'interrogent sur l'objet qu'ils doivent utiliser pour réaliser une action «Mode Interrogatif» (pendant la planification),
- lorsqu'ils arrivent devant un objet et s'interrogent sur sa disponibilité «Mode Réactif » (pendant l'exécution du plan),
- et enfin lorsqu'ils ont une action désirée qu'ils veulent réaliser dès que l'objet qui leur permet de la faire se présente à eux «Mode Proactif» (opportunité dynamique).

Nous présentons dans notre prototype les deux modes d'interaction sollicités pendant l'exécution du plan des agents (mode Réactif et Proactif).

4.2.2 Fonctionnement et mise en œuvre

Nous avons proposé une ontologie pour décrire les services des objets de l'environnement. Cette description sémantique nous permet d'avoir une représentation formelle de la fonction des objets qui nous permettra par la suite d'appliquer un raisonnement pour déduire de nouvelles informations.

Le raisonnement appliqué à notre ontologie de services nous permet de déduire automatiquement et à tout instant de l'exécution, à partir de l'état des instances du modèle, les informations suivantes :

- quel objet permettant d'effectuer l'action souhaitée par l'agent,
- quelle est sa disponibilité et la qualité de service (QoS) associée.

Ainsi, les propositions du module sémantique doivent tenir compte des critères de l'agent et de l'état de l'environnement.

Dans notre prototype nous avons scripté les différents plans d'actions que pouvait faire notre agent et nous avons simulé les demandes du module de décision (pour le mode réactif et pro-actif).

Nous avons imaginé un scénario où notre agent souhaite poster un courrier et où il a le choix entre un bureau de poste et une boite aux lettres. Les services proposés par ces deux objets sont représentés comme dans la figure 4.1.

Notre agent est enregistré pour pouvoir recevoir des notifications lorsqu'il s'approche d'un objet lui permettant de poster son courrier (en mode Pro-actif). De plus, le module de décision de l'agent interroge le module sémantique avant d'utiliser un objet de l'environnement qu'il a prévu dans le plan de l'agent dès qu'il s'en approche pour vérifier sa disponibilité au moment où il s'apprête à l'utiliser (en mode réactif).

La figure 4.2 montre deux situations d'interaction (en mode réactif et proactif).

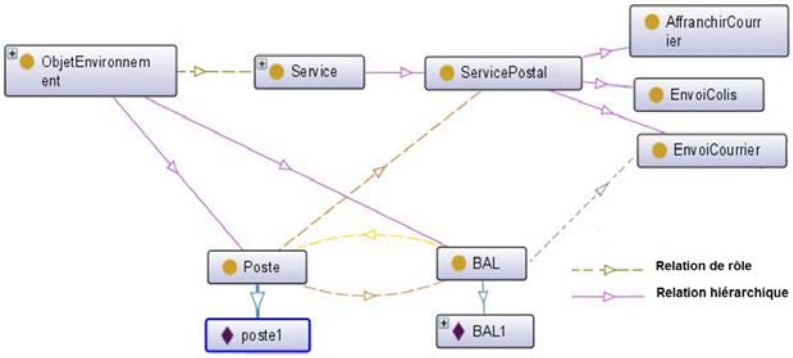


FIGURE 4.1 – Une partie de l'Ontologie de services : Service postal

(a) Mode Réactif (b) Mode Proactif

FIGURE 4.2 – (a) Le module de décision demande s'il peut traverser (b) Le module sémantique notifie à l'agent la proximité d'une boite aux lettres

4.2.3 Premiers résultats et conclusion

Les résultats observés sur cet exemple réduit simulé avec notre prototype sont suffisants pour faire une première évaluation de notre approche. Grâce aux données sémantiques (qualité de service) envoyées au module de décision de l'agent, nous avons pu observer que l'agent choisit les bons objets pour réaliser ses actions (dans notre exemple une BAL ou le bureau de poste pour poster un courrier). Les déplacements observés sont cohérents avec l'environnement. De plus, la communication avec le module sémantique rend l'agent plus réactif et renforce la crédibilité de la simulation. Par exemple, l'agent respect le code de la route (feux, passages piétons).

L'introduction des trois modes d'interactions entre le module sémantique de l'environnement et le module de décision des agents nous permet d'observer des comportements opportunistes (avec le mode pro-actif) et une réactivité des agents à ce qui se passe dans

leur environnement (mode réactif) ce qui contribue dans la crédibilité de la simulation.

4.3 Intégration du modèle dans la plate forme du projet Terra Dynamica

4.3.1 Le projet Terra Dynamiaca

Le projet Terra Dynamica, soutenu par les pôles de compétitivité Cap Digital et Advancity, et sélectionné par le FUI (Fonds Unique Interministériel) 8, regroupe plusieurs partenaires industriels et académiques, dont en particulier THALES, porteur du projet, et l'UPMC (équipe Systèmes Multi-Agents du LIP6).

Ce projet vise à développer les technologies permettant d'animer une ville virtuelle réaliste en la peuplant d'habitants et de véhicules réalisant leurs activités habituelles, mais aussi capables de réagir aux événements et de s'adapter aux évolutions de l'environnement de manière crédible.

Les domaines d'application de ces technologies sont nombreux ; Terra Dynamica vise en particulier les applications liées à la sécurité, au jeu vidéo, à l'urbanisme, et aux transports. Le consortium inclut des partenaires spécifiques pour ces thématiques. En particulier, le laboratoire LIP6 de l'UPMC (équipe SMA), en collaboration avec THALES, se charge dans le noyau du moteur d'animation comportementale permettant de doter les agents (piétons, véhicules, etc.) de la simulation interactive d'une Intelligence Artificielle afin d'obtenir des comportements crédibles. Les défis technologiques sont nombreux, incluant par exemple le haut niveau d'interactivité visé et le nombre élevé d'agents à gérer, ce qui oriente vers des architectures d'agents autonomes pouvant gérer des granularité variables (individu, groupe, foule).

4.3.2 Architecture du projet

Le Projet Terra Dynamica est composé de plusieurs démonstrateurs indépendants, au moins un par domaine applicatif. Chaque composant « contrôleur de démonstration » est entièrement développé par son partenaire responsable. Tous les démonstrateurs utilisent le composant **MAC** (*Moteur d'Animation Comportemental*) qui est considéré comme le *"noyau"* ou *"Kernel"* du projet. Le travail réalisé dans cette thèse se trouve dans le MAC situé au sein de l'application THALES (voir cadre en haut à gauche sur la figure 4.3)

Le MAC contient quatre modules essentiels pour le fonctionnement du moteur d'animation qui sont :

- *Actor :* ce module permet la création et la destruction des agents virtuels, il interprète les perceptions et exécute les actions,
- *KernelBase :* définit les services de base du MAC : contrôle d'exécution, les états de la simulation, les identifiants uniques, support à la mise au point, logging, etc,
- *KernelDataModel (KDM) :* se charge du partage de données et des services entre les modules de calcul (en vert sur la figure 4.4) avec les modules externes au MAC,
- *KernelGlue :* s'occupe de l'ordonnancement des autres modules sauf KBase et le KDM.

Les autres composantes du MAC sont définis comme suit :

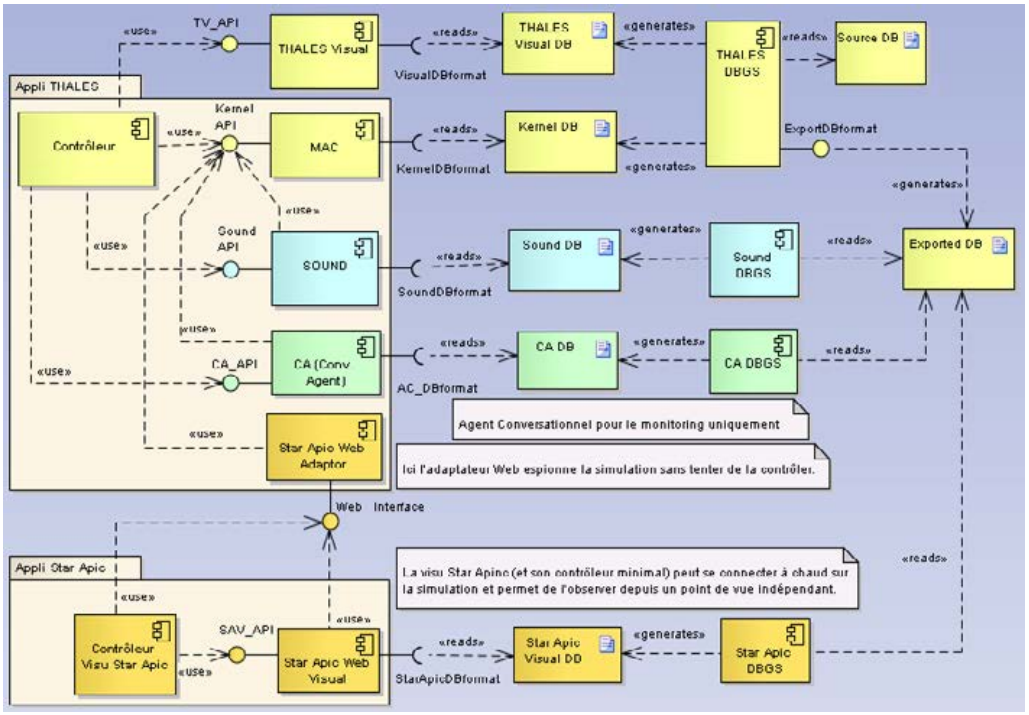


Figure 4.3 – Schéma de la structure globale du Projet Terra Dynamica

– **Environnement :** le module environnement comporte le module *Sémantique de l'environnement* définit dans cette thèse. Ce module maintient une représentation sémantique, logique et géométrique du monde simulé, en particulier des lieux et les objets significatifs. Il permet éventuellement de faire évoluer l'état de certains objets, en fonction des actions des agents. Ces informations sont transmises aux acteurs à travers le module de décision (Module d'IA de haut niveau), afin de localiser certains lieux ou objets où/avec ils vont accomplir des actions, ou bien alors pour connaître l'état de certains objets.

– **Scénarisation :** le module de scénarisation peut créer des agents, initialiser leurs variables de personnalisation, leur affecter un rôle, et une destination. Il peut également modifier des variables de personnalisation d'acteurs en cours de jeu, voire des variables d'inventaire, ou soumettre au module de décision des propositions de comportement. Le module de scénarisation peut également créer, modifier et détruire des effets spéciaux. Enfin, il peut aussi agir sur l'état des objets.

– **Les modules d'IA de haut niveau :** L'architecture décisionnelle *FlexMex* [De Sevin et al., 2012] conçue pour le projet Terra Dynamica au sein du LIP6, est prévue pour comporter plusieurs modules gérant des aspects différents du comportement des agents. Elle permet d'intégrer un ensemble de modules quelconque en entrée du module de décision en effectuant des propositions de comportement comme par exemple les différents modules qui ont été intégrés dans le projet Terra Dynamica. L'architecture *FlexMex* peut gérer à la fois des comportements réactifs à court terme, affectifs et des capacités cognitives à long terme.
  – **le module Cognitif :** ce module est chargé d'élaborer des plans pour atteindre

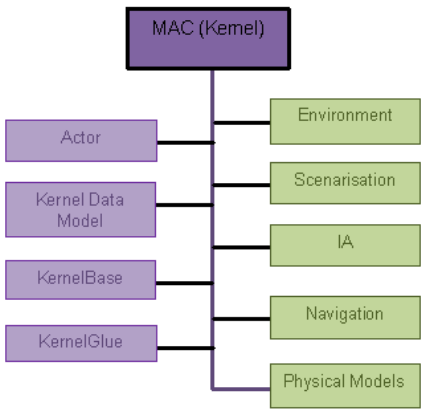


FIGURE 4.4 – Composantes du MAC et leur interdépendance

des buts. Il est doté de capacité d'anticipation et de planification à long terme.
- **le module Motivationnel :** il propose des comportements en relation avec les variables internes d'un agent telle que la faim, la soif, la fatigue,etc. C'est un module réactif, fonctionnant sur le principe de l'homéostasie qui pousse les agents à satisfaire leurs motivations courantes, leur donnant une certaine autonomie.
- **le module Affectif :** il propose des comportements affectifs en réaction aux événements de l'environnement, tout en tenant compte des caractéristiques individuelles des agents. Il s'agit également d'un module réactif [Campano, 2013].
- **le module de Coordination :** il est chargé de gérer la coordination de plusieurs agents souhaitant atteindre un but commun. Plus particulièrement, ce module est chargé de gérer les patrouilles d'agents [Poulet, 2013].
- **le module de Décision :** ce module transforme les propositions de comportements envoyées par les 4 modules cités précédemment en actions ordonnées [Reynaud, 2014].

- **le module de Navigation :** ce module est chargé de planifier le trajet d'un agent pour se rendre d'un point à un autre de l'environnement [Simo Kanmeugne, 2014].

- **le module Physique :** Le module physique gère quand à lui un état de l'agent permettant de respecter les contraintes d'enchaînement temporel liées au réalisme des animations visuelles et sonores ; par exemple un piéton assis doit d'abord se lever avant de partir à pieds, ce qui prend un certain temps pendant lequel il ne peut pas encore se déplacer.

- **Perception :** Ce module notifie les agents de ce qui se passe dans le monde simulé, à l'aide « d'informations perceptives » visuelles ou sonores. Pour cela le module observe le KDM à la recherche des effets spéciaux et de ce que font les agents, susceptibles d'être perçus par les autres agents. Seuls les agents qui peuvent raisonnablement percevoir un effet ou une condition en sont notifiés. Ceci évite aux agents d'avoir à rechercher eux même tout ce qui se passe autour d'eux, et permet d'optimiser la charge de calcul liée aux perceptions.

Un schéma de l'organisation des différents modules du projet est montré sur la figure

4.5. L'intégration et le développement des modules pour le projet Terra Dynamica est actuellement achevée et fonctionnelle.

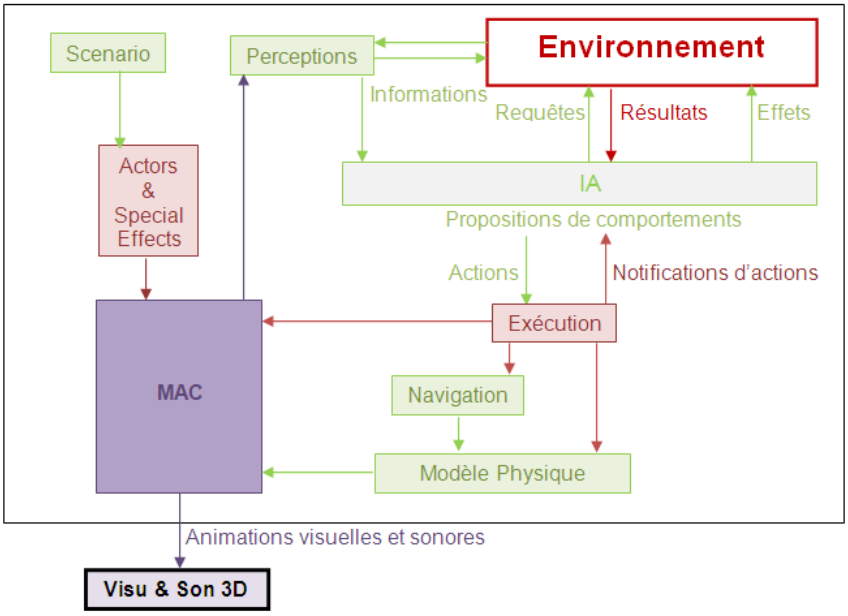


Figure 4.5 – Organisation générale des modules du projet Terra Dynamica

4.3.3 Contraintes liées au projet TD

Notre module de représentation sémantique de l'environnement a été conçu de façon à répondre aux contraintes du projet *Terra Dynamica*, qui sont :

- *la crédibilité des comportements :* en plus du fait que les agents doivent avoir leur buts propres, réagir aux événements selon leurs caractéristiques individuelles, anticiper et planifier leurs comportements, etc. Ils doivent être capables de s'adapter et de réagir aux changements qui opèrent dans leur environnement, avoir des comportement opportunistes et choisir les meilleurs objets qui leur permettent de réaliser leurs actions dans un contexte donné.
- *passage à l'échelle :* le projet *Terra Dynamica* vise la simulation d'un grand nombre d'agents, il doit donc être possibles de simuler des milliers d'agents au même moment. Cela signifie que le module sémantique doit répondre d'une manière rapide et efficace aux différentes demandes des autres modules de la simulation afin que les agents puissent répondre rapidement aux événements de leur environnement pour être crédibles .
- *définition d'une architecture générique :* l'architecture et la plate-forme utilisées doivent être suffisamment génériques pour convenir à plusieurs domaines d'application comme les jeux vidéo, la sécurité, les transports et l'urbanisme.

4.3.4 Interfaces de communication entre le module sémantique de l'environnement et les autres modules de l'architecture du projet

Le diagramme 4.6 présente les différents échanges entres le module sémantique de l'environnement et le reste de l'architecture d'agents de notre projet.

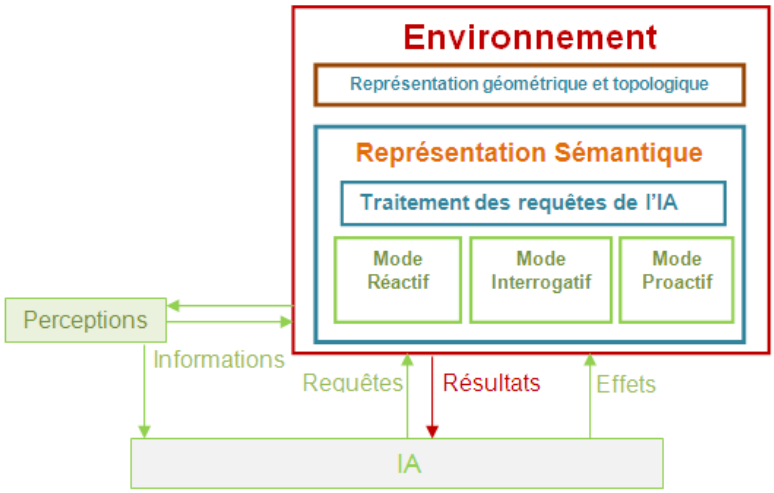


FIGURE 4.6 – Diagramme présentant les interactions du module sémantique de l'environnement avec les autres modules du projet

L'environnement est défini par deux types de représentation. D'une part, nous avons la représentation physique de l'environnement comportant les données topologiques et géométriques des objets. D'autre part, nous avons la représentation sémantique définie dans l'ontologie de services, qui est utilisée pour le traitement des requêtes du module d'IA. Ce traitement est appliqué pour répondre aux requêtes envoyées par le module d'IA selon les trois modes d'interaction.

Ces deux représentations de l'environnement sont reliées et intègrent les données reçus par les perceptions et les effets des actions des agents pour être inclues dans le traitement des requêtes.

Nous détaillerons dans les deux sections suivantes le traitement des données de perception et les échanges avec le module d'IA.

4.3.5 Perception de l'environnement

Les perceptions sont gérées au niveau de l'environnement par un système d'implémentation d'une interface d'abonnement. Les agents peuvent percevoir leur environnement et ce qui s'y passe en s'abonnant aux différents types de perception existantes, c'est à dire les différents sens humains (la vue, l'ouïe, l'odorat, etc).

Nous distinguons deux types de perceptions :

- **Perceptions Actives**, c'est à dire à l'initiative de l'agent (il s'abonne), qui par exemple cherche à localiser un distributeur bancaire pour retirer de l'argent ;
- **Perceptions Passives**, c'est à dire non initiées par l'agent, par exemple un bruit soudain de détonation.

La boucle (perception – action) des acteurs se déroule ainsi :

- le module de Perceptions récupère les données du KDM, et les traduit en informations perceptives à destination des agents gérés par les modules d'IA de haut niveau ; ceci prend en charge les perceptions « passives », afin que les agents puissent réagir à ce qui arrive dans la simulation ;
- les modules d'IA de haut niveau interrogent le module sémantique de l'environne-

ment afin d'acquérir de l'information sur l'environnement ; ceci prend en charge les perceptions « actives », impliquées dans la construction de comportements intentionnels ;
- les modules d'IA de haut niveau utilisent les informations perceptives et les résultats envoyés par le module sémantique de l'environnement pour choisir quelles actions exécuter ;
- Le module d'Exécution se charge d'exécuter les actions choisies, en s'appuyant sur les modules Physique et de Navigation avec deux types de conséquences :
  - mise à jour des données du KDM des agents (exécutant ou cible de l'action), qui seront par la suite prises en compte par le module de Perceptions ;
  - notifications de début et de fin d'action vers les modules de l'IA de haut niveau, qui aboutirons éventuellement à des notifications d'Effets vers le module Sémantique de l'environnement, afin d'appliquer les effets des actions sur la représentation sémantique de l'environnement.

L'interface qui permet d'implémenter les perceptions est IPerceptionAPI décrite dans le tableau 4.1. Les agents dotés de perceptions doivent implémenter l'interface IPerceptor décrite dans le tableau 4.2.

Méthodes	**Description**
int iGetPerceptionIndex(char *name)	résout un nom de perception en un indice positif ou nul, ou retourne 1 si le nom est inconnu
Char *iGetPerceptionName(int t)	récupère le nom de perception associé à un indice de perception, ou null si ce type est inconnu
Void iRegister(IPerceptor*)	enregistre une interface de perception associée à un agent, pour notifications ultérieures des perceptions
Void iUnregister(IPerceptor*)	dés-enregistre une interface de perception associée à un agent

Table 4.1 – Description des méthodes de IPerceptionAPI

Méthodes	**Description**
Actor *iGetActor()	retourne un pointeur sur l'agent qui perçoit
Void iNotify(int p, Sense s, SpecialEffect *e)	notification que *l'effet spécial* e génère une *perception* d'indice p par le *sens* de perception s
void iNotify(int p, Sense s, Actor *a)	notification que l'agent a génère une *perception* d'indice p par le sens de perception s

Table 4.2 – Description des méthodes de IPerceptor

Un agent enregistré à l'aide de la méthode iRegister sera notifié (à chaque cycle) de toutes les perceptions lui parvenant suite à la proximité d'un effet spécial ou d'un autre agent générateur de perceptions.

Les modules d'IA de haut niveau créent un objet Perceptor (dérivé de IPerceptor) associé à un agent, et l'enregistrent auprès du module de perceptions, qui récupère l'adresse de l'agent via iGetActor(). Ensuite, à chaque cycle, le module de perception surveille les agents et effets spéciaux présents dans le KDM, et notifie éventuellement l'objet Perceptor des perceptions qui le concernent, engendrées soit par un autre agent soit par un effet spécial ; l'objet Perceptor doit alors faire ce qu'il faut pour déclencher les comportements réactifs appropriés.

Enfin, les modules d'IA de haut niveau désenregistrent et détruisent l'objet Perceptor.

4.3.6 Communication avec les modules d'IA de haut niveau

Le module sémantique de l'environnement fournit les données sémantiques sur l'environnement aux modules de l'IA de haut niveau à travers l'API ISemanticWorldAPI. Les méthodes de cette API permettent de récupérer les informations sur les objets sémantiques de l'environnement pouvant fournir des services aux agents. Ces méthodes sont décrites dans le tableau 4.3. Nous pouvons voir comment se font les appels entre les deux modules dans le diagramme de la figure 4.7.

Méthodes	**Description**
int iGetServiceIndex(const char* servName)	elle retourne l'identifiant numérique unique d'un service sémantique à partir de son nom, ou bien elle retourne 1 si le nom du service est inconnu
char* iGetServiceName(int servID)	elle retourne le nom du service avec l'identifiant spécifié, ou bien elle retourne null si ce service est inconnu
bool iModeInterrogatif(actor, servList, coefQoS[], nbMaxRes, res)	implémente le mode Interrogatif de notre modèle, qui consiste à récupérer et trier selon la QoS les objets pouvant fournir le service demandé
bool iModeProactifRegister(SemObs *obs, int service)	elle permet d'enregistrer un observateur pour le mode Proactif de notre modèle
Void iModeProactifCancel(SemObs *obs, int service)	elle permet de désenregistrer un observateur pour le mode Proactif dans le cas où l'agent est supprimé ou bien il n'a plus besoin du service en mode proactif
Bool iModeReactif(actor, service, obj, flag, result)	elle permet d'implémenter le mode Réactif de notre modèle, qui consiste à récupérer l'information sur une instance d'objet connue

TABLE 4.3 – Description des méthodes de ISemanticWorldAPI

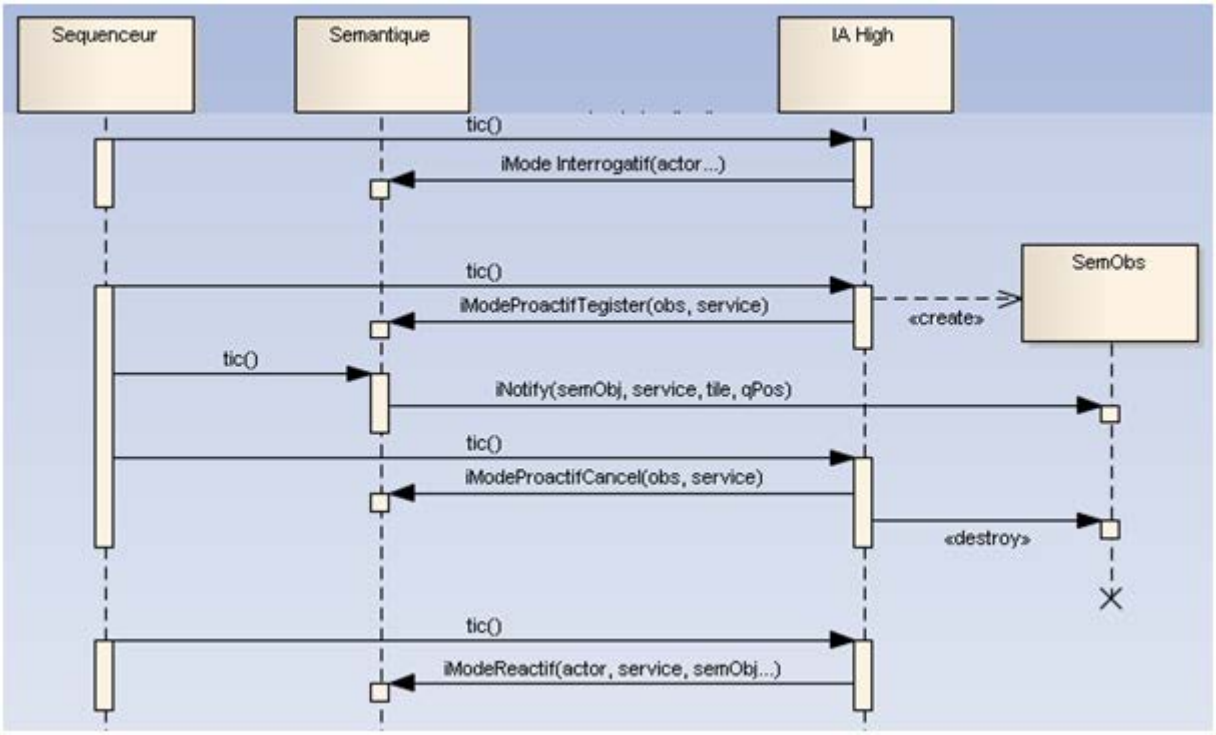


Figure 4.7 – Interactions entre le module sémantique et l'IA

4.3.7 Implémentation des trois modes d'interactions entre notre module et les modules d'IA de haut niveau

4.3.7.1 Le mode Interrogatif

Le principe est de fournir aux modules de l'IA de haut niveau la possibilité de localiser un ou plusieurs objets qui permettent de réaliser une action pour des fins de planification.

Cependant, le module sémantique de l'environnement ne doit pas fouiller l'environnement d'une manière exhaustive et ne doit pas tenir compte des disponibilités réelles des services d'objets, sous peine de rendre les agents *« extra-lucides »*.

L'appel au mode Interrogatif de la classe ISemanticWorldAPI se fait comme suit :

```
iModeInterrogatif(
Actor *actor,          // Agent cherchant un ou/des service(s)
const listOfService,   // Liste des services recherchés
const coefQoS,         // Coefficients de qualité de service entre $[0, 1]$
size_t x,              // Nombre max de résultats souhaités
RES &res               // Tableau de résultats
);
```

La description de iModeInterrogatif est donnée dans le tableau 4.4.

Lorsque le module d'IA invoque cette méthode, elle retourne false si il y a une erreur, sinon elle retourne true et les résultats sont placés dans le vecteur res fourni par l'appelant (module d'IA), triés par leur *QoS* décroissante.

Méthodes	Description
Actor *actor	la référence de l'agent KDM qui cherche le service, cela nous permet d'avoir accès à sa position et aux attributs liés à son type
ServList servList	vecteur de couples (service, importance) : – int service : indice de service lié à l'action à effectuer, – float importance : le module d'IA précise l'importance relative dans l'intervalle $]0, 1]$ du service recherché par rapport aux autres services recherchés si l'agent fait la demande de plusieurs services à la fois.
float coefQoS[]	donne des coefficients d'importance aux paramètres d'évaluation de la *QoS* (efficacité, proximité et difficulté). Par défaut tous les coefficients sont égaux = 1
int nbMaxRes	précise le nombre maximal d'instances d'objets à retourner dans le vecteur de résultats res
ObjSrvQoS_vect &res	la référence d'un vecteur de résultats retournés au module d'IA, chaque élément du vecteur est un triplet (obj, servElm, QoS) : – SemObject *obj : pointeur sur l'instance d'objet 3D permettant d'interagir avec lui et donnant accès : – au type sémantique concret de l'objet (DAB, poubelle, banc, etc), – à la position de l'origine de l'objet, – aux autres attributs de l'objet etc. – int servElm : indice du service élémentaire associé à l'objet. – float QoS : qualité de service calculée de l'objet (valeur entre 0 et 1).

TABLE 4.4 – Description de la méthode iModelnterrogatif

Notons qu'une même instance d'objet peut figurer plusieurs fois dans les résultats, associé à des services et *QoS* différentes. La *QoS* de chaque objet pour un service est calculée comme le produit de la somme pondérée par coefQoS[] des valeurs des paramètres de calcul de la *QoS* (efficacité, proximité et difficulté) de l'objet pour le service, par la somme des importances des services demandés aboutissant au service élémentaire. Ceci permet de mieux noter un service satisfaisant tous les services demandés, que seulement certains d'entre eux.

Si servList ne comprend qu'une paire (service, importance), au plus nbMax résultats sont retournés dans res.

Si servList comprend plusieurs paires (service, importance), plus de nbMax résultats peuvent être retournés dans res. En effet dans ce mode le module sémantique de l'environnement doit s'assurer que chaque service demandé est couvert par au moins une instance d'objet retourné. Si ce n'est pas le cas dans les au plus nbMaxRes premiers objets trouvés, le module sémantique de l'environnement doit ajouter, si possible, un objet par service non couvert.

Le module sémantique de l'environnement tient compte du temps de parcours estimé pour atteindre l'objet, et de la durée typique d'utilisation du service, sans tenir compte de la disponibilité ou accessibilité réelle de l'objet.

Si l'appel de la méthode iModeInterrogatif retourne true mais que le vecteur résultat res est vide, c'est qu'il n'y a pas d'objet dans la BdD permettant d'accomplir le ou les services spécifiés, ce qui ne devrait jamais se produire en pratique.

Afin de limiter la consommation CPU, le module sémantique de l'environnement utilise une technique de *géo-hachage* qui permet de limiter la zone de recherche en cherchant les objets par zones concentriques dans les cellules autour de l'agent, en n'allant dans une couronne périphérique que si les couronnes internes n'ont pas encore fourni le nombre d'objets demandé.

4.3.7.2 Le mode Proactif

Le mode proactif permet au module de l'IA de déclarer qu'un agent cherche à localiser de manière opportuniste un objet fournissant un service donné, et d'être notifié chaque fois que l'agent passe à proximité d'un tel objet. Le mode proactif peut être invoqué pour deux raisons. Soit l'agent ne connaît aucun lieu lui permettant de réaliser l'action désirée. Soit l'action que l'agent veut réaliser est très peu prioritaire, et les lieux qui lui permettent de la faire sont mal placés par rapport aux autres lieux dans lesquels il souhaite se rendre.

Pour cela, nous avons utilisé une interface d'observateur nommée SemObs dotée des méthodes iGetActor et iNotify telles qu'elles sont décrites dans le tableau 4.5.

Les méthodes qui permettent l'enregistrement et le dés-enregistrement d'un agent en mode proactif sont décrites dans le tableau 4.6.

Le module sémantique de l'environnement ne doit pas proposer sans cesse le même objet tant que l'agent reste à proximité. Quand un objet a été proposé une fois, il ne peut être reproposé que s'il s'en est d'abord éloigné de plus de la distance rMax de l'objet déjà proposé, ceci afin de ne pas proposer sans cesse la même instance d'objet à chaque cycle. Enfin, l'agent reste enregistré dans le mode proactif tant qu'il ne se dés-enregistre pas

Méthodes	Description
Actor *SemObs : :iGetActor()	cette méthode retourne l'agent KDM associé à l'observateur
Void SemObs : :iNotify(SemObject *obj, int servElm, float time, int qPos)	cette méthode notifie le module d'IA qu'une instance d'objet fournissant le service recherché a été trouvé , avec : – obj : est le pointeur sur l'instance de l'objet, – servElm : est l'indice du service associé à obj, – time : est le temps d'accès estimé pour atteindre l'objet pour pouvoir accomplir l'action, – qPos : est la position relative dans la file d'attente (0 = en tête) si elle existe.

TABLE 4.5 – Description des méthodes de l'interface SemObs

Méthodes	Description
bool iModeProactifRegister(SemObs *obs, int service)	l'appel de cette méthode retourne true si l'enregistrement est effectif, elle retourne false sinon. Un objet n'est proposé que lorsque l'agent enregistré pour le service désiré est à moins d'une distance rMax de cet objet. rMax étant un paramètre configurable que nous avons fixé
Void iModeProactifCancel(SemObs *obs, int service)	cette méthode permet le dés-enregistrement explicite d'un agent, ce qui engendre une réinitialisation des couples (acteur, objet) déjà proposés. Lorsqu'un agent est détruit, le module sémantique fait le ménage et le dés-enregistre automatiquement pour toutes ses demandes précédentes

TABLE 4.6 – Description des méthodes d'enregistrement et de dés-enregistrement d'un agent en mode Proactif

explicitement, ou qu'il n'est pas détruit.

Le mode proactif se fait en mode synchrone, où le traitement de ce type de requête est réalisé comme une tâche de fond tant que l'agent est abonné à un service. Notre choix d'utiliser un observateur qui sera exécuté en parallèle avec d'autres tâches, combiné à un système de notification décharge ainsi le module de décision d'envoyer à chaque fois une requête pour le même service.

4.3.7.3 Le mode Réactif

Au cours de la simulation, il se peut que l'état de l'instance d'objet qui a été sélectionnée pour réaliser l'action de l'agent soit modifiée par un événement ou qu'il y ait une file d'attente qui s'est formée. Le module de décision de l'agent va donc ré-interroger le module sémantique de l'environnement pour savoir si l'instance d'objet qu'il a sélectionné dispose toujours de la *QoS* qui lui a été attribuée pendant la planification, ou bien s'il y a une meilleure alternative qui se présente à l'agent au moment où il s'apprête à réaliser son action.

Le module de décision de l'agent va donc invoquer la méthode du mode Réactif du module sémantique de l'environnement iModeReactif de la manière suivante :

```
bool iModeReactif(
Actor *actor,   // Agent cherchant le service
int service,    // Indice du service recherché
SemObject *obj, // Pointeur sur l'instance d'objet qui sera évalué sur le
                  service recherché
bool flag,      // Flag d'autorisation ou/pas de proposer un autre objet
ModeRéactifRes \&res  // Tableau de résultats
);
```

L'appel de cette méthode retourne true s'il y a un résultat valide, et false sinon. Si flag est positionné à true par l'appelant, le module sémantique de l'environnement peut proposer un autre objet rendant le même service au cas où l'objet de départ est indisponible, ou si le nouvel objet est susceptible d'être plus rapidement accessible que l'ancien déjà sélectionné. Dans ce cas objBis est un pointeur non null, et les champs timeBis et qPosBis sont renseignés. L'hypothèse de base est que le nouvel objet est très proche, et que donc l'agent est raisonnablement capable d'en connaître la disponibilité et l'accessibilité réelle du nouvel objet proposé.

4.4 Fonctionnement du module sémantique de l'environnement

4.4.1 Description de l'environnement

Les lieux et les objets ne sont intéressants pour les agents que dans la mesure où ils peuvent interagir avec. Il faut donc décrire ce qui peut être accompli avec des lieux et objets, et comment. Pour cela, nous considérons que chaque lieu ou objet peut offrir aux agents un ou plusieurs services, avec des qualités de services *QoS* variables.

Un service permet à un agent d'accomplir une action dans le but de satisfaire un besoin ou d'obtenir un résultat. Le diagramme UML (figure 4.8) illustre comment sont représentés les services d'objet dans notre module.

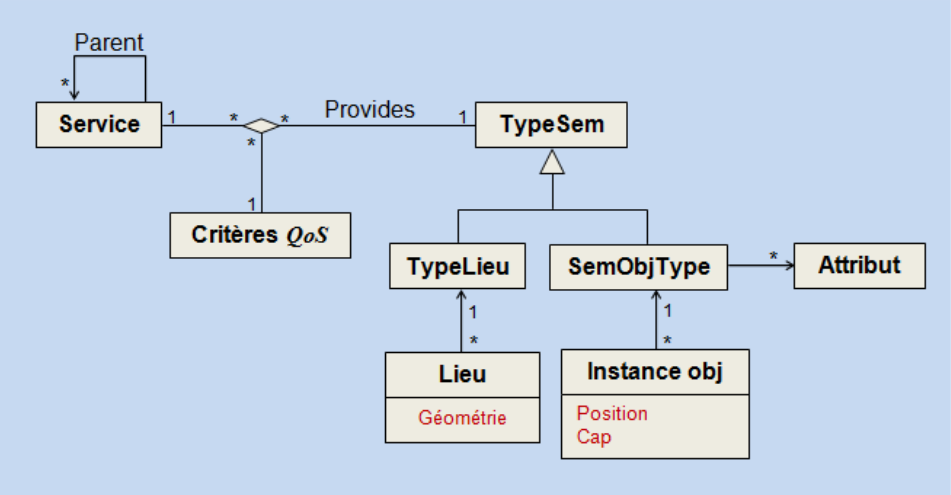


FIGURE 4.8 – Représentation des objets et lieux de l'environnement

Un service peut avoir un service parent de nature plus générale, représenté par le lien *« parent»* dans le diagramme 4.8. On appelle service élémentaire (service d'action) un service qui n'est parent d'aucun autre service.

Chaque service est associé à une action / comportement définis dans le module d'IA de haut niveau. Une action élémentaire sera associée à un service d'action, et un comportement à un service de comportement *« parent»*. Par contre les liens de parenté entre les services ne seront pas forcément les mêmes avec les liens de parenté entre les actions et les comportements définis par l'IA.

- Un TypeSem (type sémantique) désigne un type de lieu (TypeLieu) ou un type d'objet (SemObjType).
- La relation provides permet d'associer un service à un type d'objet. Plusieurs services distincts peuvent être associés au même TypeSem, et vise versa, un service donné peut être associé à des TypeSem différents.
- Les Critères QoS définissent les critères sur lesquels sera évalué la qualité d'un service fourni par un type d'objet donné (par exemple, le coût, le temps, le confort, etc).

4.4.1.1 Description d'un Objet sémantique

Une instance d'un type d'objet 3D est représentée par la classe SemObject. Elle permet aux agents de se diriger vers l'objet et éventuellement de faire la queue, d'utiliser l'objet et/ou dans certains cas de modifier son état.

Le diagramme de classe ci-dessous (figure 4.9) décrit la classe SemObject et les classes attenantes.

Chaque instance de SemObject pointe sur une instance de SemObjType (_type), qui

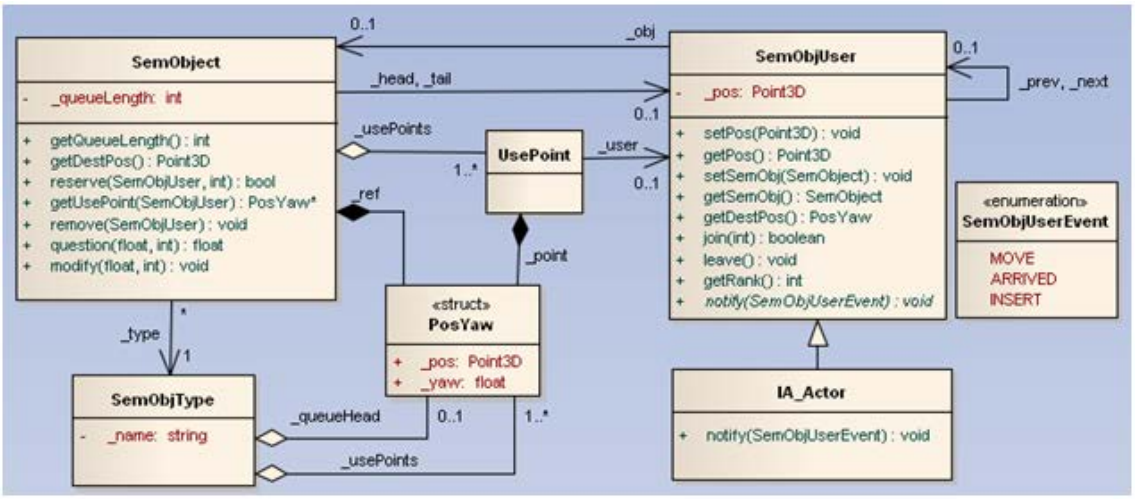


Figure 4.9 – Le diagramme de classes UML de la description d'un objet sémantique

représente le type d'objet sémantique associé : le type possède la liste des points d'utilisation de l'objet (_usePoints), et la position/orientation de la tête de queue optionnelle (_queueHead), le tout dans le repère de l'objet.

La classe SemObject possède la position/orientation de l'objet (_ref), ainsi que la table des points d'utilisations (_usePoints).

La classe SemObjUser représente un agent utilisateur de l'objet, et lui permet de faire la queue.

La classe IA_Actor permet la représentation d'un agent pour les modules de l'IA de haut niveau. Chaque instance de IA_Actor a une référence sur la classe Actor du KDM, et fédère toutes les données propres à cet agent.

IA_Actor dérive de SemObjUser, et doit implémenter sa méthode notify qui l'avertit de l'évolution de la queue avec les évènements MOVE (déplacement du prédécesseur), ARRIVED (arrivée à un point d'utilisation), et INSERT (un autre agent s'insère dans la queue juste devant l'agent courant).

Les classes SemObject et SemObjUser sont décrites respectivement dans les tableaux 4.7 et 4.8.

4.4.2 Interface de communication avec l'ontologie de services

Nous avons décrit dans notre ontologie tous les types d'objets qui existent dans l'environnement ainsi que les services de comportement et d'action qu'ils offrent aux agents et les relations qui peuvent exister entre eux (comme cela a été défini dans le chapitre 3.

Aucune instance d'objet ou donnée dynamique sur l'environnement n'est représentée dans notre ontologie, seuls les informations sémantiques et les données nous permettant de calculer la *QoS* d'un service proposé par un type d'objet sont représentés dans notre

Méthodes	Description
int getQueueLength()	retourne le rang qu'aurait un nouvel utilisateur dans la file
Point3D getDestPos()	retourne la position dans la queue
bool reserve(SemObjUser* u, int rank=0)	réserver un point d'utilisation de l'objet pour l'agent spécifié
void remove(SemObjUser *u)	retire l'agent u du SemObject
PosYaw* getUsePoint(SemObjUser *u)	si un point d'utilisation est réservé pour SemObjUser, elle retourne un pointeur sur le PosYaw correspondant, sinon elle retourne null.

TABLE 4.7 – Description des principales méthodes de SemObject

Méthodes	Description
void setSemObj(SemObject*)	désigne l'objet sémantique utilisé
SemObject* getSemObj()	récupère l'objet sémantique courant
void setPos(Point3D)	initialise la position courante d'attente dans la queue
Point3D getPos()	retourne la position courante d'attente dans la queue
PosYaw getDestPos()	retourne la position de destination courante
bool leave()	détache l'agent de l'objet courant avec getSemObj()->remove(this)
int getRank()	retourne le rang dans l'objet, ou 1 si l'agent n'est pas lié à un objet ou n'a pas encore de rang
void notify(event)	cette méthode est appelée par SemObject quand il se passe quelque chose dans la queue qui concerne l'agent courant

TABLE 4.8 – Description des principales méthodes de SemObjUsert

ontologie.

La sélection des types d'objets à retourner aux agents se fait d'abord au niveau de l'ontologie. Lorsqu'un module d'IA de haut niveau envoie une demande au module sémantique celui-ci va interpréter cette demande et va la traduire en une requête (composées de plusieurs sous-requêtes) qui permet d'interroger l'ontologie.

La recherche des objets en réponses aux demandes des modules d'IA de haut niveau se fait de la manière suivante :

Au niveau de l'ontologie

- si la requête concerne un service de comportement nous allons d'abord chercher tous les services d'action qui l'implémentent,
- pour chaque service d'action demandé nous allons récupérer tous les types d'objets qui le proposent,
- pour chaque couple trouvé *(service, type d'objet)* , nous allons calculer une *QoS* (selon les paramètres *Efficacité, Difficulté, Proximité*),
- nous pourrons alors trier les couples *(service, type d'objet)* trouvés selon leurs *QoS* calculée,

Au niveau de l'environnement
- chercher dans l'environnement dans un périmètre p autour de l'agent les instances d'objets qui correspondent à chaque type d'objet trouvé,
- enfin, inclure les paramètres de calcul *contextuels* (la proximité réelle, l'état de l'objet, la taille de la file d'attente, etc) pour trier les instances d'objets trouvées entre elles.

4.4.3 Implémentation du module sémantique

Lors de la phase de prototypage nous avons recensé plusieurs problèmes liés notamment à l'implémentation et l'interaction avec l'ontologie de services. Voici les principaux problèmes identifiés :

- la disponibilité limitée des outils du web sémantique en langage de programmation *C++*,
- l'occupation de la représentation de l'ontologie d'un très grand volume d'espace mémoire,
- les très faibles performances d'analyse et d'interrogation de l'ontologie.

Ces problèmes sont apparus après avoir implémenté notre prototype en utilisant la librairie Java Jena [1]. Nous avons donc choisi d'utiliser *owl-cpp* qui est une librairie *C++* qui permet l'analyse, l'interrogation et le raisonnement sur une ontologie en *OWL*. Pour cela, *owl-cpp* utilise *Raptor*, *Fact++* et les librairies *Boost* [2]. Notre choix était motivé par le fait qu'elle est implémentée en langage *C++* standard ce qui permet son utilisation avec plusieurs plateformes notamment celle de notre projet, mais aussi elle permet de faire des calculs rapides grâce à sa représentation compacte en mémoire.

Les principaux points forts de la librairie *owl-cpp* qui nous intéressent sont les suivants :

- l'analyse et l'import des fichiers de l'ontologie sous le format RDF/XML,
- la représentation compacte de l'ontologie et l'occupation d'un espace mémoire réduit,
- la conversion des triplets en axiomes et leur chargement dans le raisonneur,
- l'exécution des requêtes de la logique de description avec un temps de réponse rapide.

La librairie *owl-cpp* est composée de trois modules :

- **le module d'analyse :** qui utilise l'unique librairie *C* d'analyse *RDF* de *Raptor* [3]. Afin d'analyser les fichiers XML, *Raptor* utilise l'interface *SAX* de la librairie *Libxml2* [4]. *Owl-cpp* charge les ontologies uniquement à partir des flux *STL* et des emplacements des fichiers systèmes spécifiés par l'utilisateur.

- **le module de stockage de triplet *RDF* :** qui permet de stocker, de rechercher et de récupérer les triplets *RDF*. Il est possible de procéder de différentes manières pour retrouver et récupérer un triplet ou un ensemble de triplet *RDF*. En fonction de l'ID du nœud spécifié il est possible de procéder de huit façons : juste avec le

1. https ://jena.apache.org/
2. http ://www.boost.org/
3. http ://librdf.org/raptor/
4. http ://xmlsoft.org/

sujet, le prédicat ou l'objet ou bien les différentes combinaisons de ces derniers, qui comprend la configuration où aucun ID n'est spécifié, dans ce cas ça retourne tous les triplets de l'ontologie. Par ailleurs, il existe trois types de résultats : la liste complète des triplets qui correspond aux IDs spécifiés dans la recherche ; le premier triplet trouvé ; ou bien un booléen indiquant si la recherche était réussie ou pas.

– **le module de raisonnement :** qui utilise le raisonneur *Fact++* [Tsarkov and Horrocks, 2006] qui est la seule librairie open-source en *C/C++* qui fait du raisonnement sur la logique de description. *Owl-cpp* envoie les informations au raisonneur *Fact++* de l'endroit de stockage des triplet appelé *"store"* en transformant les triplets en axiomes. Cette conversion est faite avec *Visitor Deseign Pattern* [Gamma et al., 1995]. Les requêtes de logique de description sont faites directement à travers une interface avec *Fact++*.

Les interface de *owl-cpp* sont très simplifiées afin de réaliser les opérations basiques sur les ontologies. Par exemple, pour charger le fichier *"onto.owl"* de l'ontologie dans le *store* des triplets en appelant load("onto.xml, store). Ce n'est qu'une fois chargée dans le *store* que nous pouvons faire des requêtes sur notre ontologie. Par exemple, l'expression find_triples(blank, T_rdf_type : :id(), T owl Class : :id(), store), retourne tous les triplets qui déclarent une classe. Les axiomes sont copiés du *store* de triplets au *Kernel* du raisonneur *Fact++* en appelant add(store, kernel).

4.5 Conclusion sur la mise en oeuvre

Nous avons fait des choix d'implémentation qui ont contribué dans la réduction du temps de traitement des requêtes. A savoir, l'utilisation de la librairie *C++ owl-cpp* qui occupe un espace réduit en mémoire et exécute les requêtes avec un temps de réponse rapide par rapport aux autres librairies qui sont plus couramment utilisées (Jena).

Le choix d'utiliser un système d'observation et de réponse par des méthodes de notification nous a permis de réduire le temps de réponse au requêtes du module d'IA.

A l'heure actuelle, le module sémantique de l'environnement a été intégré dans la plateforme du projet Terra Dynamica et il est fonctionnel. Cependant, le logiciel développé dans le projet est utilisé comme démonstrateur, mais il est prévu d'intégrer toutes les technologies développées dans le cadre du projet dans de réelles applications. L'application du modèle à d'autres applications que celles visées dans le projet Terra Dynamica permettrait de tester sa généricité.

Dans la suite de ce chapitre, nous allons nous intéresser à l'évaluation objective de notre modèle. Nous évaluerons notamment le passage à l'échelle en faisant varier le nombre d'agent dans la simulation, la taille de l'ontologie de représentation de l'environnement et l'activation/désactivation des différents modes de communication entre les agents et leur environnement.

Évaluation Objective

4.6 Passage à l'échelle

Nous avons intégré notre module dans la plateforme du projet Terra Dynamica, celle-ci nous a permis de faire l'évaluation subjective de notre modèle comme décrite dans le chapitre 5.

Les principaux problèmes techniques liés à un tel système de simulation est le temps de réponse pour un très grand nombre d'agents. En effet, nous avons constaté à travers notre prototype que le temps de réponse augmentait avec le nombre de requêtes envoyées au module sémantique, qui sont en relation avec le nombre d'agents dans la simulation.

Parmi les objectifs du projet Terra Dynamica, nous citons **le passage à l'échelle** pour un très grand nombre d'agents dans la simulation. Afin de vérifier cet objectif nous avons estimé le temps de réponse du module sémantique selon trois paramètres :

- le nombre d'agents dans la simulation (proportionnellement au nombre de requêtes),
- la taille de l'ontologie (le nombre de concepts représentés comportant différents types d'objets et différents services),
- et le type d'interaction possible que peuvent avoir les agents et leur environnement (sollicitation intensive du module de l'environnement).

Les simulations ont été réalisées sur un ordinateur Windows XP Professional 64 bits, d'une mémoire vive de 8 Go, muni d'un processeur Intel(R) Xenon(R) CPU 3.20GHz.

4.6.1 Performances selon le nombre d'agents

Nous avons mesuré le temps de réponse aux requêtes envoyées par les modules d'IA de haut niveau au module sémantique de l'environnement pour un grand nombre d'agents.

Pour cela, nous avons augmenté la fréquence de création des agents dans notre environnement de simulation. Les agents créés sont des "agents réactifs standards" [Reynaud, 2014], c'est-à-dire ils sont dotés de l'ensemble des modules de haut-niveau de base (module de motivation, d'emploi du temps et d'instinct). Ils ont donc des buts ce qui génère automatiquement des requêtes vers le module sémantique.

Nous avons généré jusqu'à **3000** agents dans notre simulation, ce qui représente au moins le même nombre de requêtes (en mode Interrogatif) envoyées au module sémantique au moment de la création des agents.

Nous pouvons voir sur la figure 4.10 que le temps de réponse moyen pour une requête avec moins de **500** agent dans la simulation était inférieur à **0.2 ms**, celui-ci augmente d'une manière **linéaire** avec le nombre d'agents.

Le temps moyen de réponse maximal était atteint pour **3000** agents et qui était à moins de **0.3 ms** comme nous pouvons le voir sur la figure 4.10.

Nous remarquons qu'il n'y a pas une très grande différence entre le temps de réponse pour moins de **100** agents et **3000** agents, cela est du au fait que tous les agents n'envoient pas leurs requêtes au même moment vu qu'ils sont générés au fur et à mesure, en plus, lorsque les agents sont en train d'exécuter leur plan ils n'envoient pas de requêtes au module sémantique.

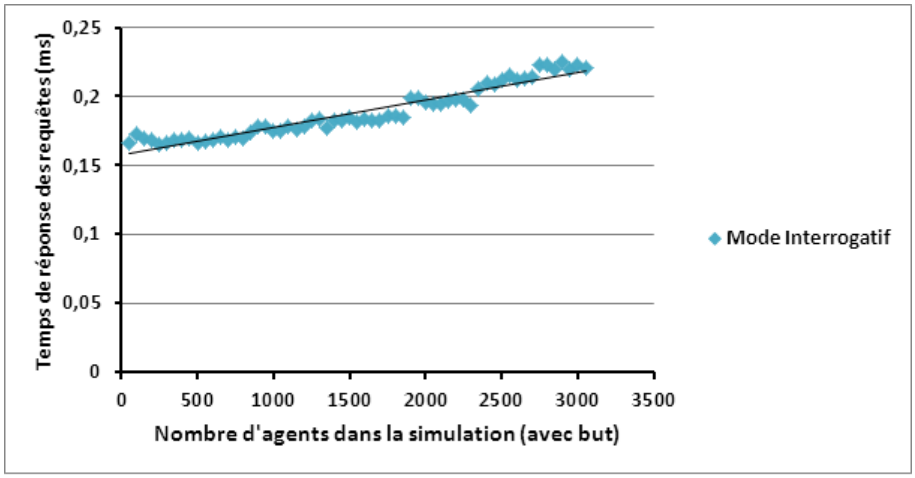


FIGURE 4.10 – Temps de réponse moyens aux requêtes du module d'IA par nombre d'agents

4.6.2 Performances selon le nombre de concepts dans l'ontologie

Cette fois-ci, nous avons augmenté le nombre de concepts dans l'ontologie de services en rajoutant des services jusqu'à atteindre **10 000** concepts représentant des services.

Nous constatons que le temps de réponse moyen lorsque nous avons moins de **500** agents dans la simulation est aux environs de **0.15 ms** et augmente d'une manière **linéaire** et **proportionnelle** avec le nombre d'agents dans la simulation et ne dépasse pas **0.25 ms** pour 2000 agents, comme nous pouvons le voir sur la figure 4.11.

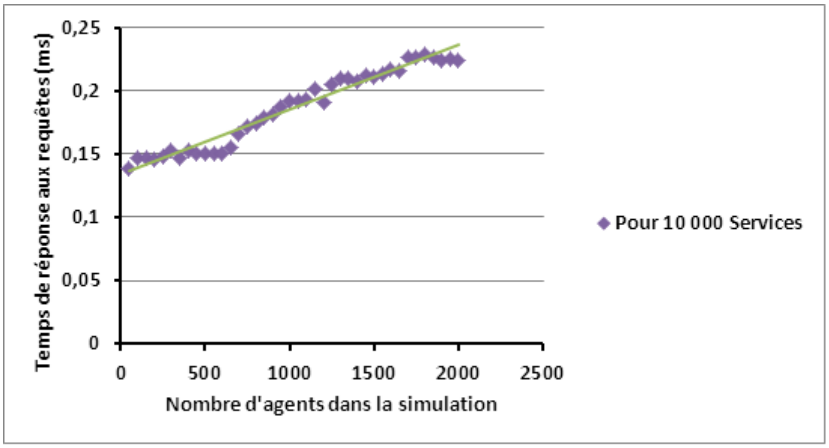


FIGURE 4.11 – Temps de réponse moyen pour 10 000 services et 2 500 agents

Cependant, en comparant entre les deux courbes, nous constatons que le temps moyen de réponse avec **10 000** services dans l'ontologie est plus élevé que celui avec la taille initiale de notre ontologie de services, mais cela reste **peu significatif**.

4.6.3 Performances selon le type d'interaction possible avec l'environnement

Les modes d'interactions réactif et pro-actif que nous avons définis pour interagir avec l'environnement virtuel nécessitent un traitement synchrone. De ce fait, les réponses du module sémantique aux requêtes envoyées dans ces deux modes peuvent être plus coûteuses en temps de traitement par rapport aux requêtes envoyées en mode interrogatif.

Nous allons mesurer l'impact que peut avoir l'activation de chacun de ces modes sur la simulation. Pour cela nous allons procéder en mesurant le temps de traitement des requêtes pour chacun des deux modes.

4.6.3.1 Avec le mode Réactif

L'activation du mode réactif permet au module de décision des agents d'émettre des requêtes pour avoir la *QoS* du service proposé par l'objet que l'agent s'apprête à utiliser. On compare cette *QoS* avec celle qui a été calculée pour le même service lors de la phase de planification. Dans le cas où cette *QoS* est inférieure à la *QoS* planifiée, le module sémantique va proposer un objet qui propose une *QoS* supérieure. Ce qui va permettre au module de décision de choisir entre les deux objets proposés et de choisir le meilleur.

La figure 4.12 montre les résultats que nous avons obtenus sur la mesure du temps de réponse des requêtes en mode réactif. Nous remarquons que le temps de réponse est variable indépendamment du nombre d'agents dans la simulation. La plupart des requêtes sont satisfaites avec un temps de réponse inférieur à **0.02 ms**. Le temps de réponse maximal atteint moins de **0,15 ms**.

Cet écart est dû aux nombre de requêtes envoyées simultanément. En effet, il se peut que le module sémantique ait à répondre à plusieurs requêtes à la fois. Nous allons donc observer un temps de latence qui va se cumuler au temps de réponse d'une requête puisque les requêtes sont traitées par ordre d'arrivée.

Le module sémantique répond en un temps très raisonnable aux requêtes en mode réactif. De ce fait, l'activation de ce mode ne ralentit pas la simulation.

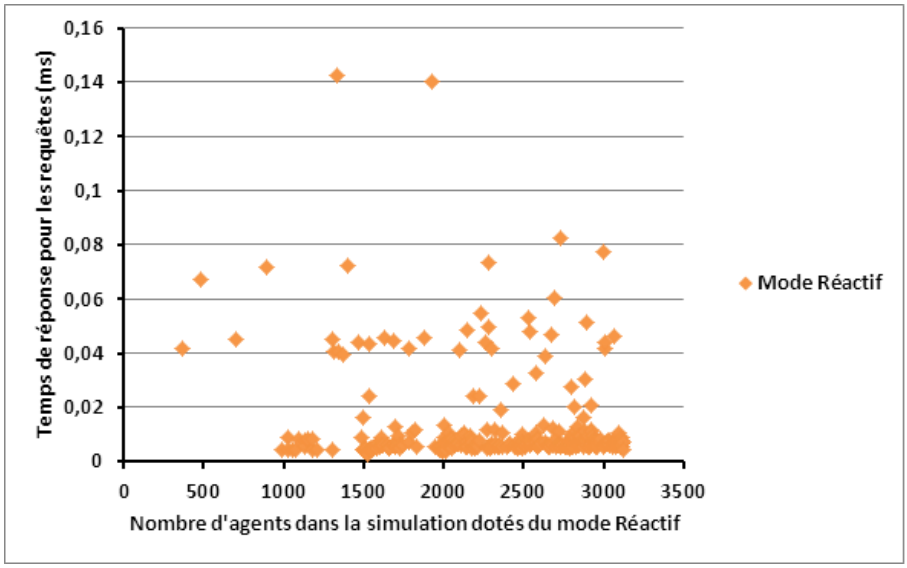


Figure 4.12 – Temps de réponse moyen pour les requêtes envoyées en *Mode Réactif*

4.6.3.2 Avec le mode Pro-actif

Le mode pro-actif permet d'observer des comportements opportunistes dans la simulation. Ces comportements concernent des actions peu prioritaires qui peuvent être réalisées à n'importe quel moment au cours de l'exécution du plan d'un agent, car elles sont indépendantes des autres actions du plan (comme par exemple jeter un objet en allant au travail).

Nous pouvons voir dans la figure 4.13 que le temps de réponse aux requêtes du mode pro-actif augmente d'une manière **exponentielle** avec le nombre d'agents dans la simulation. Cette augmentation se fait en deux phases. Nous pouvons remarquer que le temps de réponse ne dépasse pas **0.01 ms** pour **1000** agents dans la simulation. Puis nous remarquons que le temps de réponse augmente en suivant une exponentielle jusqu'à atteindre presque **7 ms** avec **3000** agents dans la simulation.

Cette augmentation disproportionnée est dû au fait que le traitement d'une requête pro-active nécessite une recherche active permettant ainsi d'envoyer des notifications aux agents dès qu'ils s'approchent d'un objet intéressant. Ce traitement est couteux ce qui explique que le temps de réponse augmente d'une manière exponentielle lorsque le nombre d'agents dans la simulation augmente.

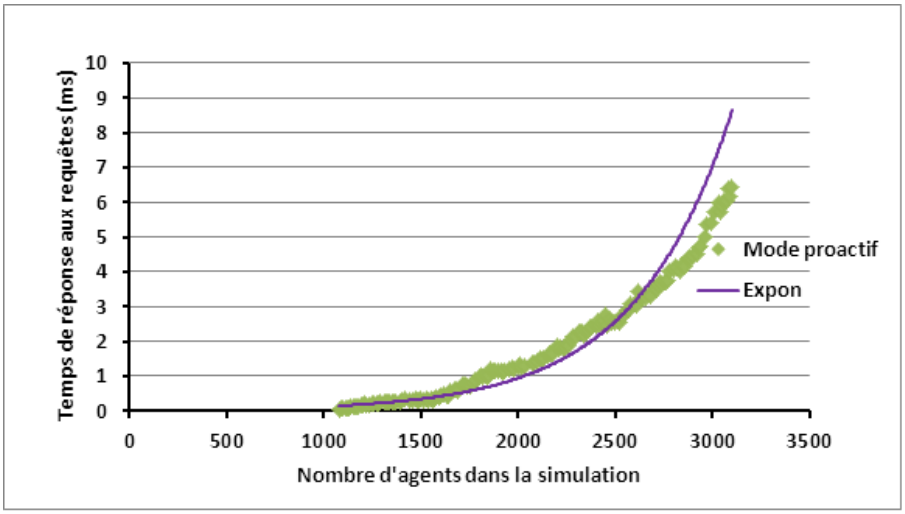


FIGURE 4.13 – Temps de réponse moyen pour les requêtes envoyées en *Mode Pro-actif*

4.6.4 Capacité de la simulation avant ralentissement

Nous nous somme intéressés par la suite au nombre d'agents simulés à partir duquel la simulation ralentit. Nous avons utilisé des cycles de **50 ms**, les résultats sont présentés dans le tableau 4.9 ci-dessous :

Pour un cycle de 50 ms nous avons **760** agents sans but (qui ne sont pas dotés de module décisionnel, donc qui n'envoient pas de requêtes au module de l'environnement) qui peuvent être simulés. Au delà de ce nombre, la simulation montre des signes de ralentissement. Les 760 agents ne font que traverser l'environnement pour aller du point source au point puis. Ce ralentissement est donc causé en grande partie par le module de

Type d'agents	Nombre d'agents maximal avant ralentissement
Agents sans but	760
Agents standards (avec but)	670

Table 4.9 – Tableau représentant le nombre d'agents maximal avant ralentissement par frame de 50 ms

navigation dont le temps de traitement dépasse les **40 ms** par frame à lui seul.

En dotant les agents de la simulation d'un module de décision qui leur alloue un but, le nombre d'agents simulé avant ralentissement baisse à **670** agents. Sachant que le processus décisionnel a besoin de 3,55 ms en moyenne comme temps de traitement pour les 670 agents par frame [Reynaud et al., 2014]. Pour le même nombre d'agents dans la simulation notre module à besoin de moins de **0,2 ms** (voir figure 4.10), ce n'est donc pas le temps de traitement des requêtes qui ralentissent la simulation.

Notre simulateur peut contenir environ **700** agents capables de réaliser des comportements crédibles avec un trafic routier d'environ **1300** véhicules avant que la simulation ne ralentisse. Cela reste tout à fait raisonnable pour un petit quartier de Paris, mais cela reste loin de l'objectif de simuler une grande ville. Notons que le module de l'environnement peut aller au delà des 700 agents simulés sans que cela n'affecte considérablement le temps de traitement des requêtes. En effet, notre modèle est capable de supporter jusqu'à **3000** agents avec un temps de réponse inférieur à **0.3 ms** (en mode interrogatif). Cette capacité de supporter un tel nombre d'agent est due au fait que les agents envoient leurs requêtes d'une manières différées les uns des autres car ils sont générés les uns après les autres. Une autre raison est que le temps de traitement d'une action représente un intervalle de temps où l'agent est occupé à réaliser son action et donc il n'envoie pas de requête vers le module sémantique. Même en dotant les agents des deux autres mode d'interaction le temps maximal de réponse du module sémantique ne dépassera pas les **7 ms** (en mode pro-actif) pour **3000** agents.

D'un autre coté, le fait d'augmenter la taille de l'environnement n'a pas un impact considérable sur le temps de traitement des requêtes du module de l'environnement. En effet, nous avons constaté que l'augmentation du nombre de services jusqu'à **10 000** ne ralentit pas le temps de traitement du module de l'environnement et la tendance linéaire de ralentissement est analogue à celle du nombre de services de départ (voir figure 4.11).

4.7 Conclusion sur l'évaluation des performances

Dans ce chapitre, nous avons décrit l'implémentation du prototype de notre modèle de représentation sémantique de l'environnement, et comment nous avons intégré notre modèle dans la plateforme du projet. L'implémentation de notre prototype nous a permis de mettre en évidence les problèmes technique que nous avons pu rencontrer lors de l'implémentation de notre modèle, mais aussi, il nous a permis de recenser les informations relatives à l'environnement que nous pouvons représenter dans notre modèle et celles qui sont utiles pour les agents afin d'avoir des comportements crédibles.

Nous avons aussi présenté l'architecture du projet Terra Dynamica ainsi que les différents modules qui la composent et nous avons décrit comment nous avons intégré notre modèle dans celle-ci. Cette architecture décrit le module sémantique comme un module se

situant au cœur de la simulation en interagissant avec tous les modules du projet.

Et enfin, nous avons fait une étude de passage à l'échelle de notre modèle en jouant sur deux paramètres (le nombre d'agents dans la simulation et le nombre de concepts dans l'ontologie de services). Celle-ci montre que notre modèle est tout à fait capable de gérer les demandes d'un grand nombre d'agents à la fois pour un nombre élevé de services et de types d'objets dans l'environnement de simulation.

Chapitre 5

Évaluation Subjective

5.1 Objectif

Dans ce chapitre, nous présentons l'évaluation de notre modèle (évaluation subjective) afin de tester sa capacité à aider les agents à avoir des interactions crédibles avec leur environnement grâce à la qualité de service qui leur est proposée. Celle-ci est destinée à aider les agents pour faire le meilleur choix d'objet parmi ceux se trouvant dans l'environnement. L'objectif de notre travail est de fournir au module décisionnel [De Sevin et al., 2012] des éléments d'aide à la décision des agents rendant leur choix d'objet cohérent avec l'action qu'ils veulent réaliser.

Pour vérifier la validité de notre approche, il est nécessaire de s'assurer que la *QoS* retournée aux agents leur permet de choisir **le meilleur élément** de l'environnement (au sens de la crédibilité comportementale) capable de satisfaire leur action selon un contexte donné.

L'environnement virtuel urbain possède une particularité qui est d'avoir des objets hétérogènes pouvant satisfaire la même action avec des qualités différentes et selon un contexte donné (par exemple, s'asseoir sur un banc public ou une chaise dans un café. Ces deux différents objets offrent le même service mais avec une qualité différente liée au prix, l'ambiance, le confort, etc). Nous voulons donc vérifier si les observateurs humains sont d'accord avec les propositions de l'environnement par rapport à une action dans un certain contexte tout en les orientant le moins possible dans leur jugement.

Nous proposons un protocole expérimental destiné à cette tâche d'évaluation subjective, que nous avons appliqué à notre modèle de représentation. Ce protocole doit évaluer la crédibilité comportementale des agents engendrée par les propositions de l'environnement qui utilise la qualité de service pour trier les objets possibles permettant de réaliser une action. A notre connaissance, aucun protocole n'a été proposé pour l'évaluation de la crédibilité des comportements des agents découlant d'un modèle de représentation sémantique de l'environnement.

Nous définissons dans notre modèle *un comportement d'agent crédible* par un comportement qu'un être humain pourrait réaliser s'il se trouvait dans la même situation que l'agent. La crédibilité est une notion subjective qui dépend fortement du regard des observateurs humains, mais aussi complexe regroupant plusieurs critères. En effet un comportement crédible peut être assimilé à un comportement possible, mais aussi à un comportement fréquent [Campano, 2013].

Pour réaliser cette évaluation, nous avons choisi de montrer des vidéos de simulation produites avec notre modèle de représentation de l'EV à des observateurs humains, et de leur poser des questions à propos des propositions de l'environnement (représentées par

les objets choisis par les agents pour réaliser leurs actions).

Les scénarios que nous avons choisi d'implémenter pour réaliser ces vidéos correspondent à des situations où l'on peut observer différentes *QoS* proposées par le même type d'objet selon différents contextes.

Notons que la crédibilité des propositions de l'environnement devrait correspondre à l'action que l'agent souhaite réaliser (retirer de l'argent) et non pas à son plan global (retirer de l'argent, pour pouvoir acheter de quoi manger), car l'environnement ne connait pas tout le plan de l'agent et il n'est pas sensé l'influencer dans ces choix d'action. Notre modèle sémantique de l'environnement a été utilisé dans un modèle cognitif d'agents [Reynaud, 2014] où a été réalisée une étude de la crédibilité sur le choix des agents tenant compte de leur plan global.

L'évaluation que nous présentons dans ce chapitre est organisée de la façon suivante : nous formulons les hypothèses que nous souhaitons vérifier à propos de notre modèle de représentation (section 5.2), nous expliquons le protocole expérimental utilisé pour vérifier ces hypothèses (section 5.3), puis nous détaillons les résultats obtenus (section 5.4). Nous terminons par la discussion de ces résultats (section 5.5) et une conclusion.

5.2 Hypothèses Générales

Nous avons voulu prouver par notre évaluation subjective que la *QoS* fournie par notre module permet aux agents de faire le meilleur choix d'objet pour réaliser leurs actions. Les agents ont besoin de s'informer sur leur environnement selon différents moments de la simulation. C'est pour cela que nous avons défini nos trois types d'interaction entre les agents et leur environnement. Ainsi nous dirons que les objets proposés aux agents au moment opportun pour réaliser leurs actions permettent d'augmenter la crédibilité de leurs comportements.

Notre modèle tient compte d'un certain nombre de paramètres telles que,(i) l'efficacité, qui vérifie si l'objet fournit le service recherché avec un rendu maximal. La qualité du rendu est obtenue en vérifiant si tous les critères pour avoir le service sont fournis par l'objet (le critère de confort, de coût, d'hygiène, etc), (ii) la proximité, qui détermine pour un objet donné s'il est facilement accessible ou pas dans la ville, c'est-à-dire si nous pouvons le trouver en abondance ce qui faciliterait sa recherche (les boites aux lettres sont nombreuses dans la ville ce qui les rend plus faciles à trouver par rapport à un bureau de poste pour une personne qui ne connait pas le quartier), (iii) la difficulté, permet d'évaluer les conditions supplémentaires qui peuvent s'ajouter pour accéder au service en utilisant un type d'objet (devoir consommer (ce qui représente un coût supplémentaire) pour pouvoir s'asseoir sur une chaise de café). Les paramètres que nous venons de citer ci-dessus, nous permettent d'obtenir une liste d'objets qui répond à la demande de l'agent indépendamment de ses préférences individuelles (préférer le Mc Donald's au Quick par exemple).

Les agents ont besoin de s'informer sur leur environnement selon différents moments de la simulation. C'est pour cela que nous avons défini nos trois types d'interaction entre les agents et leur environnement : (i) Le mode interrogatif, pour répondre aux requêtes des agents lorsqu'ils construisent leur plan. Ils auront en retour les objets qui leur permettent de réaliser leurs actions. (ii) le mode réactif, pour proposer la meilleure alternative qui s'offre à l'agent entre le moment où il a posé sa requête et le moment où il s'apprête à réaliser son action, (iii) le mode proactif, qui lui permet de réaliser des actions d'une manière opportuniste.

Nous allons donc vérifier que chaque paramètre de calcul de la *QoS* qu'il soit seul ou conjugué avec les autres paramètres de la *QoS* a un rôle dans la crédibilité des comporte-

ments des agents. Nous allons aussi vérifier que les modes d'interaction avec l'environnement permettent d'augmenter la crédibilité des comportements des agents.

5.2.1 Crédibilité dans les choix

Nos hypothèses sont fondées sur le principe que la *QoS* est utilisée pour filtrer les objets de l'environnement qui offrent aux agents la possibilité de réaliser au mieux leurs actions tout en leur permettant d'avoir des comportements crédibles.

- **Hyp1 :** les objets proposés aux agents sont jugés pertinents par un observateur humain par rapport aux actions que les agents doivent réaliser,
- **Hyp2 :** les critères de sélection de la *QoS* qui ont conduit les agents à leurs choix correspondent à ceux des observateurs humains soumis à la même situation.

5.2.2 Crédibilité des comportements des agents

Nous avons veillé à ne pas influencer les choix des agents, et nous avons voulu proposer tous les objets possibles qui permettent aux agents de réaliser leurs actions de manière à ce que l'objet le plus pertinent soit proposé en premier. Notre objectif est de permettre aux agents d'avoir un comportement crédible en choisissant les objets qui leur sont proposés avec la *QoS* la plus élevée.

- **Hyp3 :** les propositions de l'environnement augmentent la crédibilité des comportements des agents,
- **Hyp4 :** les observateurs humains jugent que l'environnement a proposé tous les objets possibles à l'agent et que celui-ci les a pris en compte lors de son choix de comportement.

5.2.3 Crédibilité des interactions avec l'environnement

Nous proposons de renforcer les interactions entre les agents et leur environnement pour leur permettre d'avoir des comportements crédibles de part la possibilité de réagir en fonction d'une situation donnée et de pouvoir évaluer un contexte pour réagir en conséquences.

- **Hyp5 :** l'utilisation des différents modes d'interaction avec l'environnement augmente la crédibilité des comportements des agents,
  - **Hyp5a :** l'utilisation du *mode réactif* augmente la crédibilité des comportements des agents,
  - **Hyp5b :** l'utilisation du *mode Proactif* augmente la crédibilité des comportements des agents.

Notons que l'évaluation du *mode Interrogatif* découlera de la vérification des hypothèses **Hyp1**, **Hyp2**, **Hyp3**, et **Hyp4** que nous avons citées plus haut.

5.3 Protocole

L'approche que nous avons adoptée consiste à soumettre des vidéos de simulation avec notre modèle à des observateurs, auxquels nous avons posé des questions sur les aspects que nous voulions évaluer.

Nous avons produit plusieurs vidéos avec divers scénarios correspondant à différents contextes afin de vérifier nos hypothèses sur des actions variées. Nous avons aussi produit des vidéos avec différentes configurations de notre modèle, en jouant sur les paramètres pour mettre en avant le paramètre de la *QoS* que nous souhaitions évaluer.

Nos vidéos de simulation adaptent des scénarios dans lesquels : (i) la *QoS* dépend du contexte et de l'action souhaitée par l'agent, (ii) la *QoS* varie selon les poids attribués aux paramètres de calcul, afin de montrer l'importance de chacun d'entre eux, (iii) l'activation / désactivation des modes d'interaction entre les agents et leur environnement pour montrer l'impact de chaque mode sur la crédibilité des comportements des agents.

Le premier scénario met en situation différents agents voulant réaliser la même action selon différents paramétrages de calcul de la *QoS*. Le second scénario consiste à observer le comportement des agents dotés ou pas des différents modes d'interaction avec leur environnement.

Nous avons mis en ligne le questionnaire et les vidéos sur un site web, de sorte qu'un participant répondant au questionnaire ne soit interrogé que sur un seul aspect de notre évaluation par page web, tirée aléatoirement parmi les quatre séries de vidéos. Nous avons traité un seul paramètre d'évaluation par page de questionnaire. Dans chaque page, nous avons présenté deux vidéos sur lesquelles nous avons posé des questions. Nous avons mis dans un ordre aléatoire les vidéos de chaque page. Nous avons combiné deux types de questions (fermées/ouvertes) afin de permettre aux observateurs de donner leur avis plus librement.

Dans un premier temps, nous avons diffusé notre questionnaire en interne au sein du LIP6 (Laboratoire Informatique de Paris 6), les doctorants du LIMSI (Laboratoire d'Informatique pour la Mécanique et les Sciences de l'Ingénieur) mais aussi auprès des amis et de la famille qui n'ont pas de connaissance sur notre modèle. L'objectif était de faire une pré-évaluation pour revoir la manière de poser nos questions mais aussi pour s'assurer que le contenu des vidéos était compréhensible par les évaluateurs en leur permettant de se focaliser sur l'aspect que nous voulions évaluer.

Après avoir validé et corrigé notre questionnaire d'évaluation, nous avons lancé l'évaluation sur un plus grand nombre de participants. Nous avons diffusé le questionnaire sur la liste de diffusion «liste des échos» du RISC (Relais d'Information sur les Sciences de la Cognition) [1].

Nous présenterons dans ce qui suit le questionnaire que nous avons proposé 5.3.3. Puis, nous donnerons les détails de production des vidéos d'évaluation présentées dans chaque page du questionnaire 5.3.4. Nous présenterons et nous discuterons par la suite les résultats que nous avons obtenus dans notre évaluation subjective 5.4.

5.3.1 Principe

5.3.1.1 Évaluation selon le contexte

Nous avons proposé deux séries de vidéos réalisées avec deux scénarios différents. Chaque série de vidéos contenait deux vidéos produites avec un paramétrage différent pour le calcul de la *QoS*. Nous avons voulu évaluer l'apport de chacun des paramètres de calcul de la *QoS* en le mettant à nul dans l'équation de calcul de la *QoS*. Par la suite nous avons comparé les deux vidéos de chaque série sur le critère de *crédibilité des comportements des agents*.

L'ignorance du paramètre *d'Efficacité* dans le calcul de la *QoS* suppose que le module sémantique de l'environnement évalue les objets selon le service de comportement qu'ils implémentent en ignorant les critères d'évaluation du service d'action demandé par l'agent, ce qui veut dire ses préférences. Par ailleurs, l'ignorance du paramètre de *Difficulté* implique que le module sémantique de l'environnement ne tiendra pas compte des conditions supplémentaires qui peuvent s'ajouter pour atteindre un service qui peut dépendre

1. www.risc.cnrs.fr/

d'un autre service proposé par le même type d'objet. Et enfin, l'ignorance du paramètre de *Proximité* implique que le module sémantique ne tiendra pas compte de l'abondance des types d'objets qui peuvent proposer le service, donc, il ne fera pas la différence entre un type d'objet très accessible, donc moins difficile à trouver qu'un type d'objet pas très fréquent dans l'environnement et qui peut être le seul moyen pour avoir un service donné.

5.3.1.2 Évaluation selon les modes d'interaction

Nous avons proposé différents modes d'interactions entre les agents et leur environnement. Nous avons évalué l'impact de chacun des trois modes proposés sur la crédibilité du comportement des agents, en désactivant le mode que nous souhaitions évaluer pour générer une vidéo. Nous avons par la suite comparé avec une autre vidéo comportant le même scénario, et générée cette fois-ci en activant le mode évalué.

Le mode *Interrogatif* est utilisé dans toutes les vidéos d'évaluation car il est indispensable pour que les agents puissent exécuter leurs plans.

Les requêtes envoyées en mode *Proactif* permettent à l'agent de réaliser une action *peu prioritaire* lorsque l'occasion se présente. De ce fait, il ne sera pas nécessaire d'adapter tout le plan en fonction de cette action qui peut être réalisée à tout moment d'une manière opportuniste. Comme par exemple, jeter une serviette en papier après avoir terminé de manger un sandwich en allant au boulot.

Les requêtes envoyées en mode *Réactif* permette à l'agent de s'adapter aux changements d'état d'un objet qu'il souhaite utiliser pour réaliser son action. Nous pouvons imaginer que l'état d'occupation d'un distributeur de billet peut changer entre le temps où l'agent se dirige vers lui et le moment où il arrive.

5.3.2 Configuration de notre modèle pour générer les vidéos

Nous avons réalisé quatre séries de vidéos, chacune correspondant à l'évaluation d'un aspect de notre modèle. Nous avons procédé en générant pour chaque série de vidéos une vidéo avec la configuration complète de notre modèle, et une vidéo sans la caractéristique que nous souhaitions évaluer.

Une configuration complète de notre modèle consiste à sélectionner et ordonner les objets répondants à la requête d'un agent en fonction de la QoS calculée avec les trois paramètres que nous avons définis (Efficacité, Difficulté et Proximité). Ces paramètres ont tous le même poids pour le calcul de la QoS. Pour compléter cette configuration du modèle, l'environnement doit répondre à tous les modes d'interaction avec les agents (mode Interrogatif, Réactif et Proactif).

Les quatre groupes de vidéos sont produits avec les conditions suivantes :

- **Groupe 1 (Paramètre de Difficulté) :** nous avons produit deux vidéos en relation avec l'hypothèse que le choix des agents est plus crédible lorsque le paramètre de *Difficulté* est pris en compte dans le calcul de la QoS (condition D). La première vidéo v1a est produite en calculant la QoS sans le paramètre de difficulté (condition E $\wedge$ ND $\wedge$ P). La deuxième vidéo v1b est produite en utilisant la configuration complète de notre modèle (condition E $\wedge$ D $\wedge$ P).
- **Groupe 2 (Paramètre de Proximité) :** nous avons produit deux vidéos pour ce groupe qui est en relation avec l'hypothèse que le choix des agents est plus crédible lorsque le paramètre de *Proximité* est pris en compte dans le calcul de la QoS (condition P). La vidéo v2a est produite sans tenir compte du paramètre de proximité dans le calcul de la QoS (condition E $\wedge$ D $\wedge$ NP). Contrairement à la vidéo v2b qui

a une configuration prenant en compte tous les paramètres de la *QoS* (condition E $\wedge$ D $\wedge$ P).

- **Groupe 3 (Mode Réactif) :** deux vidéos ont été produites pour ce groupe en relation avec l'hypothèse Hyp5a, portant sur l'impact du mode réactif sur la crédibilité des comportements des agents vis-à-vis de leur environnement. Nous avons proposé d'observer les comportements des agents dans la vidéo v3a produite avec la configuration complète de notre modèle (condition (E $\wedge$ D $\wedge$ P) $\wedge$ (MI $\wedge$ MR $\wedge$ MP)), et de la vidéo v3b produite sans le mode réactif (condition (E $\wedge$ D $\wedge$ P) $\wedge$ (MI $\wedge$ NMR $\wedge$ MP)).
- **Groupe 4 (Mode Proactif) :** nous avons produit une seule vidéo pour ce groupe en relation avec l'hypothèse Hyp5b, portant sur l'impact du mode proactif sur la crédibilité des comportements des agents. La vidéo v4 est produite avec la configuration complète de notre modèle (condition (E $\wedge$ D $\wedge$ P) $\wedge$ (MI $\wedge$ MR $\wedge$ MP)).

5.3.3 Description du questionnaire d'évaluation

Le questionnaire comprenait 5 pages. La première page concernait le recueil d'informations sur les participants, notamment sur leur profession, domaine d'études ou de recherche et leur degré de familiarité avec les simulations comportementales (incluant les jeux vidéos et tout type d'animation 3D).

Les autres pages du questionnaire contenaient chacune deux vidéos. Chaque page correspondait à l'évaluation d'une caractéristique de notre modèle. En haut de page était placé un texte contenant la description de l'action qui doit être réalisée dans la vidéo.

Nous avons posé les mêmes questions pour les deux vidéos de chaque page. Chaque vidéo était suivie d'une série de questions. Ces questions étaient soit des questions fermées qui étaient obligatoires, ou des questions ouvertes non obligatoires.

Les réponses aux questions fermées se basent sur l'échelle de *Likert* à 5 niveaux, dont les réponses possibles étaient : *pas du tout d'accord / pas d'accord / moyennement d'accord / d'accord / tout à fait d'accord.*

En bas de chacune des questions (où les participants devaient répondre selon l'échelle de Likert) il y avait un champ de commentaires libres.

Pour chaque participant l'ordre des pages était sélectionné aléatoirement. Nous avons fait la même chose pour l'ordre des vidéos qui apparaissaient sur chacune des pages. Ce procédé, appelé contrebalancement des passations, est utilisé pour éviter des biais dans les réponses des participants qui seraient liés à l'ordre dans lequel ils voient les vidéos.

Les participants avaient le droit de revoir plusieurs fois les vidéos présentes sur ces pages. En revanche ils ne pouvaient pas revenir en arrière une fois la page validée.

5.3.3.1 Page d'accueil

La page d'accueil contenait le texte suivant :

*« **Introduction :** Le projet de recherche « Terra Dynamica » a pour objectif de peupler une ville virtuelle (en l'occurrence, la reproduction de Paris en 3D) à l'aide de personnages animés capables de réaliser les activités des habitants de manière crédible et réaliste. Dans ce projet, Thales Training & Simulation et l'Université Pierre et Marie Curie développent des modèles informatiques pour que l'environnement virtuel (dans notre cas la ville) et les personnages puissent communiquer, pour que ces derniers se comportent de manière*

appropriée. »

*« **Expérimentation :** Vous allez participer à l'évaluation de notre modèle. L'expérience dure environ 10 minutes. À chaque fois, nous vous présentons une ou plusieurs vidéos correspondant à une simulation dans la plate-forme Terra Dynamica et nous vous demandons de répondre à quelques questions simples sur ces vidéos. Chaque scène est inspirée d'une situation réelle ; leur analyse ne nécessite aucune expertise en informatique. »*

Les évaluateurs devront cliquer sur un lien qui va les renvoyer vers la première page du questionnaire.

Des captures d'écran du questionnaire sont montrés dans l'annexe A.

5.3.3.2 Évaluation du paramètre "Difficulté"

Cette page concerne l'évaluation du critère *Difficulté* de la *QoS*. Nous conclurons aussi sur le critère *d'Efficacité* de la *QoS* si les participants jugent le comportement des personnages crédible (en choisissant l'objet approprié pour réaliser l'action souhaitée).

Scénario : Le scénario que nous avons défini pour cette évaluation consiste à trouver un objet dans la ville qui va permettre à l'agent de *s'asseoir*. Plusieurs choix s'offrent alors à lui :

- l'agent va s'asseoir sur *un banc public* (vidéo v1a, figure 5.1(a)),
- l'agent va s'asseoir dans *un bar café* (vidéo v1b, figure 5.1(b)).

(a) Vidéo v1a : avec le paramètre "Difficulté"

(b) Vidéo v1b : sans le paramètre "Difficulté"

Figure 5.1 – (a) L'agent va s'asseoir sur un banc public (b) L'agent va s'asseoir dans un bar café

Description : *Notre personnage souhaite s'asseoir pour se reposer. Précisons, que notre personnage ne souhaite réaliser aucune autre action (les vidéos de cette page seront suivies par une liste de question décrites dans les tableaux 5.1, 5.2).*

Label	Question
C1a	« Le choix du personnage dans la vidéo est crédible »
Prox1a	« Le choix du personnage est motivé par la proximité »
Diff1a	« Le choix du personnage est motivé par la gratuité du banc »
Eff1a	« Le choix du personnage est motivé par la possibilité de s'asseoir sur le banc (fonctionnalité) »
Disp1a	« Le choix du personnage est motivé par le fait qu'il y ait une place libre sur le banc (disponibilité)»

TABLE 5.1 – Questions sur la **Vidéo v1a :** *QoS* avec le paramètre "Difficulté"

Label	Question
C1b	« Le choix du personnage dans la vidéo est crédible »
Prox1b	« Le choix du personnage est motivé par la proximité »
Diff1b	« Le choix du personnage est motivé par le prix du service "s'asseoir au café" »
Eff1b	« Le choix du personnage est motivé par la possibilité de s'asseoir au café (fonctionnalité) »
Disp1ab	« Le choix du personnage est motivé par le fait qu'il y ait des places dans le café (disponibilité) »

TABLE 5.2 – Questions sur la **Vidéo v1b :** *QoS* sans le paramètre "Difficulté"

5.3.3.3 Évaluation du paramètre "Proximité"

Cette page du questionnaire concerne l'évaluation du critère de *Proximité* de la *QoS*. Nous validerons en conséquent le critère *d'Efficacité* de la *QoS* si les participants étaient d'accord sur les deux choix de l'agent dans les différentes vidéos.

(a) Vidéo v2a : avec le paramètre "Proximité" (b) Vidéo v2b : sans le paramètre "Proximité"

FIGURE 5.2 – (a) L'agent va dans un restaurant (b) L'agent va dans des toilettes publiques

Scénario : Dans ce scénario, un agent souhaite *aller aux toilettes*, il se trouve dans un quartier de la ville. Plusieurs choix s'offrent alors à lui :

– l'agent va se diriger vers *un fast food* pour utiliser leurs toilettes(vidéo v2a, 5.2(a)),

– l'agent va se diriger vers des *toilettes publiques* (vidéo v2b, figure 5.2(b)).

Description : *Notre personnage souhaite aller aux toilettes. Précisons que le personnage ne souhaite réaliser aucune autre action. Nous allons comparer deux situations pour voir comment notre système va réagir (les vidéos de cette page seront suivies par une liste de question décrites dans le tableau 5.3, 5.4).*

Label	Question
C2a	« Le choix du personnage dans la vidéo est crédible »
Eff2a	« Le choix du personnage est motivé par la possibilité d'utiliser les toilettes du restaurant (fonctionnalité) »
Disp2a	« Le choix du personnage est motivé par la disponibilité des toilettes du restaurant (disponibilité) »
Diff2a	« Le choix du personnage est motivé par le prix du service "aller aux toilettes" du restaurant »
Prox2a	« Le choix du personnage est motivé par la proximité des toilettes du restaurant (proximité) »
CHX	« Le personnage se rend compte qu'il n'a pas d'autres choix que les toilettes du restaurant »

Table 5.3 – Questions sur la **Vidéo v2a :** *QoS* avec le paramètre "Proximité"

Label	Question
C2b	« Le choix du personnage dans la vidéo est crédible »
Disp2b	« Le choix du personnage est motivé par la disponibilité des toilettes publiques (disponibilité) »
Prox2b	« Le choix du personnage est motivé par la proximité des toilettes publiques (proximité) »
Diff2b	« Le choix du personnage est motivé par la gratuité des toilettes publiques »
Eff2b	« Le choix du personnage est motivé par la possibilité d'aller aux toilettes (la fonctionnalité) »
CNN	« Le personnage devait connaitre le quartier, ce qui ne lui a pas posé de difficultés pour trouver les toilettes publiques »
FND	« À votre avis, quel est le plus facile à trouver entre ces différents objets ? »

Table 5.4 – Questions sur la **Vidéo v2b :** *QoS* sans le paramètre "Proximité"

5.3.3.4 Évaluation du mode "Réactif"

Cette page de notre questionnaire concerne l'évaluation de l'apport en crédibilité du *Mode Réactif* de notre modèle. Nous avons posé exactement les mêmes questions pour les deux vidéos.

Scénario : Dans ce scénario, plusieurs agents souhaitent retirer de l'argent, ils se trouvent dans le même endroit et vont se diriger vers des distributeurs automatiques. Nous proposerons deux situations, dans la première les agents ne pourront pas accéder aux données sur leur environnement avec le mode réactif (vidéo v3a, figure 5.3(a)). Dans la seconde, les agents pourront accéder aux données sur leur environnement avec le mode réactif (vidéo v3b, figure 5.3(b)).

(a) Vidéo v3a : avec le mode Réactif (b) Vidéo v3b : sans le mode Réactif

FIGURE 5.3 – (a) Les agents se dirigent vers les distributeurs libres (b) Les agents font la queue derrière le même distributeur

Description : *Veuillez observer le comportement des personnages qui souhaitent retirer de l'argent (les vidéos de cette page seront suivies d'une série de questions décrites dans le tableau 5.5).*

Label	Question
C3	« Le comportement des personnages est crédible »
Q1	« Les personnages n'arrivent pas à identifier la file d'attente qui avance le plus rapidement »
Q2	« Les personnages n'arrivent pas à voir ce qui se passe autour d'eux »
Q3	« Les personnages ne sont pas capables d'identifier les autres distributeurs comme des objets leur permettant de faire leur action »
Q4	« Les personnages ne tiennent pas compte de l'état des autres distributeurs »

TABLE 5.5 – Questions sur les **Vidéos v3a et v3b :** Avec et sans le mode "Réactif"

5.3.3.5 Évaluation du mode "Proactif"

Cette page de notre questionnaire concerne l'évaluation de l'apport en crédibilité du *Mode Proactif* de notre modèle.

Scénario : Un agent capable de mettre des priorités sur les actions qu'il veut réaliser devra retirer de l'argent pour aller manger au restaurant. Au cours de son chemin il désire

jeter son reçu qu'il considère comme une action non prioritaire et qu'il pourra réaliser d'une manière opportuniste.

Figure 5.4 – Vidéo v4 (mode Proactif) : L'agent se dirige vers une poubelle après avoir retiré de l'argent du distributeur, puis il se rend au restaurant

Description : *Notre personnage souhaite aller manger, mais il doit d'abord passer par le distributeur pour retirer de l'argent (cette vidéo (figure 5.4) sera suivie d'une liste de questions décrites dans le tableau 5.6).*

Label	Question
C4	« Le comportement des personnages est crédible »
Q5	« Le personnage était attentif à ce qu'il voyait sur son chemin vers le restaurant »
Q6	« Le personnage était réactif en voyant la poubelle sur son chemin vers le restaurant »
Q8	***Autre situation :*** *Supposons que la poubelle qu'a utilisé le personnage se trouvait sur le trottoir d'en face.* « Le comportement du personnage est crédible s'il avait traversé la rue pour jeter son reçu, puis qu'il retraverse pour qu'il aille au restaurant »

Table 5.6 – Questions sur la **Vidéo v4 :** Avec le mode "Proactif"

5.3.4 Réalisation des scénarios d'évaluation

Nous avons produit sept vidéos. Celles-ci ont été enregistrées avec le simulateur du projet *Terra Dynamica*, dans lequel s'intègre notre modèle. Nous expliquons ci-après comment notre modèle a été configuré pour produire chacune de ces vidéos, et nous donnons une description pour chacune d'entre elles.

5.3.4.1 Évaluation du paramètre *Difficulté*

a. Scénario

Dans le premier scénario (page 1 du questionnaire) nous avons un agent qui cherche l'objet qui propose le service qui lui permet de "s'asseoir". L'agent aura alors le choix entre le *banc public* et *les chaises d'un café*. Nous avons mis l'agent dans le même contexte (le même endroit de la ville) avec les mêmes conditions (bancs publics vides, et bar café ouvert). Nous avons généré la première vidéo en prenant en compte le paramètre *Difficulté* dans le calcul de la *QoS* afin de classer les propositions d'objets à l'agent. Ce paramètre a été ignoré pour la génération de la deuxième vidéo.

b. Requête de l'agent

La requête de l'agent sera interprétée par l'environnement de telle sorte à faire ressortir ses préférences à travers le service qu'il aura demandé. Dans le cas de notre scénario l'agent va demander le service $s =$*S'asseoir* avec $s \in S_a$. Le service d'action $s =$ *S'asseoir* implémente le service de comportement $s' =$*se Reposer* (avec $s' \in S_c$).

Le service s sera évalué selon les paramètres suivants : $C_r(s) = \{$*Confort, Calme*$\}$. L'ensemble $EQ(s)$ *(Estimated Quality)* contient toutes les combinaisons des couples (critère, valeur) possibles pour le service s, avec $EQ(s) = \{(c, v) \in C_r(s) \times V_q(s, c)\} = \{$ *(Confort, L),(Confort, M), (Confort, S), (Calme, L),(Calme, M), (Calme, S)*$\}$.

L'environnement va interpréter la requête de l'agent en fonction du service d'action qu'il souhaite réaliser, il va donc lui proposer un objet qui propose le service d'action $s =$*S'asseoir* avec un minimum de *Calme* et de *Confort*.

c. Contexte de la simulation

L'agent a le choix entre deux types d'objets qui lui permettent de réaliser l'action $s =$ *S'asseoir*. D'une part, il aura l'objet $o_1 =$ *Banc public*, qui dispose de places libres et qui propose le service *gratuitement*. D'autre part, il y a l'objet $o_2 =$ *Bar Café*, qui dispose de chaises à l'intérieur pour s'assoir. Cependant, l'agent doit consommer quelque chose sur place pour pouvoir s'asseoir au *Bar Café*. Cette contrainte rentra en compte dans le paramètre de *Difficulté* pour le calcul de la *QoS* proposée par cet objet o_2.

Nous avons défini pour chaque type d'objet proposant le service $s =$ *s'asseoir* l'ensemble $EQ(s, o_i)$ des effets attendus par son utilisation. Avec :

$EQ(s) = \{$ *(Confort, L),(Confort, M), (Confort, S), (Calme, L),(Calme, M), (Calme, S)* $\}$, les différentes combinaisons de paires (*critère , valeur*) pour le service s.
$EQ(s, o_1) = \{$ *(Confort, M), (Calme, S)* $\}$,
$EQ(s, o_2) = \{$ *(Confort, M), (Calme, M)* $\}$.

L'objet $o_2 =$ *Bar Café* propose le service $s =$*S'asseoir* avec une relation de *dépendance* (*dep*) avec un autre service $s' =$ *Boire café* qui exige que l'agent consomme pour pouvoir accéder au service souhaité $s =$ *s'asseoir*.

Cette relation de dépendance va donc rajouter une condition à notre agent qui est représentée par le critère *Coût* qui a comme valeur du service :

$EQ(s', o_2) = \{$ *(Coût, S)* $\}$.

d. Initialisation des paramètres de calcul de la QoS

Afin de bien mettre en évidence le paramètre de *Difficulté* que nous voulons évaluer dans ces vidéos, nous avons choisi un scénario où les deux autres paramètres de calcul de la *QoS Efficacité* et *Proximité* ont la même valeur pour les deux types d'objet o_1 =*Banc public* et o_2 =*Bar café*.

Les trois paramètres de calcul de la QoS pour chaque objet sont définis comme suit :

Efficacité ($Eff(o_1, s)$)

$$Eff(o_1, s) = \frac{|EQ(s) \cap EQ(o_1, s)|}{|QC(o_1, s)|} \text{ [2]}$$

Avec :
$QC(o_1, s)$ = les critères appliqués à l'objet o_1 pour évaluer le service s.
$QC(o_1, s) = \{$ *Confort, Calme* $\}$, donc, $|QC(o_1, s)| = 2$
et,
$|EQ(s) \cap EQ(o_1, s)| = \{$*(Confort, L), (Confort, M), (Confort, S), (Calme, L), (Calme, M), (Calme, S)*$\} \cap \{$*(Confort, M), (Calme, S)*$\} = \{$*(Confort, M), (Calme, S)*$\}$.
Nous aurons donc :

$$Eff(o_1, s) = \frac{2}{2} = 1$$

Difficulté ($Diff(o_1, s)$)

Le service s = *s'asseoir* proposé par l'objet o_1 = *Banc public* n'a pas de dépendance avec un autre service proposé par le même objet. La valeur du paramètre de difficulté est donc nulle $Diff(o_1, s) = 0$.

Proximité ($Prox(o_1, s)$)

Nous avons calculé le critère de Proximité sur un rayon de 50 mètres. Le principe est de comparer le nombre d'instances de l'objet o_1 = *Banc public* par rapport à tous les autres objets qui peuvent proposer le même service recherché dans un périmètre observable par l'agent.

Dans notre cas, l'agent a le choix entre deux types d'objet o_1 = *Banc public* et o_2 = *Bar café*. Nous allons donc comparer entre le nombre d'instances de o_1 par rapport à o_2.

Nous avons 2 bancs publics et 2 bar café dans un rayon de 50 mètres, la proximité est calculée comme suit :

$$Prox(o_1, s) = \frac{2}{4} = 0.5$$

2. On note par $|X|$ le nombre d'éléments de l'ensemble X

Efficacité ($Eff(o_2, s)$)

$$Eff(o_2, s) = \frac{|EQ(s) \cap EQ(o_2, s)|}{|QC(o_2, s)|}$$

Avec :
$QC(o_2, s) =$ { *Confort, Calme* }, donc $|QC(o_2, s)| = 2$
et,
$|EQ(s) \cap EQ(o_2, s)| =$ { *(Confort, L),(Confort, M), (Confort, S), (Calme, L), (Calme, M), (Calme, S)* } $\cap$ { *(Confort, M), (Calme, M)* } = { *(Confort, M), (Calme, M)* }.

$$Eff(o_2, s) = \frac{2}{2} = 1$$

Difficulté ($Diff(o_2, s)$)

Le service $s = s'asseoir$ proposé par l'objet $o_2 = Bar\ café$ a une relation de dépendance avec un autre service $s' = Boire\ café$ proposé par le même objet.

$$Diff(o, s) = \frac{|C_r(s')| - |QC(o, s) \cap QC(o, s')|}{|C_r(s')|}$$

Avec :
L'ensemble des critères d'évaluation du service s' est : $C_r(s') =$ { *Coût* }, et $QC(o_2, s') =$ { *Coût* }.

Il n'y a donc aucun critère d'évaluation en commun entre les deux services proposés par le *Bar café*, donc $|QC(o, s) \cap QC(o, s')| = 0$. Cela veut dire que l'agent doit rajouter une condition supplémentaire (*Coût*) pour accéder au service s qu'il souhaite en utilisant l'objet $o_2 = Bar\ café$.

$$Diff(o_2, s) = \frac{1 - 0}{1} = 1$$

Proximité ($Prox(o_2, s)$)

Nous avons 2 bancs publics et 2 bar café dans un rayon de 50 mètres, la proximité est calculée comme suit :

$$Prox(o_2, s) = \frac{2}{4} = 0.5$$

4.3.4.2 Évaluation du paramètre *Proximité*

a. Scénario

Dans ce deuxième scénario d'évaluation, nous avons un agent qui souhaite aller aux toilettes. L'agent devra choisir entre deux types d'objet (les toilettes publiques et celles d'un restaurant). L'agent ne connait pas le quartier. Il sera mis dans le même contexte pour réaliser les deux scénarios (même endroit de la ville) et avec les mêmes conditions (Toilettes publiques gratuites, restaurants ouverts). Lors de la réalisation de la première vidéo , nous avons pris en compte le paramètre *Proximité* tel que nous l'avons défini dans

notre modèle dans le calcul de la *QoS* afin de classer les propositions d'objets à l'agent. Nous avons par la suite ignoré ce paramètre pour la réalisation de la deuxième vidéo.

b. Requête de l'agent

L'agent fait la demande dans sa requête du service $s =$*Aller aux toilettes* avec $s \in S_a$.

Le service s sera évalué selon les paramètres suivants : $C_r(s) = \{$*Hygiène*$\}$. L'ensemble $EQ(s)$ *(Estimated Quality)* contient toutes les combinaisons des couples (critère, valeur) possibles pour le service s, avec $EQ(s) = \{(c, v) \in C_r(s) \times V_q(s, c)\} = \{$ *(Hygiène, L),(Hygiène, M), (Hygiène, S)*$\}$.

Le module sémantique va scruter l'environnement pour chercher les objets qui proposent le service $s =$*Aller aux toilettes*, tout en satisfaisant les deux critères du service qui sont le coût et l'hygiène.

c. Contexte de la simulation

L'endroit où se trouve l'agent dans la ville lui offre la possibilité de choisir entre deux types d'objets $o_1 =$ *Toilettes publiques* et $o_2 =$ *Fast-Food*. Ces derniers seront départagés par le critère $C_r(s) = \{$*Hygiène*$\}$.

L'agent se trouve à proximité de plusieurs objets de type *Fast-Food*, qui sont en nombre donc facilement localisables. Par contre, les agents doivent consommer pour pouvoir utiliser leurs toilettes, ce qui engendre un coût. Contrairement à l'objet $o_1 =$*Toilettes publiques*, qui sont gratuites mais il n'existent qu'une seule entité dans l'endroit où se trouve notre agent et donc plus difficile à trouver dans l'environnement, ce qui va compter dans le paramètre *Proximité* pour le calcul de la *QoS* (qui nous intéresse dans ce scénario).

Les effets attendus par l'utilisation des types d'objets $o_1 =$ *Toilettes publiques* et $o_2 =$ *Fast-Food*, proposant le service $s =$ *Aller aux toilettes*, sont définis comme suit :

$EQ(s, o_1) = \{$ *(Hygiène, S)* $\}$,
$EQ(s, o_2) = \{$ *(Hygiène, M)* $\}$.

Les objets $o_2 =$ *Fast-Food* proposent le service $s =$*Aller aux toilettes* avec une relation de *dépendance* (*dep*) avec un autre service $s' =$ *Manger* qui exige que l'agent consomme pour pouvoir accéder au service souhaité $s =$ *Aller aux toilettes*.

Cette relation de dépendance va donc rajouter une condition à notre agent qui est représentée par le critère *Coût* qui a comme valeur du service :

$EQ(s', o_2) = \{$ *(Coût, M)* $\}$,

d. Initialisation des paramètres de calcul de la QoS

Les trois paramètres de calcul de la QoS sont définis comme suit :

Efficacité ($Eff(o_1, s)$)

$$Eff(o_1, s) = \frac{|EQ(s) \cap EQ(o_1, s)|}{|QC(o_1, s)|}$$

Avec :

$QC(o_1, s) =$ { *Hygiène* }, donc, $|QC(o_1, s)| = 1$
et,
$|EQ(s) \cap EQ(o_1, s)| =$*{(Hygiène, L), (Hygiène, M), (Hygiène, S)}∩{(Hygiène, S))}*
$|EQ(s) \cap EQ(o_1, s)| =$*{(Hygiène, S)}*.

$$Eff(o_1, s) = \frac{1}{1} = 1$$

Difficulté ($Diff(o_1, s)$)

Le service $s =$ *Aller aux toilettes* proposé par l'objet $o_1 =$ *Toilettes publiques* n'a pas de dépendance avec un autre service proposé par le même objet. La valeur du paramètre de difficulté est donc nulle $Diff(o_1, s) = 0$.

Proximité ($Prox(o_1, s)$)

Sur un rayon de 50 mètres, nous avons 1 *Toilettes publiques* et 4 *Fast-food*, la proximité est donc calculée comme suit :

$$Prox(o_1, s) = \frac{1}{5} = 0.2$$

Efficacité ($Eff(o_2, s)$)

$$Eff(o_2, s) = \frac{|EQ(s) \cap EQ(o_2, s)|}{|QC(o_2, s)|}$$

Avec :
$QC(o_2, s) =$ { *Hygiène* }, donc, $|QC(o_2, s)| = 1$
et,
$|EQ(s) \cap EQ(o_2, s)| =$*{(Hygiène, L), (Hygiène, M), (Hygiène, S)}∩{(Hygiène, M))}*
$|EQ(s) \cap EQ(o_2, s)| =$*{(Hygiène, M)}*.

$$Eff(o_2, s) = \frac{1}{1} = 1$$

Difficulté ($Diff(o_2, s)$)

Le service $s =$ *Aller* proposé par l'objet $o_2 =$ *Fast-food* a une relation de dépendance avec un autre service $s' =$ *Manger* proposé par le même objet.

$$Diff(o, s) = \frac{|C_r(s')| - |QC(o, s) \cap QC(o, s')|}{|C_r(s')|}$$

Avec :
L'ensemble des critères d'évaluation du service s' est : $C_r(s') =$ { *Coût* }, et $QC(o_2, s') =$ { *Coût* }.

Il n'y a donc aucun critère d'évaluation en commun entre les deux services proposés par le *Fast-food*, donc $|QC(o, s) \cap QC(o, s')| = 0$. Cela veut dire que l'agent doit rajouter une condition supplémentaire (*Coût*) pour accéder au service s qu'il souhaite en utilisant l'objet $o_2 =$.

$$Diff(o_2, s) = \frac{1-0}{1} = 1$$

Proximité ($Prox(o_2, s)$)

La proximité est calculée comme suit :

$$Prox(o_2, s) = \frac{4}{5} = 0.8$$

4.3.4.3 Évaluation du mode *Réactif*

Notre scénario consiste à faire réaliser la même action par différents agents. Notre but est de créer plusieurs situations différentes afin de voir comment nos agents vont réagir et de voir s'ils peuvent s'adapter aux nouvelles situations.

Notons que nos agents interrogent leur environnement en mode Interrogatif pour choisir les objets qu'ils vont utiliser pour réaliser leurs actions. Donc il y a un temps d'écart entre le moment où l'objet est choisi pour réaliser une action et le moment où il sera utilisé pour exécuter l'action. Il y a donc une probabilité que l'état de l'objet choisi (hors service, occupé, fermé, etc) ou que la situation (fille d'attente, manifestation, etc) change entre temps. Nous allons donc évaluer la crédibilité des comportements des agents en observant leur capacité de réagir et de s'adapter aux changements d'état ou de situation des objets qu'ils veulent utiliser.

Nous avons fait réaliser par plusieurs agents l'action de retirer de l'argent au distributeur de banque (page 3 du questionnaire). Nous avons quatre distributeurs l'un à coté de l'autre, qui sont tous en état de marche et disponibles au départ. Les agents auront le choix entre différents comportements selon différentes situations :

- se diriger vers un distributeur libre,
- se mettre dans la file d'attente où il y a le moins d'agents qui attendent (dans le cas où tous les distributeurs sont occupés),
- chercher un autre distributeur ailleurs (dans le cas où il y a de très longues files d'attente derrière tous les distributeurs).

4.3.4.4 Évaluation du mode *Proactif*

Notre vidéo est réalisée avec la configuration complète de notre modèle, de ce fait, l'agent peut agir d'une manière opportuniste lorsque l'occasion s'offre à lui.

Une partie du plan de l'agent était de retirer de l'argent au distributeur pour aller manger dans un restaurant. Parmi les actions que devait faire l'agent nous avons l'action *"jeter un papier"*. Cette action n'est pas prioritaire pour l'agent, il pourra donc l'exécuter lorsqu'il trouvera une poubelle sur son chemin en allant vers le restaurant. En activant le mode Proactif, l'agent choisira de jeter son papier dans la poubelle qui se trouve sur son chemin en se dirigeant vers le restaurant. Dans le cas où le mode proactif est désactivé, l'agent adoptera un tout autre plan où le module de décision sera contraint de choisir les objets que l'agent doit utiliser au moment de la planification de ses actions. De se fait, il se peut qu'une autre poubelle soit proposée sachant qu'elle aura la même QoS que la poubelle qu'il a utilisée dans la vidéo, sauf qu'au moment de choisir l'instance d'objet *poubelle*, le module de l'environnement ne pourra pas prendre en compte les paramètres contextuels (comme la distance) pour lui proposer la meilleure *instance* d'objet qui convient à la situation de l'agent.

5.4 Résultats

106 participants ont contribué à cette étude, dans une tranche d'âge allant de 18 à 74 ans. 50,58% d'entre eux étaient un peu familiarisés avec les jeux vidéos et les environnements virtuels, alors que 36,47% ne l'étaient pas et 12,95% l'étaient beaucoup.

Nous présentons dans cette section les différentes hypothèses que nous avons posées et les résultats qui leurs sont associés.

Nous avons noté par *Mo* le mode (valeur la plus fréquente d'une distribution), la moyenne par μ et l'écart-type par σ.

Le test statistique le plus adapté pour le traitement de nos résultats est le *test t de Student* [Gosset, 1908] (résultat que nous avons noté par t).

Nous présenterons dans la section suivante, tous les résultats que nous avons obtenus dans notre évaluation subjective. Par la suite nous discuterons ces résultats et nous conclurons sur ce chapitre.

5.4.1 Hypothèses sur la crédibilité dans les choix

L'hypothèse traitée dans cette section est liée à l'hypothèse générale **Hyp1**. Celle-ci concerne les vidéos v1a, v1b, v2a et v2b. Les vidéos v1a et v2a correspondent à une configuration complète de notre modèle, en utilisant tous les paramètres de calcul de la *QoS* pour proposer les objets aux agents. En revanche, les vidéos v1b et v2b correspondent à une configuration incomplète du modèle en excluant du calcul de la *QoS* respectivement les paramètres de *Difficulté* et de *Proximité*.

Ces deux configurations proposent aux agents la même liste d'objets qui doivent correspondre à l'action qui sera réalisée, mais ils seront présentés dans un ordre différents selon le paramétrage de calcul de la *QoS* utilisé.

Hyp1 : *les objets proposés aux agents sont jugés pertinents par les observateurs humain par rapport aux actions que les agents doivent réaliser.*

Les participants sont dans l'ensemble *d'accord* à *tout à fait d'accord* sur le fait que le choix des personnages était motivé par la fonction rendue par l'objet choisi pour réaliser l'action dans toutes les vidéos. La moyenne la plus faible pour la question Eff a été obtenue par la vidéo v1b, et la plus haute a été obtenue pour la vidéo v1a (voir tableau 5.7 et figure 5.5).

	μ	σ	*Mo*
vidéo v1a / Eff	4.57	0.88	5
vidéo v1b / Eff	3.59	1.44	5
vidéo v2a / Eff	3.95	1.25	5
vidéo v2b / Eff	4.06	1.15	5

TABLE 5.7 – Résultats obtenus sur la question portant sur le paramètre *d'Efficacité* pour les vidéos v1a, v1b, v2a et v2b

Les scores obtenus sont significatifs quand ils sont comparés à une moyenne de 3 avec un test t de *Student* ((v1a, Eff) : $t = 16.64$, $p < .0001$; (v2a, Eff) : $t = 7.14$, $p < .0001$; (v1b, Eff) : $t = 3.77$, $p < .0001$; (v2b, Eff) : $t = 8.54$, $p < .0001$). Ces résultats

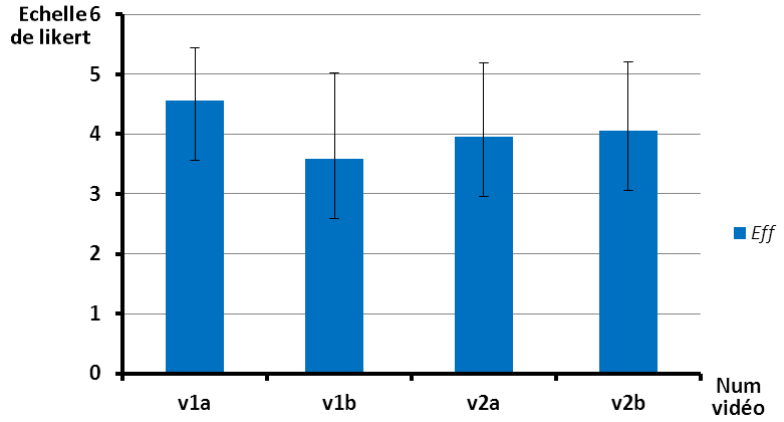


Figure 5.5 – Moyenne et écart-type des réponses à la question Eff par vidéo (échelle de Likert de 1 à 5)

soutiennent **Hyp1**.

Hyp1a : *les observateurs vont juger que les comportements des agents sont plus crédibles lorsque leur choix est basé sur le calcul de la QoS en tenant compte du paramètre de Difficulté, que lorsqu'ils se basent sur un calcul de QoS sans le paramètre de Difficulté.*

Les participants sont dans l'ensemble *d'accord* à *tout à fait d'accord* sur le fait que le comportement des agents était crédible dans la vidéo v1a. La moyenne la plus élevée pour la question C1 portant sur la crédibilité des comportements des agents a été obtenue par la vidéo v1a comparée à celle obtenue par la vidéo v1b (voir tableau 5.8).

	μ	σ	Mo
vidéo v1a / C1a	4.54	0.94	5
vidéo v1b / C1b	3.02	1.47	4

Table 5.8 – Résultats sur la crédibilité pour les vidéos v1a, v1b

Les résultats obtenus avec le test t de *Student* sont significatifs entre la vidéo v1a et v1b ($t = 8.11$; $p < .0001$). Ces résultats soutiennent **Hyp1a**.

Hyp1b : *les observateurs vont juger que les comportements des agents sont crédibles lorsque leur choix est basé sur le calcul de la QoS en tenant compte du paramètre de Proximité.*

Les participants sont dans l'ensemble *d'accord* à *tout à fait d'accord* sur le fait que le comportement de l'agent était crédible dans les deux vidéos v2a et v2b. La moyenne des réponses obtenue pour la vidéo v2b produite avec la condition (E $\wedge$ D $\wedge$ NP) été plus élevé que celle obtenue pour la vidéo v2a qui a été produite avec la configuration complète de notre modèle (condition E $\wedge$ D $\wedge$ P) (voir tableau 5.9). En effet, il n'est pas aberrant d'utiliser les toilettes publiques pour aller aux toilettes.

	μ	σ	Mo
vidéo v2a / C2a	3.47	1.41	5
vidéo v2b / C2b	4.29	1.08	5

TABLE 5.9 – Résultats sur la crédibilité pour les vidéos v2a et v2b

Les scores obtenus pour les deux vidéos v2a et v2b sont significatifs quand ils sont comparés entre eux avec un test t de *Student* ($t = 4.28$; $p < .0001$).

Cependant, la question C2a a obtenue une moyenne de 3.47 pour la vidéo v2a, sachant que la valeur la plus fréquente était 5 (sur l'échelle de Likert de 1 à 5). Notons que 70% des participants estimaient que le comportement de l'agent dans la vidéo v2a été crédible et la majorité avaient répondu *tout a fait d'accord* sur la question C2a.

Les scores obtenus sont significatifs quand ils sont comparés à une moyenne de 3 avec le test t de *Student* (v2a/C2a : $t = 3.11$; $p < .001$).

Pour autant, nous ne pouvons pas valider notre hypothèse **Hyp1b**. Nous discuterons ces résultats dans la section 5.5.

5.4.2 Hypothèses sur les paramètres de calcul de la QoS

Nous allons analyser dans ce qui suit les résultats liés à l'hypothèse générale Hyp2, concernant le groupe de vidéos v1a, v1b, v2a et v2b.

5.4.2.1 Paramètre de « Difficulté »

La vidéo v1a correspondait au calcul de la QoS avec le paramètre de *Difficulté* (condition $E \wedge D \wedge P$), et qui n'est pas pris en compte dans la vidéo v1b (condition $E \wedge ND \wedge P$).

Hyp2a : *les observateurs vont juger que les choix des agents sont liés au paramètre de Difficulté lorsqu'il est pris en compte dans le calcul de la QoS, mais pas lorsqu'il n'est pas pris en compte dans le calcul.*

Les participants sont dans l'ensemble *d'accord* à *tout à fait d'accord* sur le fait que le choix de l'agent était motivé par le critère de *Difficulté* (représenté ici par le prix pour accéder au service) dans la vidéo v1a. En comparant entre la moyenne obtenue par la vidéo v1a et v1b pour la question Diff, la moyenne de la vidéo v1a est plus élevée que celle de la vidéo v1b où nous n'avons pas pris en compte le paramètre de *Difficulté* dans le calcul de la QoS (voir tableau 5.10 et figure 5.6).

	μ	σ	Mo
vidéo v1a / Diff	4.38	1.09	5
vidéo v1b / Diff	2.18	1.39	1

TABLE 5.10 – Résultats sur la prise en compte du critère de difficulté dans la décision de l'agent dans les vidéos v1a et v1b

Les scores obtenus sur la question Diff pour ces deux vidéos sont significatifs entre eux ($t = 11.58$; $p < .0001$). Ces résultats soutiennent **Hyp2a**.

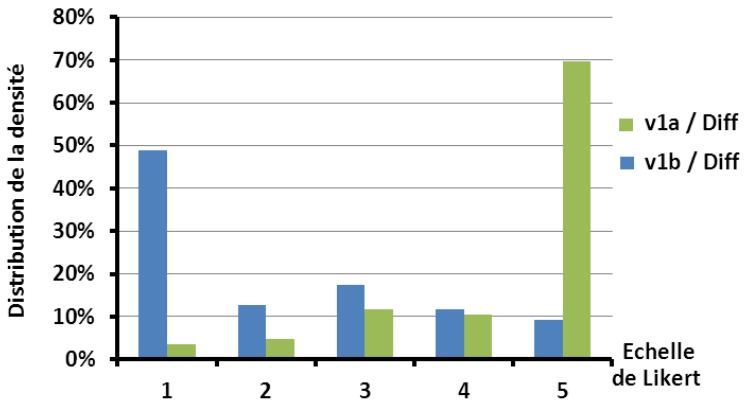


Figure 5.6 – Densité de la distribution des réponses à la question Diff pour les vidéos v1a et v1b

5.4.2.2 Paramètre de « Proximité »

La vidéo v2a correspondait au calcul de la *QoS* avec le paramètre de *Proximité* (condition E $\wedge$ D $\wedge$ P), contrairement à la vidéo v2b où nous avons ignoré le paramètre de proximité pour le calcul de la *QoS* (condition E $\wedge$ D $\wedge$ NP) pour trier les objets proposés aux agents.

Hyp2b : *les observateurs vont juger que les choix des agents sont liés au paramètre de Proximité lorsqu'il est pris en compte dans le calcul de la QoS, mais pas lorsqu'il n'est pas pris en compte dans le calcul.*

Les participants sont dans l'ensemble *d'accord* à *tout à fait d'accord* sur le fait que le choix de l'agent était motivé par le critère de *Proximité* (la facilité de trouver des toilettes ou pas) pour la vidéo v2a. La moyenne la plus élevée obtenue pour la question Prox a été celle de la vidéo v2a, en comparant avec celle obtenue pour la vidéo v2b (voir tableau 5.11 et figure 5.7). Les participants considèrent qu'il est plus facile de trouver un restaurant pour utiliser leur toilettes que de trouver les toilettes publiques. Ce constat est vérifié par les réponses à la question Fnd où 97% des participants ont répondu qu'il était plus facile de trouver un *Bar-café* ou *Restaurant* contrairement au 3% restant qui ont répondu *"Toilettes publiques"* (voir figure 5.8).

	μ	σ	Mo
vidéo v2a / Prox	3.65	1.46	5
vidéo v2b / Prox	3.61	1.18	3

Table 5.11 – Résultats sur la prise en compte du critère de proximité dans la décision de l'agent dans les vidéos v2a et v2b

Les scores obtenus sur la question Prox pour les vidéos v2a et v2b quand ils sont comparés entre eux *ne sont pas très significatifs* mais montrent au moins une tendance (t = 0.19 ; p = 0.8).

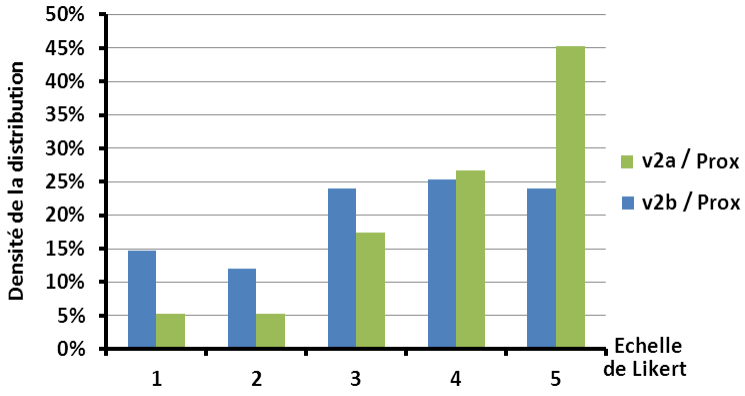


FIGURE 5.7 – Densité de la distribution des réponses à la question Prox pour les vidéos v2a et v2b

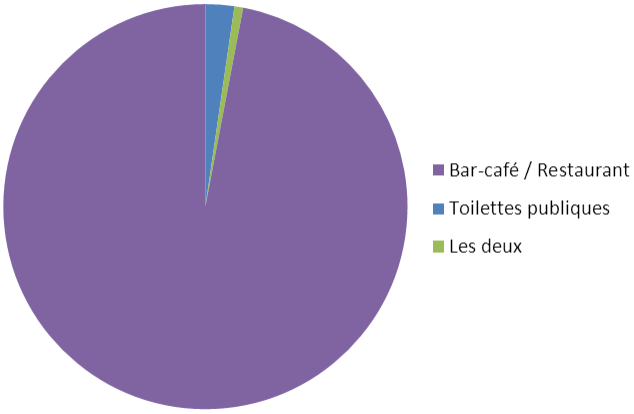


FIGURE 5.8 – Répartition des réponses à la question Fnd

Les réponses aux questions Prox pour les vidéos v2a et v2b sont soutenues par les réponses à la question Chx où les participants étaient *d'accord* à *tout à fait d'accord* sur le fait que le choix du personnage s'est porté sur les toilettes du restaurant parce qu'il n'y avait pas de toilettes publiques à proximité.

Les scores obtenus sont significatifs quand ils sont comparés à une moyenne de 3 avec un test t de *Student* (Chx : $t = 2.71$; $p < .001$).

	μ	σ	Mo
Chx	3.60	1.31	3
Cnn	4.15	1.12	5

TABLE 5.12 – Résultats obtenus sur les questions Chx et Cnn pour les vidéos v2a et v2b

En revanche, les participants sont dans l'ensemble *d'accord* à *tout à fait d'accord* sur le fait que si l'agent a choisi les toilettes publiques c'est qu'il connaissait forcément l'endroit

et donc l'existence des toilettes publiques à proximité, ce qui ne lui avait pas donné de mal pour les trouver (voir tableau 5.12). Nous avons vérifié que les scores obtenus pour la question Cnn étaient significatifs en les comparant avec une moyenne de 3 avec un test t de *Student* (v2b, Cnn : t = 8.84 ; p < .0001).

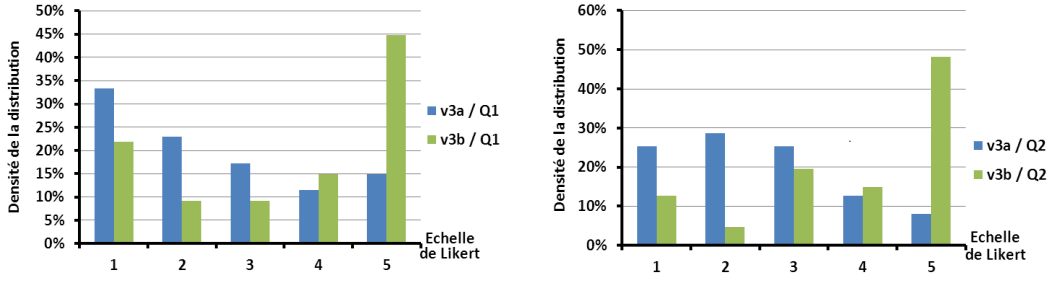
Ces résultats soutiennent **Hyp2b** mais avec une certaine réserve. Nous discuterons ces résultats dans la section 5.5.

5.4.3 Hypothèses sur la crédibilité des comportements des agents

Les hypothèses traitées dans cette section sont liées à l'hypothèse générale Hyp3, concernant le groupe de vidéos v3a, v3b et v4. La vidéo v3a correspondait à la configuration sans le mode d'interaction *Réactif*, et la vidéo v3b intègre ce mode d'interaction entre les agents et leur environnement. La vidéo v4 correspondait à une configuration avec le mode *Proactif*.

Hyp3 : *les propositions de l'environnement augmentent la crédibilité des comportements des agents.*

Les participants sont dans l'ensemble *d'accord* à *tout à fait d'accord* sur le fait que les agents n'étaient pas capables d'identifier la file d'attente qui avançait le plus rapidement ni de voir ce qui se passait autour d'eux dans la vidéo *v3b*. Au contraire, les participants sont dans l'ensemble *pas du tout d'accord* à *pas d'accord* dans la vidéo *v3a*. Les moyennes pour les questions Q1a (est-ce qu'ils sont d'accord sur le fait que les personnages n'arrivaient pas à identifier la fille d'attente qui avançait le plus rapidement ?) et Q2a (est-ce que les personnages arrivaient à voir ce qui se passait au tour d'eux ?) obtenues par la vidéo *v3a* sont très basses et celles obtenues par la vidéo *v3b* sont très élevées (voir tableau 5.13 et figure 5.9).



(a) Réponses aux questions Q1a et Q1b

(b) Réponses aux questions Q2a et Q2b

Figure 5.9 – (a)Densité de la distribution des réponses aux questions Q1a et Q1b pour les vidéos v3a et v3b (b) Densité de la distribution des réponses aux questions Q2a et Q2b pour les vidéos v3a et v3b

Les résultats sur les questions Q1 et Q2 pour les vidéos v3a et v3b sont significatifs entre eux (v3a, v3b / Q1 : t = 4.29 ; p < .0001 ; v3a, v3b / Q2 : t = 6.60 ; p < .0001). Ces résultats soutiennent **Hyp3**.

	μ	σ	Mo
vidéo v3a / Q1a	2.52	1.44	1
vidéo v3a / Q2a	2.45	1.23	2
vidéo v3b /Q1b	3.52	1.63	5
vidéo v3b /Q2b	3.82	1.41	5

TABLE 5.13 – Résultats obtenus sur les questions Q1a, Q1b, Q2a et Q2b pour les vidéos v3a et v3b

Hyp4 : *Les observateurs humains approuvent que l'environnement a proposé tous les objets possibles à l'agent et que celui-ci les a pris en compte lors de son choix de comportement.*

Les participants sont dans l'ensemble *d'accord* à *tout à fait d'accord* sur le fait que les agents avaient toutes les informations nécessaires sur les objets qu'ils pouvaient utiliser pour réaliser leurs actions, et qu'ils ont bien tenu compte de ces informations dans le choix de leur comportement dans les deux vidéos *v3a* et *v3b*. Les moyennes les plus élevées ont été obtenues par la vidéo *v3b* pour les questions Q3b (est-ce que les personnages sont capables d'identifier les autres distributeurs comme des objets leur permettant de faire leur action ?) et Q4b (est-ce que les personnages tiennent compte de l'état des autres distributeurs ?) et les plus faibles ont été obtenues par la vidéo *v3a* pour les questions Q3a et Q4a (voir les tableaux 5.14 et 5.15 et la figure 5.10).

Ces résultats sur les questions Q3 et Q4 pour les deux vidéos v3a et v3b sont significatifs entre eux (v3a, v3b / Q3 : $t = 13.75$; $p < .0001$; v3a, v3b / Q4 : $t = 12.68$; $p < .0001$). Ces résultats soutiennent **Hyp4**.

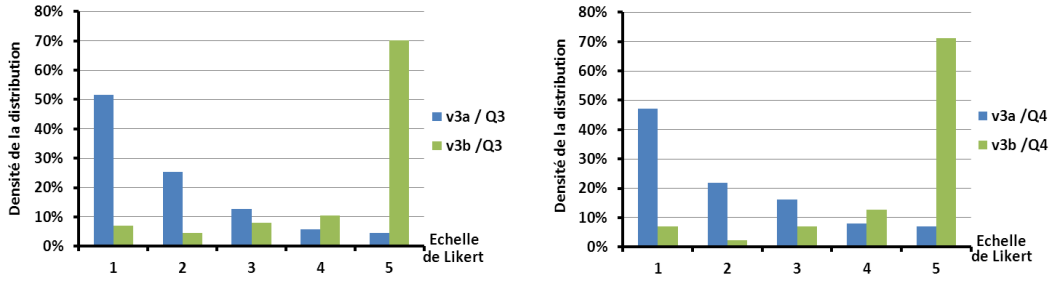
	μ	σ	Mo
vidéo v3a / Q3a	1.86	1.13	1
vidéo v3b / Q3b	4.32	1.22	5

TABLE 5.14 – Résultats obtenus sur les questions Q3a et Q3b pour les vidéos v3a et v3b

	μ	σ	Mo
vidéo v3a / Q4a	2.06	1.26	1
vidéo v3b / Q4b	4.39	1.16	5

TABLE 5.15 – Résultats obtenus sur les questions Q4a et Q4b pour les vidéos v3a et v3b

D'un autre coté, les participants sont dans l'ensemble *pas d'accord* à *pas du tout d'accord* sur le fait que les agents n'avaient pas toutes les informations nécessaires sur les objets qu'ils pouvaient utiliser pour réaliser leurs actions, et qu'ils n'ont pas tenu compte de ces informations dans le choix de leur comportement dans la vidéo v4. La moyenne la



(a) Réponses aux questions Q3a et Q3b (b) Réponses aux questions Q4a et Q4b

FIGURE 5.10 – (a)Densité de la distribution des réponses aux questions Q3a et Q3b pour les vidéos v3a et v3b (b) Densité de la distribution des réponses aux questions Q4a et Q4b pour les vidéos v3a et v3b

plus élevée a été obtenue par la vidéo v4 pour la question Q6 (est-ce que le personnage était réactif en voyant la poubelle sur son chemin vers le restaurant ?) et la plus faible a été obtenue sur la question Q2a pour la vidéo v3a (voir tableau 5.16 et figure 5.11).

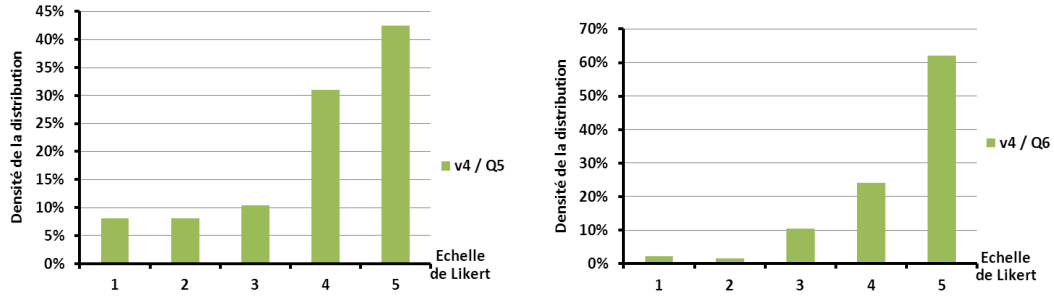
	μ	σ	Mo
vidéo v4 / Q5	3.92	1.26	5
vidéo v4 / Q6	4.43	0.90	5

TABLE 5.16 – Résultats sur les questions Q5 et Q6 pour la vidéo v4

Ces résultats sont significatifs quand ils sont comparés à une moyenne de 3 avec un test t de *Student* (v4, Q5 : t = 6.809 ; p < .0001 ; v4, Q6 : t = 14.813 ; p < .0001).

Les résultats sur la question Q2 pour les vidéos v3a et v3b sont significatifs entre eux (v3a, v3b / Q2 : t = 6.60 ; p < .0001).

Ces résultats soutiennent encore une fois l'hypothèse **Hyp4**.



(a) Réponses à la question Q5 (b) Réponses à la question Q6

FIGURE 5.11 – (a)Densité de la distribution des réponses à la question Q5 pour la vidéo v4 (b) Densité de la distribution des réponses à la question Q6 pour la vidéo v4

5.4.4 Hypothèses sur la crédibilité des interactions avec l'environnement

Nous traitons dans cette section l'hypothèse générale Hyp5, concernant le groupe de vidéos v3a, v3b et v4.

Hyp5 : *la crédibilité des comportements des agents augmente en utilisant les différents modes d'interactions avec l'environnement.*

Hyp5a : *l'utilisation du mode Réactif augmente la crédibilité des comportements des agents.*

Les participants sont dans l'ensemble *d'accord* à *tout à fait d'accord* sur le fait que le comportement des agents était crédible dans la vidéo v3a. Celle-ci a obtenue une moyenne plus élevée pour la question C3a par rapport à celle obtenue pour la vidéo v3b pour la question C3b (voir tableau 5.17 et figure 5.12).

Les résultats sur la question portant sur la crédibilité C3 pour s vidéos v3a et v3b sont significatifs entre eux ($t = 11.94$; $p < .0001$). Ces résultats soutiennent **Hyp5a**.

	μ	σ	Mo
vidéo v3a / C3a	3,98	1,22	5
vidéo v3b / C3b	1,82	1,17	1

TABLE 5.17 – Résultats sur la crédibilité des comportements des agents dans les vidéos v3a et v3b

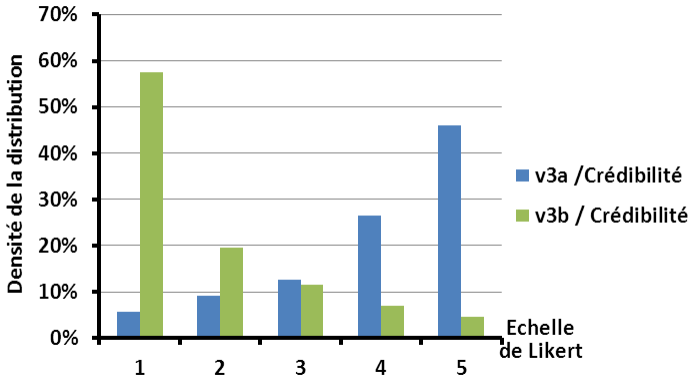


FIGURE 5.12 – Densité de la distribution des réponses sur la crédibilité des comportements des personnages pour les vidéos v3a et v3b

Hyp5b : *l'utilisation du mode Proactif augmente la crédibilité des comportements des agents.*

Les participants sont dans l'ensemble *d'accord* à *tout à fait d'accord* sur le fait que le comportement de l'agent était crédible dans la vidéo *v4*. La question C4 portant sur la

crédibilité posée sur la vidéo *v4* a obtenue une moyenne plus élevée que celle qui a été posée sur la deuxième situation pour la question Q8 (voir tableau 5.18 et figure 5.13).

Les résultats sur les questions portant sur la crédibilité C4 et Q8 sont significatifs entre eux ($t = 10.52$; $p < .0001$). Ces résultats soutiennent **Hyp5b**.

	μ	σ	Mo
vidéo v4 / C4	3,9	1,27	5
vidéo v4 / Q8	1,86	1,17	1

Table 5.18 – Résultats sur la crédibilité des comportements de l'agent dans la vidéo v4 et la seconde situation

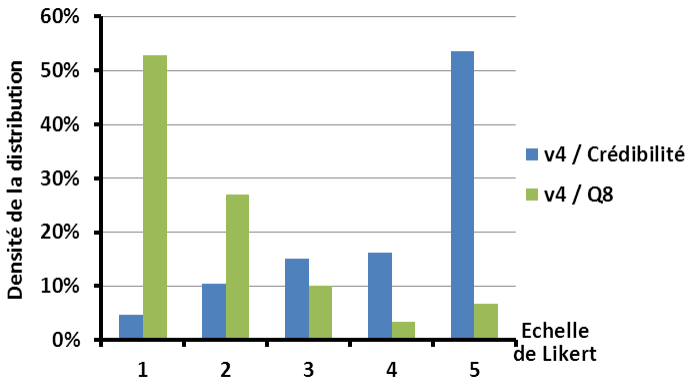


Figure 5.13 – Densité de la distribution des réponses à la question C4 et Q8 portant sur la crédibilité du comportement du personnage pour la vidéo v4 et la deuxième situation

5.5 Discussion

Une grande majorité des hypothèses que nous avons formulées sur notre modèle de représentation sémantique de l'environnement sont soutenues par les résultats obtenus par notre évaluation. Nous pouvons à présent déduire que notre modèle permet aux agents de faire des choix d'objet cohérents avec les actions qu'ils doivent réaliser (Hypothèse **Hyp1**). Cette pertinence des objets proposés aux agents est obtenue grâce aux critères de sélection pris en compte dans le calcul de la *QoS* (hypothèse **Hyp2**). D'un autre coté, notre modèle permet d'augmenter la crédibilité des comportements des agents (Hypothèse **Hyp3**) grâce aux différents modes d'interaction qu'ils peuvent avoir avec leur environnement (hypothèse **Hyp5**), mais aussi de l'efficacité et la pertinence de tous les objets possibles proposés aux agents (hypothèse **Hyp4**). Ces résultats ont été discutés et mis en rapport avec plusieurs éléments que nous abordons ci-après.

Concernant l'hypothèse **Hyp1** portant sur la pertinence des objets sélectionnés et proposés avec notre modèle, les scores obtenus pour la question portant sur l'efficacité des propositions d'objet dans toutes les vidéos nous laissent dire que notre modèle permet

de sélectionner et de proposer aux agents des objets pertinents par rapport aux actions des agents (avec des moyennes allant de **3,57** à **4,06** sur une échelle de Likert de 1 à 5). Ces résultats soutiennent majoritairement les prévisions que nous avions faites. Cela montre que l'environnement sélectionne et propose des objets qui répondent efficacement aux demandes des agents mais aussi qui sont adaptés au contexte des agents sans se limiter à des objets que nous humains pouvons considérer comme évidents (comme le banc et les toilettes publiques), mais aussi les objets qui peuvent rendre le service recherché mais qui demande plus de réflexion pour y parvenir (comme aller aux toilettes d'un restaurant).

Concernant les hypothèses **Hyp1a et Hyp1b** portant sur la crédibilité des comportements en prenant en compte ou pas respectivement les paramètres de *Difficulté* et de *Proximité* dans le calcul de la *QoS* pour sélectionner les objets, les résultats soutiennent majoritairement les prévisions que nous avions faites, à l'exception de la comparaison entre les résultats obtenus pour les vidéos v2a et v2b. En effet la moyenne obtenue pour la question portant sur la crédibilité pour la vidéo v2a était plus élevée que celle obtenue pour la vidéo v2b. Ces résultats ne contredisent pas la crédibilité du comportement de l'agent dans la vidéo v2b, car la moyenne obtenue pour cette question était supérieure à 3 (sur une échelle de Likert de 1 à 5) qui étaient significatifs en les comparant à la moyenne de 3 avec le test t de *Student*. Cela peut s'expliquer par le fait qu'il n'est pas du tout aberrant d'aller aux toilettes publiques surtout quand on connait le quartier et donc où elles se trouvent. Parmi les réponses des participants ayant été *d'accord* à *tout à fait d'accord* sur la crédibilité de la vidéo v2a, ils étaient parallèlement *d'accord* à *tout à fait d'accord* sur le fait que le personnage connaissait le quartier avec une moyenne des score de **4,19**.

En ce qui concerne l'hypothèse **Hyp2a**, les participants considèrent que le critère de *Difficulté* (représenté dans les vidéos par le prix du service s'asseoir) a été pris en compte dans le choix du personnage dans la vidéo v1b, ce qui n'était pas le cas pour la vidéo v1a. Ces résultats soutiennent les prévisions que nous avons faites, ce qui nous laisse dire qu'ils auraient eux aussi pris en compte ce critère pour faire un choix dans une telle situation.

Concernant l'hypothèse **Hyp2b**, les résultats obtenus ne contredisent pas notre hypothèse mais ne permettent pas non plus de soutenir les prévisions que nous avions faites. Notre hypothèse porte sur la prise en compte du paramètre de *Proximité* dans la sélection de l'objet qui rend au mieux le service demandé. Pour valider notre hypothèse nous avons posé plusieurs questions dont quelques unes étaient les mêmes mais posées de deux manières différentes. Le but était de vérifier si les agents tenaient compte du paramètre de *Proximité* représenté dans notre modèle par la faciliter de trouver des objets offrant un service donné. Nous avons donc comparé entre la facilité de trouver un *Restaurant* pour utiliser leur toilettes et des *Toilettes publiques*. Les résultats pour la question **Prox** (demandant aux participants s'ils étaient d'accords sur le fait que l'agent soit motivé par la proximité de l'objet qu'il a choisi d'utiliser dans chaque vidéo), pour les deux objets dans les deux vidéos ne montraient pas une grande différence entre les deux mais les résultats pour le *Restaurant* étaient plus élevés que pour celle des *Toilettes publiques*. Nous avons posé cette question différemment dans **FND** où nous avons demandé aux participants de choisir quel été l'objet le plus facile à trouver. La majorité des participants considéraient qu'il est plus facile de trouver un restaurant à **97%**. Cette différence entre les réponses aux deux questions est peut être due au fait que les participants se sont mis dans le contexte de la vidéo et donc ils avaient considéré que les toilettes publiques n'étaient pas aussi difficiles à trouver vu que pour eux elles étaient à proximité. Parmi les réponses des participants ayant été *tout à fait d'accord* sur la facilité de trouver les toilettes publiques, nous avons eu : *«l'accés était simple et rapide», «il les voit de l'autre coté de la route», «Le personnage connait peut-être déjà ces toilettes publiques», «Ils ne sont pas situés bien loin de son lieu*

de départ». Ces commentaires expliquent bien que les participants ont pris en compte la distance dans la vidéo, ces réponses s'expliquent avec les réponses à la question **CNN** (nous leur avons demandé si le choix de l'agent était lié au fait qu'il devait connaître le quartier) où la majorité des participants étaient *d'accord* à *tout à fait d'accord* sur le fait que le personnage devait connaitre le quartier où l'endroit des toilettes publiques pour s'y être rendu.

Cependant, les participants étaient majoritairement d'accord sur le fait que le personnage avait choisi les toilettes du restaurant parce qu'il n'avait pas d'autres choix. Nous avons constaté d'après les commentaires des participants qu'ils ont raisonné selon le proverbe ***«Un tiens vaut mieux que deux tu l'auras»***. Parmi les commentaires des participants nous avons *«Certainement, car elle n'allait pas marcher pendant des kilomètres sans être sûre de trouver des toilettes», «Elle n'a pas la certitude de trouver des toilettes publique plus loin», «On ne sait pas s'il sait qu'il y en a à une centaine de mètres»*. D'autres participants qui n'étaient pas d'accord considèrent que le personnage aurait pu demander *«Le personnage n'a pas cherché du tout à se renseigner à ce sujet, à savoir s'il existe des toilettes publiques à proximité.», «le personnage ne regarde pas autour pour se renseigner»*. Nous n'avons pas considéré dans notre modèle que le personnage pouvait demander à un passant. Cette piste est intéressante pour des évaluations futures qui pourraient augmenter la crédibilité des comportements des agents. Nous discuterons de cette piste dans les perspectives de cette thèse dans le chapitre 6 comme étant un autre mode d'interaction avec l'environnement (interprétation des panneaux de signalisation ou d'information et renseignement auprès d'un passant).

Concernant **Hyp3**, les résultats obtenus soutiennent les prévisions que nous avions faites. Nous pouvons dire que les propositions de l'environnement augmentent la crédibilité des comportements des agents en leur permettant de voir ce qui se passe autour d'eux en leur proposant tous les objets qui leur permettent de réaliser leurs actions en tenant en compte de leurs états. Les participants qui n'étaient pas d'accord sur les questions en relation avec l'hypothèse **Hyp3** était en raison d'un biais lié à la qualité graphique et d'animation des personnages dans les vidéos, cela même si nous avons précisé en préambule de ne pas tenir compte de ça. Parmi les commentaires laissés par les participants nous avons *«ils ne semblent pas effectuer vraiment une action dessus (bras le long du corps, sur les hanches...)», «on ne les voit pas courbés sur le guichet ou pianotant le clavier», «ils ne tournent pas la tête»*. Cependant, les autres participants qui ont tenus compte des commentaires de préambule, étaient majoritairement d'accord sur la crédibilité des comportements des agents dans la vidéo qui été produite avec la configuration complète de notre modèle.

Concernant l'hypothèse **Hyp4**, nous avons demandé aux participants si le personnage était attentif et réactif à son environnement pour vérifier que l'agent obtenait tous les objets qui l'intéressent par rapport à l'action qu'il va réaliser au moment opportun. Les résultats que nous avons obtenus soutiennent majoritairement notre hypothèse. Pour la même raison que pour l'hypothèse Hyp3, les participants qui ont émis des réponses non attendues ont pris en compte la qualité graphique et d'animation des personnages dans les vidéos.

Concernant l'hypothèse **Hyp5a**, les résultats obtenus étaient ceux attendus, ainsi, ils nous permettent de valider notre hypothèse et d'affirmer que le *mode Réactif* augmente la crédibilité des comportements des agents. Une fois encore les résultats non attendus

étaient causé par le biais associant le terme Crédibilité à la qualité des animations et le rendu graphique des vidéos. Nous pouvons trouver des commentaires comme *«ils semblent glisser sur le sol. Leurs pas ne correspondent pas à leurs mouvements de jambes», «les gens restent trop longtemps devant le distributeur certains personnages sont habillés exactement pareil», «La position de leurs mains ne montre pas l'intention de retirer de l'argent»*.

Et enfin, concernant l'hypothèse **Hyp5b**, les résultats obtenus soutiennent majoritairement notre hypothèse portant sur l'augmentation de la crédibilité des comportements des agents en utilisant le *mode Proactif*. Les participants dont les réponses n'étaient pas attendues semblaient juger le choix d'action du personnage. Ils considèrent que l'agent n'aurait pas du prendre de reçu au distributeur si c'était pour le jeter après. Parmi les réponses des participants nous avons *1- on ne jette pas un reçu de retrait, cela comporte des informations confidentielles. Quand on ne veut pas de reçu, on coche la case NON. 2 - de toute façon il pouvait le jeter au restaurant», «s'il souhaite jeter son reçu dans ce cas il ne le prend pas», «Si c'est pour jeter un reçu autant le refuser au DAB..»*. D'autre part, les participants considèrent que si le personnages choisit de faire un détour pour jeter son reçu c'est parce qu'il n'y avait pas d'autres poubelles qui lui permettaient de le faire, mais aussi ils portent un jugement morale sur l'action du personnage en considérant que c'est un geste civique. Parmi ces réponses nous citons *«Parce que le personnage est manifestement une personne responsable et éco-citoyenne», «S'il désire jeter son papier dans une poubelle et que c'est la poubelle la plus proche...», «oui cela montre quelqu'un qui fait preuve de civisme en voulant jeter son papier dans une poubelle»*.

5.6 Conclusion

Nous avons présenté dans cette section l'évaluation de notre modèle que nous avons réalisé afin de valider les différentes hypothèses que nous avons posées. Notre évaluation est basée sur les observations de participants de divers domaines balayant une grande tranche d'âge et de connaissances variées sur les environnements virtuels.

Avant de lancer notre évaluation à grande échelle, nous avons d'abord fait une pré-évaluation en interne mais aussi auprès de la famille et des amis afin d'ajuster le contenu des vidéos, les commentaires et le type de questions que nous allions poser, mais aussi pour valider notre protocole expérimental. Cela nous a permis de nous assurer que les observateurs ont bien compris le contenu des vidéos afin qu'ils se concentrent sur les questions que nous leur avons posées.

Une fois notre modèle expérimental corrigé et validé à petite échelle, nous avons lancé notre évaluation sur un plus grand nombre de participants anonymes et n'ayant aucune connaissances sur le projet Terra Dynamica. Les résultats obtenus nous ont permis de valider nos hypothèses et déduire que notre modèle permet d'augmenter la crédibilité des comportements des agents en leur proposant les objets leur permettant de réaliser leur action au moment opportun en se basant sur les paramètres que nous avons défini pour calculer la QoS qui nous permet de classer les objets à proposer aux agents selon leur pertinence.

Nous avons proposé aux participants de justifier leurs réponses, ce qui nous a permis de comprendre leur choix de réponse. Leurs commentaires nous ont conduit vers une nouvelle perspective qui consiste à introduire un mode de communication entre les agents afin d'avoir des renseignements.

Notre évaluation était suffisante pour valider la pertinence de notre modèle sur l'aspect de la crédibilité. Cependant, il aurait été intéressant d'enrichir cette évaluation en

comparant le choix d'objet des agents proposé par notre module avec une configuration complète de notre modèle avec les choix d'objets des avatars contrôlés directement par les participants. Cette comparaison nous permettrait de renforcer la validation de l'hypothèse **Hyp2**.

Chapitre 6

Conclusion

Nous avons proposé dans cette thèse un modèle de représentation de la sémantique de l'environnement virtuel qui nous permet de déduire des informations d'aide à la décision, permettant ainsi aux agents d'exhiber des comportements crédibles. Ce travail s'inscrit dans le cadre d'un projet industriel *Terra Dynamica*, dont les domaines applicatifs sont variés comprenant la sécurité, l'urbanisme, le transport et les jeux vidéo. Le modèle que nous proposons offre aux agents la possibilité de choisir le meilleur objet de l'environnement adapté à leur contexte pour réaliser au mieux leurs actions. Il permet aussi aux agents d'être réactifs et attentifs à ce qui se passe dans leur environnement grâce aux différents types de notifications qu'ils reçoivent de leur environnement via les modes d'interactions que nous avons proposé.

La crédibilité d'une simulation dans un environnement virtuel passe par la crédibilité comportementale de ses personnages. Pour qu'un comportement soit crédible, il doit être cohérent et consistant par rapport à l'environnement. En effet, la prise de décision des agents est faite à partir de la représentation de l'environnement. Pour cela, les agents doivent avoir une bonne représentation de leur environnement. La plupart des modèles de représentation de la sémantique de l'environnement offrent à l'architecture décisionnelle des connaissances sémantiques leur permettant de sélectionner les objets de l'environnement qui offrent la fonction désirée pour l'exécution de l'action. Cependant, les différentes solutions déjà proposées ne sont pas suffisantes pour permettre de comparer entre les différents services proposés par les objets de l'environnement. Une telle comparaison devrait permettre de choisir le meilleur objet capable de proposer le service recherché.

Nous avons proposé dans cette thèse un modèle de représentation de l'environnement qui fournit, en plus des informations de perception aux agents, d'autres informations pour l'aide à la décision qui portent sur l'utilisation des objets de l'environnement. Nous avons proposé de représenter les services proposés par les objets de l'environnement en s'inspirant des techniques utilisées dans le domaine du Web sémantique. Pour ce faire, nous avons représenté les différents services de l'environnement dans une ontologie de services, où nous avons défini les différents types de services correspondant aux actions possibles dans l'environnement. Nous avons défini les paramètres non fonctionnels des services de l'environnement qui nous ont permis de calculer une qualité de service (QoS) proposée par un objet par rapport à une action donnée. Cette QoS nous permet de trier les différents objets qui permettent de réaliser la même action et de proposer ainsi l'objet présentant la meilleure QoS.

La capacité de ce modèle de proposer les meilleures objets pour réaliser les actions des agents pour avoir des comportements crédibles a été vérifiée avec succès dans notre évaluation. Nous avons fait une évaluation subjective conduite auprès de participants qui

devaient donner leur avis sur le caractère crédible des comportements des personnages produits avec notre modèle de représentation sémantique de l'environnement. Pour ce faire, nous avons proposé un protocole d'évaluation qui est à notre connaissance le premier protocole d'évaluation dédié pour évaluer la crédibilité comportementale des agents engendrée par un modèle de représentation sémantique de l'environnement. Les résultats positifs obtenus lors de notre évaluation subjective valident les capacités de notre modèle à proposer des objets similaires à ceux qu'auraient choisis des êtres humains dans la même situation, permettant ainsi aux agents d'avoir des comportements crédibles aux yeux des observateurs humains. Ils montrent aussi que notre modèle permet aux agents d'avoir des comportements opportunistes et réactifs aux événements qui se produisent dans leur environnement. Cependant, nous n'avons pas pu valider tous les paramètres d'évaluation qui nous ont permis de classer les objets entre eux. En effet, même si notre modèle a été validé globalement pour avoir permis d'avoir des comportements crédibles, nous n'avons pas pu valider le paramètre *"Proximité"*. Les tests statistiques que nous avons réalisé n'ont pas pu montrer que ce paramètre était un paramètre décisif pour les agents dans leurs choix. Cependant, nous avons remarqué que certaines réponses étaient biaisées à cause de la qualité graphique des vidéos. Nous prévoyons donc de refaire l'évaluation de ce paramètre. Nous détaillerons cette perspective dans la section suivante.

Notre projet de simulation prévoit de mettre en scène un grand nombre d'agent à la hauteur du nombre d'habitants d'une ville. Nous avons donc fait une évaluation qualitative qui nous a permis de vérifier avec succès la capacité de notre modèle pour passer à l'échelle pour un grand nombre d'agents.

Perspectives

Plusieurs perspectives découlent de notre travail que nous avons présenté dans cette thèse. Nous présentons quelques perspectives de travaux à court, moyen et long terme qui peuvent être développées.

Perspectives à court terme

Évaluation du modèle

Nous pensons qu'il est possible, dans un premier temps, de reprendre l'évaluation du paramètre "Proximité" pour le calcul de la QoS qui n'a pas été vérifié avec la première évaluation subjective que nous avons réalisée. Effectivement, certains participants n'avaient pas jugé que le choix des personnages était dû au paramètre de proximité tel que nous l'avons défini, c'est-à-dire, la faciliter de trouver l'objet qui propose un service donné dû au fait de son abondance dans l'environnement comme par exemple des boites postales, ou bien des restaurants, etc. Néanmoins, ils ont jugé que le comportement des agents étaient crédibles. Nous prévoyons alors de refaire les vidéos avec un autre contexte et un autre scénario qui mettra plus en évidence l'aspect que nous voulons évaluer et qui évitera le problème d'incompréhension ou l'induction en erreur à cause de la qualité graphique de la simulation. Nous rajouterons d'autres questions pour lever l'ambiguïté et qui focaliseront l'attention des participants sur le paramètre que nous voulons évaluer.

De même, nous souhaitons reprendre les résultats de notre première évaluation en excluant les réponses des participants qui n'ont pas bien compris le but de notre évaluation (par exemple ceux qui ont tenu compte de la qualité de l'animation graphique). Nous verrons ainsi si les résultats obtenus sont plus satisfaisants ou pas par rapport à ce que nous avons obtenu avec toutes les réponses des participants. A première vue, et en analysant rapidement les différentes réponses que nous avons obtenus avec les participants qui n'ont pas répondu aux questions ouvertes (donc nous ne pouvons pas savoir s'ils ont réellement compris ce que nous voulons évaluer) ces résultats devraient conforter nos résultats de départ.

Il serait aussi intéressant d'évaluer et de valider la crédibilité des choix des agents avec un autre protocole d'évaluation. Cela consiste à utiliser un avatar dans la simulation qui sera guidé par les participants. Les participants auront à suivre un plan qu'il leur sera proposé, ils devront alors choisir les objets de l'environnement qui vont leur permettre d'exécuter les actions du plan. Nous pourrions alors comparer les objets choisis par les participants avec ceux qui ont été proposés à notre agent pour exécuter le même plan avec notre modèle. Nous proposerions par la suite aux participants de visualiser les comportements de notre agent pour l'exécution du même plan. Nous observerions les participants

pendant leur visualisation des comportements de notre agent face aux mêmes situations auxquelles ils ont été confrontés pour exécuter leur plan. Nous terminerions par poser des questions aux participants sur leurs choix et expliquer la similarité ou la divergence de leurs choix avec ceux de notre agents. Cette autre approche d'évaluation ne remet pas en cause notre premier choix d'évaluation et le protocole que nous avons proposé pour le faire. Elle nous permettrait de valider notre modèle différemment, elle mettrait ainsi les observateurs en situation participative [Pirard, 2010] alors qu'ils étaient seulement juges dans l'évaluation actuelle. Cependant, cette approche d'évaluation est plus coûteuse en temps et risque d'être biaisée par la subjectivité des participants. En effet, il se peut que quelque participants ne répondent pas en exprimant tout ce qu'ils pensent. Il est donc important d'adapter le questionnaire selon chaque situation, il est aussi important de bien choisir les questions à poser de sorte à ne pas influencer les participants dans leurs réponses. Ce travail de validation par questionnaire se fait dans le domaine des *Sciences de l'Homme et de la Société* et nécessite beaucoup de temps [Blatier, 1999].

Nous souhaitons aussi à court terme évaluer notre modèle sur un autre aspect portant sur son adaptabilité et sa portabilité en l'intégrant dans une autre architecture d'agent. En effet, notre modèle de représentation de la sémantique de l'environnement a été intégré et testé sur une seule plateforme qui est celle du projet Terra Dynamica qui a son propre architecture décisionnelle telle qu'elle a été définie dans [De Sevin et al., 2012]. Il serait donc intéressant de tester sa capacité d'adaptation sur un autre type d'architecture. Nous voulons donc comparer avec d'autres travaux de simulation réaliste qui utilisent une architecture différente de celle que nous avons utilisé dans le projet Terra Dynamica pour la sélection d'objet. Comme par exemple un modèle de simulation utilisant des agents BDI [1] qui se basent sur les connaissances des agents sur leur environnement et leurs perceptions [Van Oijen and Dignum, 2011].

Nous prévoyons aussi de mettre à l'épreuve notre modèle dans un autre type d'application correspondant à d'autres types d'environnement virtuel (comme les environnements virtuels informés pour l'apprentissage [Edward et al., 2010], [Chevaillier et al., 2009], [Querrec, 2002]), ce qui nous permettra de valider la portabilité de notre modèle de représentation de l'environnement et sa capacité d'adaptation sur des domaines d'application différents.

Affiner la description des objets de l'environnement

Notre modèle de représentation de l'environnement pourrait connaître certaines améliorations concernant le niveau de détail pour la description des objets de l'environnement, qui nous permettra par conséquent d'affiner notre processus de sélection des instances d'objets de l'environnement que nous proposons aux agents. Actuellement, notre modèle ne fait pas la différence entre deux objets appartenant au même type d'objet comme par exemple, deux fast-foods un qui appartient à l'enseigne Quick et l'autre au Mac Donald's. Cette différence peut être dans le prix ou bien dans les menus. Ce type de détail dans la description des instances d'objets nous permettra par exemple de proposer à un agent qui n'a pas beaucoup d'argent le fast-food qui propose le menu le moins cher. Pour ce faire, notre module devra avoir un accès aux profils des agents pour pouvoir adapter ses propositions en fonction de chaque agent et déduire ses préférences individuelles. Dans ce cas, le processus de sélection des instances d'objets tiendra compte en plus des paramètres contextuels (la distance, de la disponibilité et de l'accessibilité, etc) des propriétés

1. Belief-Desire-Intention

de chaque instance d'un type d'objet. Le processus de sélection du type d'objet offrant la meilleure *QoS* restera le même, c'est le choix de l'instance d'objet appartenant au même type d'objet sélectionné avec la meilleure *QoS* qui change. Dans ce cas, il faut tenir compte du temps de traitement pour pouvoir assurer la qualité de notre modèle sur l'aspect du passage à l'échelle.

Perspectives à moyen terme

Obtention des connaissances sur l'environnement d'un autre agent

Les interactions entre un agent et son environnement peuvent êtres plus ou moins fortes selon la façon dont l'agent acquiert l'information en provenance de son environnement [Guessoum, 1996]. En effet, il y a deux façons de le faire, soit à travers un autre agent en communiquant avec lui, ou bien directement en interagissant avec son environnement. Nous avons fait le constat en lisant les commentaires des participants lors de l'évaluation subjective de notre modèle que certains participants ont soulevé le manque de communication entre les agents qui auraient pu renseigner d'autres agents ou acquérir de l'information en communicant entre eux. Effectivement, la communication et les comportements sociaux entre les agents donnent de la crédibilité à leurs comportements [Bates, 1994].

Le projet Terra Dynamica n'était pas orienté sur la communication verbale entre les agents, nous nous sommes donc dispensés d'obtenir des connaissances sur l'environnement à travers d'autres agents. Toutefois, si nous souhaitons faire en sorte que les agents soient capables de communiquer et d'échanger des connaissances entre eux, nous devons alors réfléchir à une solution qui mémorise les connaissances des agents au cours de la simulation, ainsi nous serons capables de dire si un agent a la capacité de répondre à la question sur l'environnement qui lui a été posée par un autre agent. Nous pouvons aussi aller plus loin dans le détail en mémorisant le parcours des agents, ainsi nous pourrons dire s'il a pu passer à proximité de l'objet que l'autre agent recherche et mettre une probabilité sur le fait qu'il ait fait attention pour percevoir l'objet recherché en passant devant lui ou pas. L'agent qui doit interroger un autre doit aussi faire le bon choix d'agent pour avoir une chance d'avoir une réponse à sa question. Ainsi il ne va pas interroger un agent qui vient de sortir d'une bouche de métro car ses chances d'avoir une réponse seront réduites.

Interprétation des éléments informatifs de l'environnement

L'environnement d'une ville contient de nombreux objets porteurs d'information dédiée pour guider et renseigner les personnages qui le peuplent. Nous pouvons trouver des plans de la ville, les plaques de signalisation, le menu d'un restaurant, etc. Tous ces objets informatifs sont considérés comme une source d'information que nous utilisons au quotidien pour nous informer. Nous pouvons ainsi adapter les propositions de l'environnement en fonction de la disponibilité et de la proximité des ressources d'information dans la ville. Ainsi, nous pouvons proposer à un agent d'aller chercher l'information qu'il recherche sur le plan du métro par exemple pour s'orienter, ou bien considérer qu'il est assez proche d'un panneau de signalisation pour attirer son attention vers lui (lui demander de scruter son environnement afin d'obtenir de l'information) lui indiquant qu'il peut se rendre à la poste à pied en prenant une direction donnée.

Évidemment, les agents ne sont pas capables d'interpréter tous seuls les informations proposés par les objets, ils l'auront en réponse à leur première requête, mais ils devront rajouter une nouvelle action à leur plan (aller vers le panneau d'information).

Proposition de comportements au module de décision

Actuellement, notre modèle propose uniquement les meilleurs objets qui permettent aux agents d'avoir des comportements crédibles. Notre modèle a la liberté de proposer des objets variés pour répondre à une requête du module de décision. Par exemple, si l'agent veut se divertir et que le module de décision ne précise pas une action précise pour le faire, alors notre module va lui proposer des objets comme un théâtre, un cinéma, une salle de concert, etc après tout dépendra de l'endroit où se trouvera l'agent.

Nous voulons que notre module de l'environnement soit plus entreprenant pour proposer des comportements en réaction à des événements et qui peuvent opérer dans l'environnement qui n'ont pas été pris en compte lors de la planification des actions des agents. Par exemple, un food truck vient s'installer sur une place. Dès qu'il ouvre l'environnement va alors changer d'état et les agents auront à leur disposition un nouveau service. La présence d'un food truck peut susciter l'envie de manger pour certains agents. Nous pouvons alors proposer au module de décision des agents qui se trouvent à proximité d'aller manger un morceau.

Perspectives à long terme

Faire de l'apprentissage

Nous n'exploitons pas pour le moment la dynamique de l'environnement, c'est-à-dire, les déplacements des agents et les flux de groupes dans l'environnement. Nous considérons qu'il serait intéressant de combiner notre modèle à un modèle d'apprentissage qui permet de déterminer l'affluence de foule dans certains endroits selon une heure donnée de la journée. Par exemple, une rue commerçante avec des restaurants aura une grande affluence pendant la pause déjeuner des travailleurs. L'apprentissage nous permettra de déterminer les endroits à éviter à une période donnée. Nous pouvons ainsi combiner cette information avec la *QoS* pour proposer une instance d'objet selon l'endroit où se trouvera l'agent et le moment de la journée (heure de pointe ou creuse), mais aussi de la météo (beau temps ou pluvieux). Cette information pourrait compléter notre modèle de sélection d'instance d'objet que nous allons proposer et l'affiner. Par exemple, si un agent veut retirer de l'argent et que la *QoS* d'un distributeur de billet ait la plus haute valeur, le choix de l'instance pour l'exécution de l'action sera important car si nous proposons un distributeur de billet se trouvant dans un endroit très fréquenté, il y aura une grande probabilité que cet instance soit fréquemment sollicitée, de ce fait l'agent risquera de devoir attendre avant d'accéder au service de retirer de l'argent. Tel est le cas par exemple d'un distributeur qui se trouve à proximité d'une bouche de métro. Dans ce cas, il serait préférable d'aller plus loin et d'utiliser une autre instance de distributeur de billets.

Cette solution serait très intéressante pour les projets d'aménagement de la ville par exemple. Ainsi, nous pourrions observer les mouvements des personnages de l'environnement et décider des meilleurs aménagements de la ville.

Intégrer notre modèle dans un agent conversationnel

Une autre perspective pour l'application de notre modèle serait de développer un agent conversationnel qui peut répondre aux questions sur la ville. Cela était l'une des tâches du projet Terra Dynamica qui a été réalisée par le partenaire DAVI [2]. La différence avec ce que nous proposons réside dans la manière de traiter les requêtes des utilisateurs et

2. http ://www.davi.fr/

le traitement des réponses. En effet, DAVI a utilisé son propre modèle de traitement des requêtes et utilise la base de connaissances du système du projet (Base de données du projet) pour répondre aux questions des utilisateurs. L'utilisation de notre modèle de représentation des connaissances de l'environnement dans un agent conversationnel nous permettrait de faire des raisonnements de haut niveau. Ainsi, nous serons capables par exemple de répondre à des touristes sur les différents endroits qui pourraient les intéresser. Ces derniers pourraient exprimer l'action qu'ils veulent faire, par exemple aller manger, notre modèle se chargera de leur proposer des restaurants. Nous pourrions aussi imaginer que notre modèle poserait des questions pour leur demander leurs préférences ou bien plus de détails sur ce qu'ils voudraient faire exactement pour adapter les propositions.

Annexe A

Questionnaire de l'évaluation subjective

Nous présenterons dans cette annexe les captures d'écran des pages de notre site web d'évaluation subjective, qui nous a permis de réaliser les expérimentations dans le chapitre 5.

FIGURE A.1 – Page d'introduction

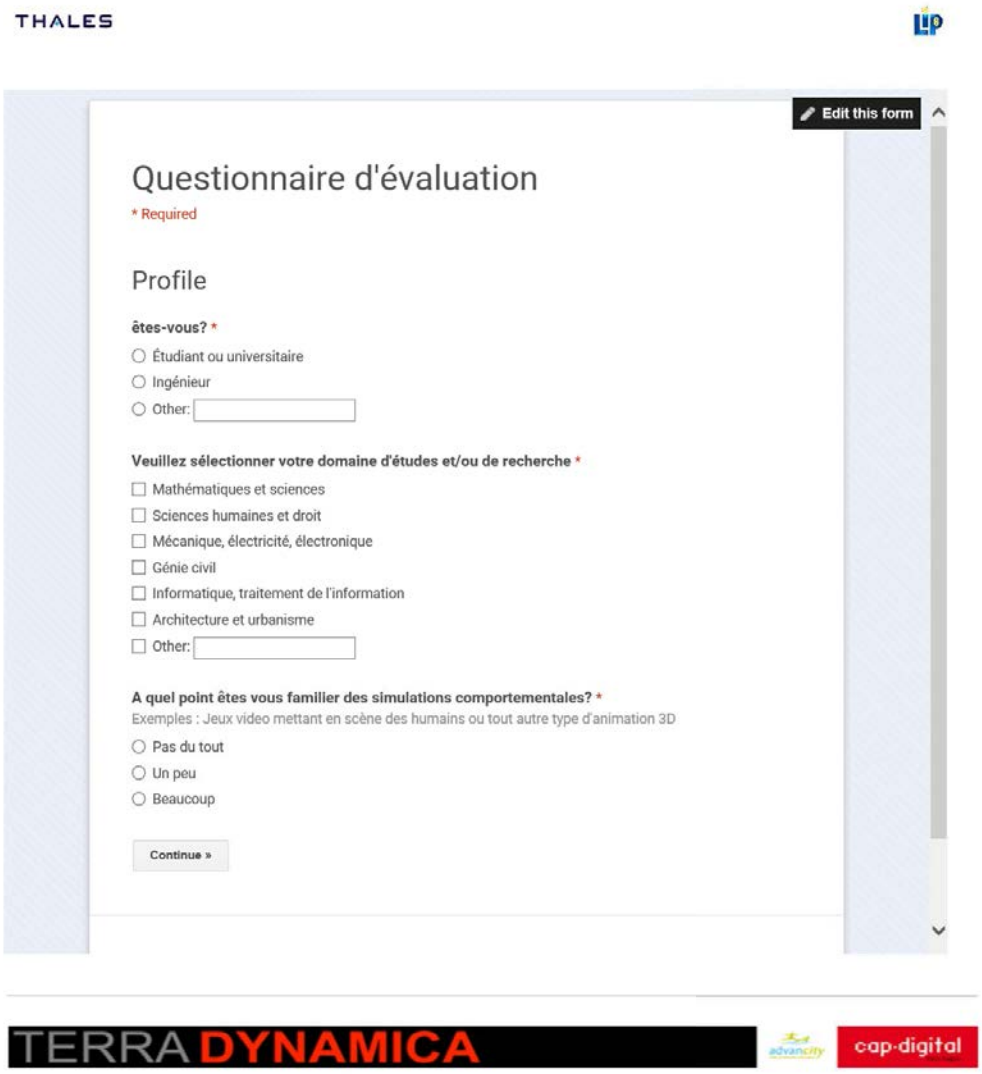


FIGURE A.2 – Page d'informations sur les participants

FIGURE A.3 – Début de la page d'évaluation du critère "Difficulté"

FIGURE A.4 – Début de la page d'évaluation du critère "Proximité"

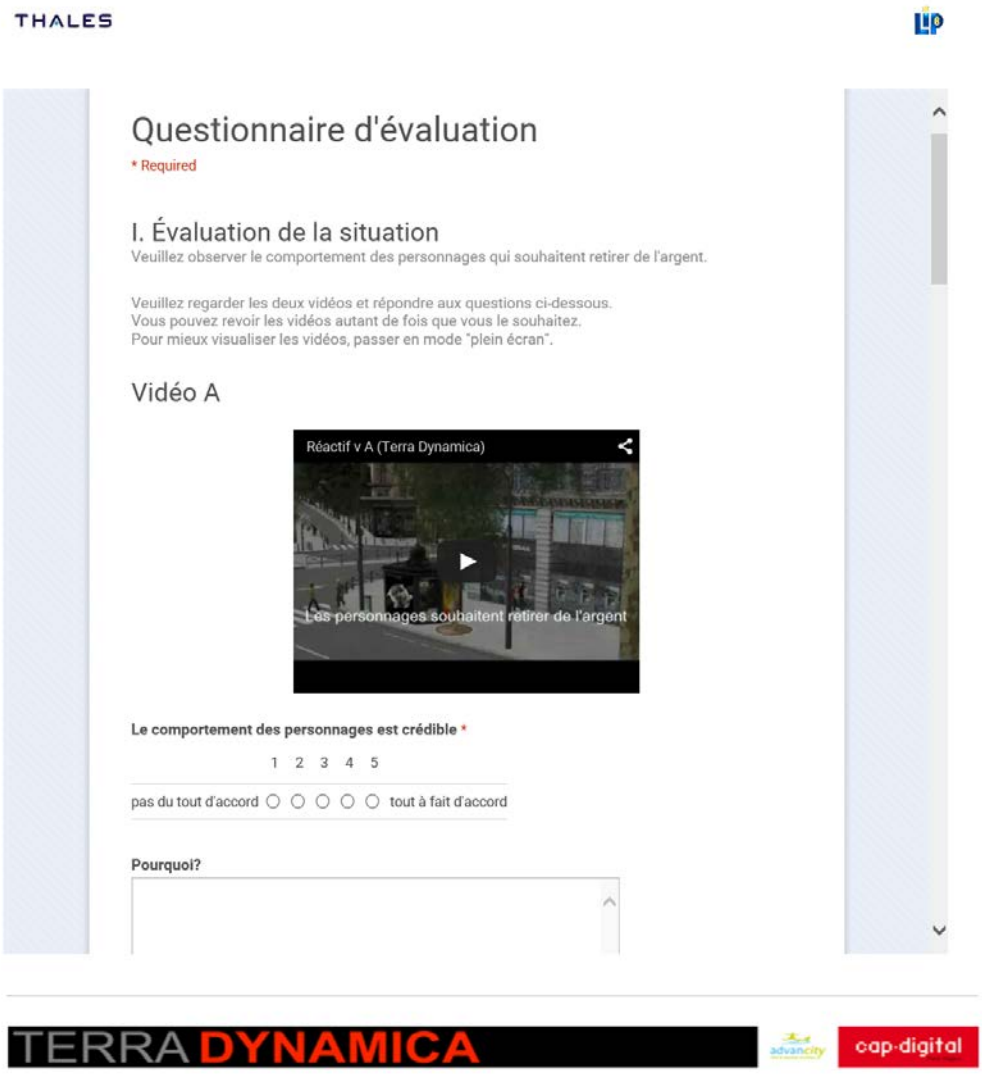


FIGURE A.5 – Début de la page d'évaluation du mode "Réactif"

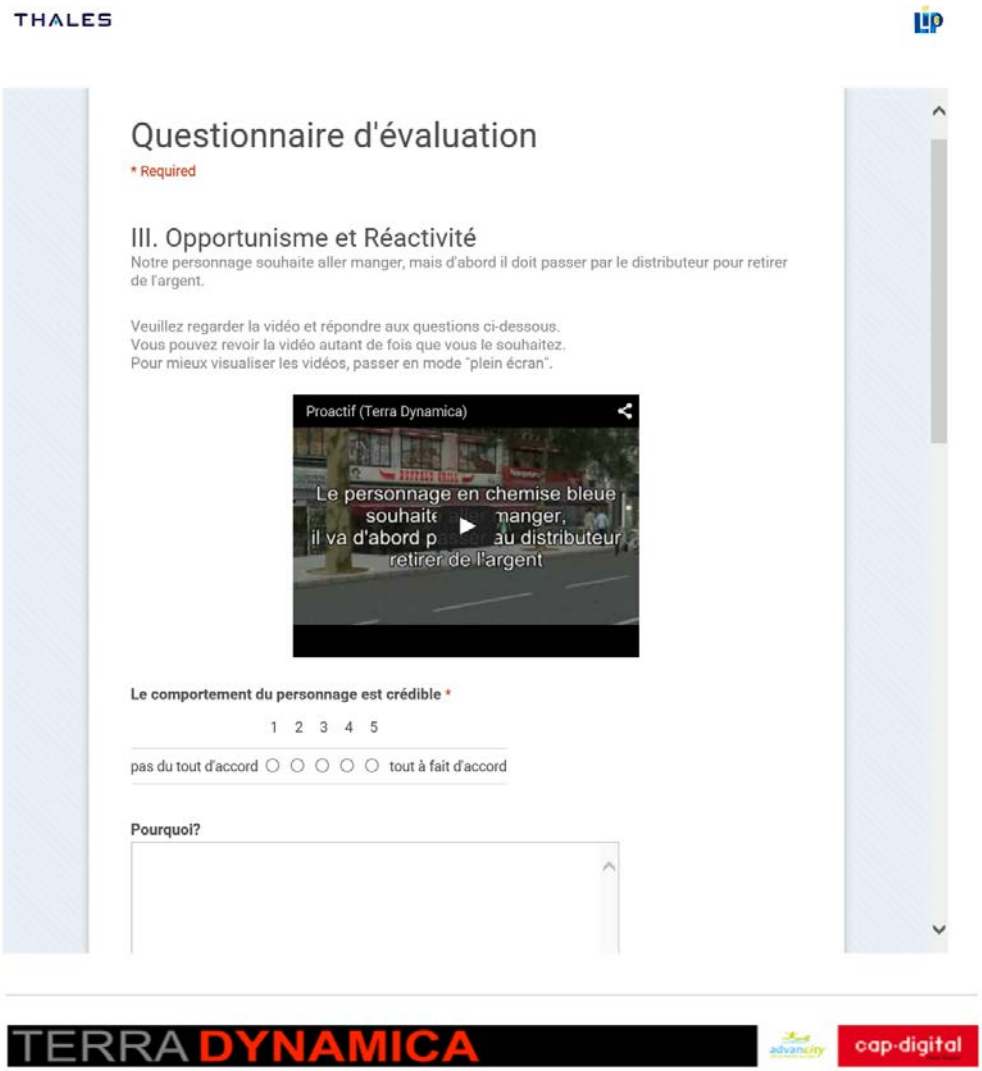


FIGURE A.6 – Début de la page d'évaluation du mode "Proactif"

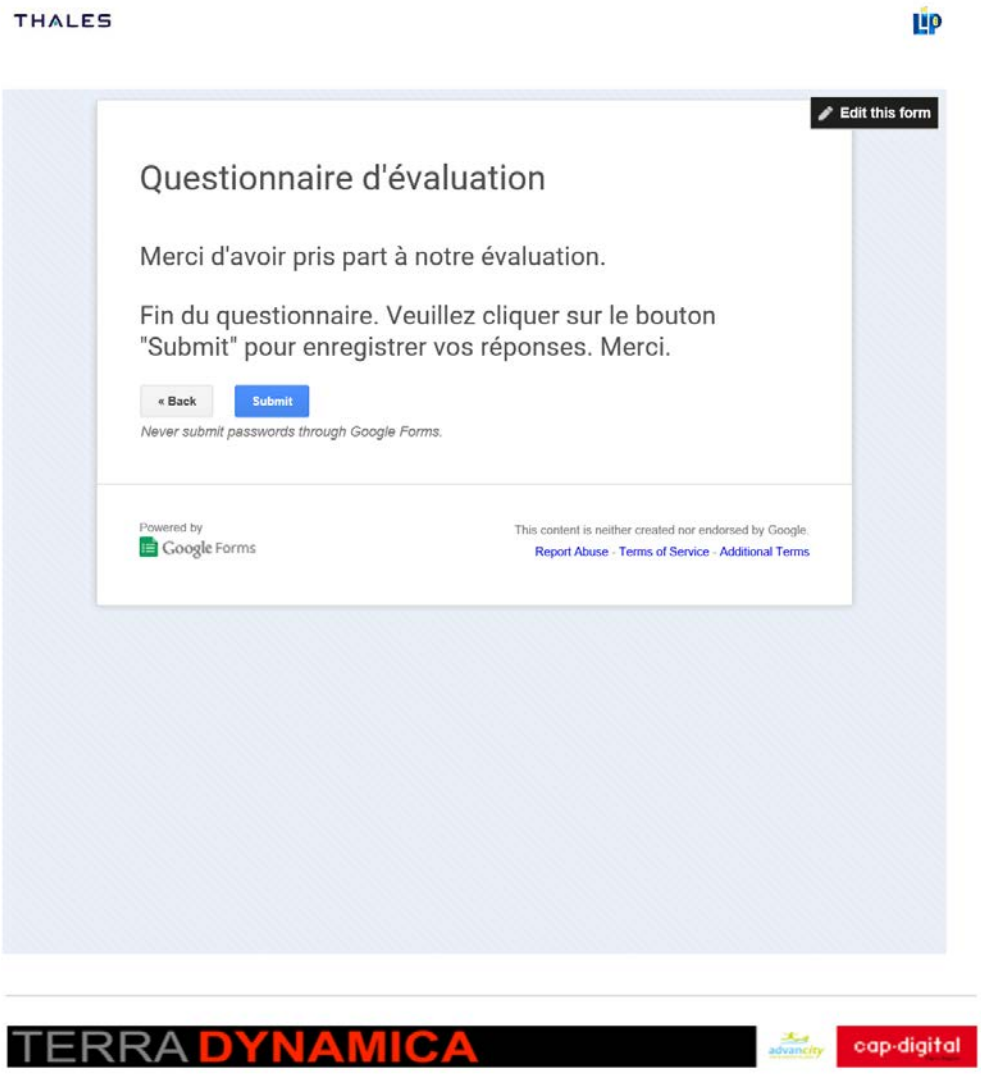


FIGURE A.7 – Page de fin du questionnaire

Table des figures

2.1 Méta-modèle pour la modélisation des concepts du domaine dans Mascaret [Chevaillier et al., 2009] . . . 21
2.2 Lien entre les différentes ontologies utilisées dans les travaux de [De Troyer et al., 2003] . . . 25
2.3 Processus conceptuel VR-Wise proposé dans [Pellens et al., 2005] (de gauche à droite) : étape de spécification, étape de mapping et étape de génération . 25
2.4 Modèle sémantique XML-based de l'environnement virtuel autour de l'*Item Digital*, proposé dans [Gutierrez et al., 2005] . . . 27
2.5 Architecture de l'EV sémantique basée sur les ontologies proposée dans [Grimaldo et al., 2006a] . . . 28
2.6 Exemple de l'ontologie de domaine utilisée dans [Grimaldo et al., 2008] . . . 29
2.7 Graphe OWL décrivant une scène dans [Kalogerakis et al., 2006] . . . 31
2.8 Architecture du framework à trois couches proposé dans [Chang et al., 2005] 33
2.9 Composants de la plateforme ISReal proposée dans [Kapahnke et al., 2010] 34
2.10 Exemple d'annotation sémantique d'un objet graphique XML3D avec RDFa. 35
2.11 Architecture du modèle sémantique de l'environnement virtuel de la plateforme ISReal proposée dans [Kapahnke et al., 2010] . . . 35
2.12 Classe ServiceProfile de l'ontologie OWL-S . . . 40

3.1 Interactions du module sémantique de l'EV avec le module décisionnel des agents : le module décisionnel envoie des requêtes pour trouver un objet qui propose le service recherché, et le module sémantique lui renvoie une liste d'objets triés selon la QoS qu'ils proposent. . . . 51
3.2 Ontologie de services : une couche méta contenant les concepts génériques du modèle, et une couche sémantique contenant les concepts de l'environnement et les relations qui existent entre eux. . . . 53
3.3 Concepts généraux de l'ontologie de services «Niveau méta». . . . 54
3.4 Hiérarchie de Services : Services de comportement S_c et Services d'action S_a. 55
3.5 Ontologie de services : exemple du service «Manger» . . . 56
3.6 Ontologie de services : Les différents critères d'évaluation appliqués au service «Manger» . . . 57

4.1 Une partie de l'Ontologie de services : Service postal . . . 69
4.2 (a) Le module de décision demande s'il peut traverser (b) Le module sémantique notifie à l'agent la proximité d'une boite aux lettres . . . 69
4.3 Schéma de la structure globale du Projet Terra Dynamica . . . 71
4.4 Composantes du MAC et leur interdépendance . . . 72
4.5 Organisation générale des modules du projet Terra Dynamica . . . 73

4.6 Diagramme présentant les interactions du module sémantique de l'environnement avec les autres modules du projet 74
4.7 Interactions entre le module sémantique et l'IA 77
4.8 Représentation des objets et lieux de l'environnement 82
4.9 Le diagramme de classes UML de la description d'un objet sémantique . . . 83
4.10 Temps de réponse moyens aux requêtes du module d'IA par nombre d'agents 88
4.11 Temps de réponse moyen pour 10 000 services et 2 500 agents 88
4.12 Temps de réponse moyen pour les requêtes envoyées en *Mode Réactif* 89
4.13 Temps de réponse moyen pour les requêtes envoyées en *Mode Pro-actif* . . . 90

5.1 (a) L'agent va s'asseoir sur un banc public (b) L'agent va s'asseoir dans un bar café 99
5.2 (a) L'agent va dans un restaurant (b) L'agent va dans des toilettes publiques 100
5.3 (a) Les agents se dirigent vers les distributeurs libres (b) Les agents font la queue derrière le même distributeur 102
5.4 Vidéo v4 (mode Proactif) : L'agent se dirige vers une poubelle après avoir retiré de l'argent du distributeur, puis il se rend au restaurant 103
5.5 Moyenne et écart-type des réponses à la question Eff par vidéo (échelle de Likert de 1 à 5) 111
5.6 Densité de la distribution des réponses à la question Diff pour les vidéos v1a et v1b 113
5.7 Densité de la distribution des réponses à la question Prox pour les vidéos v2a et v2b 114
5.8 Répartition des réponses à la question Fnd 114
5.9 (a)Densité de la distribution des réponses aux questions Q1a et Q1b pour les vidéos v3a et v3b (b) Densité de la distribution des réponses aux questions Q2a et Q2b pour les vidéos v3a et v3b 115
5.10 (a)Densité de la distribution des réponses aux questions Q3a et Q3b pour les vidéos v3a et v3b (b) Densité de la distribution des réponses aux questions Q4a et Q4b pour les vidéos v3a et v3b 117
5.11 (a)Densité de la distribution des réponses à la question Q5 pour la vidéo v4 (b) Densité de la distribution des réponses à la question Q6 pour la vidéo v4 117
5.12 Densité de la distribution des réponses sur la crédibilité des comportements des personnages pour les vidéos v3a et v3b 118
5.13 Densité de la distribution des réponses à la question C4 et Q8 portant sur la crédibilité du comportement du personnage pour la vidéo v4 et la deuxième situation 119

A.1 Page d'introduction 134
A.2 Page d'informations sur les participants 135
A.3 Début de la page d'évaluation du critère "Difficulté" 136
A.4 Début de la page d'évaluation du critère "Proximité" 137
A.5 Début de la page d'évaluation du mode "Réactif" 138
A.6 Début de la page d'évaluation du mode "Proactif" 139
A.7 Page de fin du questionnaire 140

Liste des tableaux

2.1 Récapitulatif des différentes approches de modélisation de la sémantique des EVs dans quatre critères : représentation de la sémantique du contenu de l'EV, indépendance du domaine d'application, découplage de la représentation graphique et résultant d'une phase de modélisation conceptuelle. . 37
2.2 Les différentes catégories d'évaluation et leur critères de satisfaction [Isbister and Doyle, 2002] 46

4.1 Description des méthodes de IPerceptionAPI 75
4.2 Description des méthodes de IPerceptor 75
4.3 Description des méthodes de ISemanticWorldAPI 76
4.4 Description de la méthode iModeInterrogatif 78
4.5 Description des méthodes de l'interface SemObs 80
4.6 Description des méthodes d'enregistrement et de dés-enregistrement d'un agent en mode Proactif 80
4.7 Description des principales méthodes de SemObject 84
4.8 Description des principales méthodes de SemObjUsert 84
4.9 Tableau représentant le nombre d'agents maximal avant ralentissement par frame de 50 ms 91

5.1 Questions sur la **Vidéo v1a :** *QoS* avec le paramètre "Difficulté" 100
5.2 Questions sur la **Vidéo v1b :** *QoS* sans le paramètre "Difficulté" 100
5.3 Questions sur la **Vidéo v2a :** *QoS* avec le paramètre "Proximité" 101
5.4 Questions sur la **Vidéo v2b :** *QoS* sans le paramètre "Proximité" 101
5.5 Questions sur les **Vidéos v3a et v3b :** Avec et sans le mode "Réactif" . . . 102
5.6 Questions sur la **Vidéo v4 :** Avec le mode "Proactif" 103
5.7 Résultats obtenus sur la question portant sur le paramètre *d'Efficacité* pour les vidéos v1a, v1b, v2a et v2b 110
5.8 Résultats sur la crédibilité pour les vidéos v1a, v1b 111
5.9 Résultats sur la crédibilité pour les vidéos v2a et v2b 112
5.10 Résultats sur la prise en compte du critère de difficulté dans la décision de l'agent dans les vidéos v1a et v1b 112
5.11 Résultats sur la prise en compte du critère de proximité dans la décision de l'agent dans les vidéos v2a et v2b 113
5.12 Résultats obtenus sur les questions Chx et Cnn pour les vidéos v2a et v2b 114
5.13 Résultats obtenus sur les questions Q1a, Q1b, Q2a et Q2b pour les vidéos v3a et v3b 116
5.14 Résultats obtenus sur les questions Q3a et Q3b pour les vidéos v3a et v3b 116
5.15 Résultats obtenus sur les questions Q4a et Q4b pour les vidéos v3a et v3b 116
5.16 Résultats sur les questions Q5 et Q6 pour la vidéo v4 117

5.17 Résultats sur la crédibilité des comportements des agents dans les vidéos v3a et v3b 118
5.18 Résultats sur la crédibilité des comportements de l'agent dans la vidéo v4 et la seconde situation 119

Bibliographie

[Abaci et al., 2005] Abaci, T., Cíger, J., and Thalmann, D. (2005). Action semantics in smart objects workshop paper. In *Workshop towards Semantic Virtual Environments*.

[Abbattista et al., 2002] Abbattista, F., Lops, P., Semeraro, G., Andersen, V., and Andersen, H. (2002). *Evaluating virtual agents for e-commerce*. AAMAS'02. Facoltà di Ingegneria.

[Aylett and Cavazza, 2001] Aylett, R. and Cavazza, M. (2001). Intelligent virtual environments : a state-of-the-art report. In *Eurographics 2001, STAR Reports volume*, pages 87 – 109.

[Aylett and Luck, 2000] Aylett, R. and Luck, M. (2000). Applying artificial intelligence to virtual reality : Intelligent virtual environments. *Applied Artificial Intelligence*, 14 :3 – 32.

[Badawi and Donikian, 2004] Badawi, M. and Donikian, S. (2004). Autonomous agents interacting with their virtual environment through synoptic objects. In *Procedings of The 17th Conference on Computer Animation and Social Agents*.

[Barange et al., 2011] Barange, M., De Loor, P., Louis, V., Querrec, R., Soler, J., Trinh, T.-H., Maisel, E., and Chevaillier, P. (2011). Get involved in an interactive virtual tour of brest harbour : Follow the guide and participate. In *Procedings of Intelligent Virtual Agents (IVA '11)*, pages 93 – 99.

[Barbosa et al., 2010] Barbosa, P., Silva, D., Ramalho, G., and Azevedo Tedesco, P. (2010). Simulating the emergence of social relationship networks in groups of believable agents : The X-BARIM model. In *Proceedings of the 20th Brazilian Symposium on Artificial Intelligence, São Bernardo do Campo, (BSAI'10)*, pages 133 – 142.

[Barot et al., 2013] Barot, C., Lourdeaux, D., BURKHARDT, J. M., Amokrane, K., and Lenne, D. (2013). V3S : A Virtual Environment for Risk-Management Training Based on Human-Activity Models. *Presence, teleoperators and virtual environments*, 22(1) :1 – 19.

[Bart et al., 2006] Bart, G., Regis, F., and Sadou, S. (2006). A component-oriented substitution model. In *Proceedings of the 9th International Conference on Software Reuse (ICSR'06)*.

[Bates, 1994] Bates, J. (1994). The role of emotion in believable agents. *Communications of the ACM*, 37(7) :122 – 125.

[Benyahia, 1993] Benyahia, I. (1993). *Automatisation de la circulation des informations en temps réel*. PhD thesis, Université Pierre et Marie Curie, Paris (France).

[Berners-Lee et al., 2001] Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The semantic web. *Scientific American*, 284 :34 – 43.

[Bille et al., 2004] Bille, W., Pellens, B., Kleinermann, F., and De Troyer, O. (2004). Intelligent modelling of virtual worlds using domain ontologies. In *Proceedings of the Workshop of Intelligent Computing (WIC), held in conjunction with the MICAI 2004 conference*, pages 272 – 279.

[Blatier, 1999] Blatier, C. (1999). Attributions causales et comportement délinquant. validation d'un questionnaire traduit en français. *Journal de Thérapie Comportementale et Cognitive*, 9 (3) :89 – 93.

[Bleul and Weise, 2005] Bleul, S. and Weise, T. (2005). An ontology for quality-aware service discovery. In *Proceedings of the First International Workshop on Engineering Service Compositions (WESC'05)*.

[Blum et al., 2004] Blum, A., Jackson, J., Sandholm, T., and Zinkevich, M. (2004). Preference elicitation and query learning. *J. Mach. Learn. Res.*, 5 :649 – 667.

[Blumberg, 1996] Blumberg, B. (1996). *Old Tricks, New Dogs : Ethology and Interactive Creatures*. PhD thesis, University of Cambridge. Type : Thèse de Doctorat.

[Bowman et al., 2003] Bowman, D. A., North, C., Chen, J., Polys, N. F., Pyla, P. S., and Yilmaz, U. (2003). Information-rich virtual environments : theory, tools, and research agenda. In *Proceedings of the ACM symposium on Virtual reality software and technology (VRST'03)*, pages 81 – 90.

[Brogni et al., 2007] Brogni, A., Vinayagamoorthy, V., Steed, A., and Slater, M. (2007). Responses of participants during an immersive virtual environment experience. *IJVR*, 6(2) :1 – 10.

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

[Burkhardt, 2007] Burkhardt, J.-M. (2007). Immersion, représentation et coopération : discussion et perspectives de recherches empiriques pour lérgonomie cognitive de la réalité virtuelle. *Intellectica*, 45(1) :59 – 87.

[Campano, 2013] Campano, S. (2013). *COR-E : un modèle pour la simulation d'agents affectifs fondé sur la théorie COR*. PhD thesis, Université Pierre et Marie Curie (UPMC). Type : Thèse de Doctorat.

[Cassell, 2000] Cassell, j. (2000). More than just another pretty face : Embodied conversational interface agents. *Communications of the ACM*, 43(4) :70 – 78.

[Cavazza, 1998] Cavazza, M. (1998). High-level interpretation in dynamic virtual environments.

[Chang et al., 2005] Chang, P. H.-M., Chien, Y.-h., Kao, E. C.-c., and Soo, V.-w. (2005). A knowledge-based scenario framework to support intelligent planning characters. In *In Procedings of Intelligent Virtual Agents (IVA'05)*, pages 134 – 45. Springer LNCS. Vol.

[Chevaillier, 2006] Chevaillier, P. (2006). *Les systèmes multi-agents pour les environnements virtuels de formation. Habilitation à diriger des recherches*. PhD thesis, Université de Bretagne Occidentale, Brest (France).

[Chevaillier et al., 2009] Chevaillier, P., Querrec, R., and Septseault, C. (2009). Veha : un métamodèle d'environnement virtuel informé et structuré. *RSTI-TSI, Techniques et Sciences Informatiques*, pages 28(6 – 7) : 715 – 740.

[Chevaillier et al., 2011] Chevaillier, P., Trinh, T.-h., Barange, M., Soler, J., De Loor, P., and Querrec, R. (2011). Semantic modelling of virtual environments using mascaret. In *Proceedings of the Fourth Workshop on Software Engineering and Architectures for Realtime Interactive Systems SEARIS, IEEE VR 2011*.

[Christoph, 2005] Christoph, N. (2005). Empirical evaluation methodology for embodied conversational agents. In Ruttkay, Z. and Pelachaud, C., editors, *From Brows to Trust*, volume 7 of *Human-Computer Interaction Series*, pages 67 – 90. Springer Netherlands.

[Chu and Li, 2008] Chu, Y.-L. and Li, T.-Y. (2008). Using pluggable procedures and ontology to realize semantic virtual environments 2.0. pages 1–6.

[De Rosis et al., 2004] De Rosis, F., Pelachaud, C., and Poggi, I. (2004). Transcultural believability in embodied agents : a matter of consistent adaptation. In *Agent Culture : Designing HumanAgent Interaction in a Multicultural World, Laurence Erlbaum Associates, Mahwah*, pages 75 – 106.

[De Sevin et al., 2012] De Sevin, E., Reynaud, Q., and Corruble, V. (2012). Flexmex : Flexible multi-expert meta-architecture for virtual agents. In *Advances in Cognitive Systems (ACS'12)*.

[De Troyer et al., 2003] De Troyer, O., Bille, W., Romero, R., and Stuer, P. (2003). On generating virtual worlds from domain ontologies. In *Proceedings of the 9th International Conference on Multi-Media Modeling*.

[Dobson et al., 2005] Dobson, G., Lock, R., and Sommerville, I. (2005). Qosont : a qos ontology for service-centric system. In *Proceedings of 31st EUROMICRO Conference on Software Engineering and Advanced Applications (CSEAA'05)*.

[Dokonal et al., 2004] Dokonal, W., Martens, B., and Plosch, R. (2004). Creating and using virtual cities. In *Proceedings of eCAADe conference(education in CAAD in Europe)*, pages 580 – 585.

[Donikian, 1997] Donikian, S. (1997). Vuems : A virtual urban environment modeling system. In *Procedings of Conference on Computer Graphics International (CGI'97)*, pages 84 – 92. IEEE Computer Society.

[Doyle, 2002] Doyle, P. (2002). Believability through context using "knowledge in the world" to create intelligent characters. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems : Part 1*, AAMAS '02, pages 342 – 349. ACM Press.

[Edward et al., 2010] Edward, L., Amokrane, K., Lourdeaux, D., and Barthès, J.-P. A. (2010). Knowledge representation - an ontology for managing a virtual environment. In *ICAART*, pages 332 – 335.

[Egges et al., 2003] Egges, A., Kshirsagar, S., and Magnenat-Thalmann, N. (2003). A model for personality and emotion simulation. In Palade, V., Howlett, R. J., and Jain, L., editors, *Knowledge-Based Intelligent Information and Engineering Systems*, volume 2773 of *Lecture Notes in Computer Science*, pages 453 – 461. Springer Berlin Heidelberg.

[Farenc et al., 1999a] Farenc, N., Babski, C., Garat, F., and Thalmann, D. (1999a). Database and virtual human representation on the web. In *Proceedings of Advances in Databases and Multimedia for the New Century - A SwissJapanese Perspective*, pages 61 – 68. World Scientific.

[Farenc et al., 1999b] Farenc, N., Boulic, R., and Thalmann, D. (1999b). An informed environment dedicated to the simulation of virtual humans in urban context. In *Proceedings of EUROGRAPHICS'99*, pages 309 – 318.

[Ferber, 1995] Ferber, J. (1995). *Les systèmes multi-agents : Vers une intelligence collective*. InterEditions.

[Ferber and Muller, 1996] Ferber, J. and Muller, J.-P. (1996). Influences and reaction : a model of situated multiagent systems. In *Proceedings of 2nd International Conference on Multi-Agent Systems (ICMAS'96)*, pages 72 – 79.

[Fishburn, 1970] Fishburn, P. C. (1970). Utility theory for decision making. In *Research Analysis Corporation*.

[Forbus and Wright, 2001] Forbus, K. D. and Wright, W. (2001). Some notes on programming objects in the sims^{TM}. In *Northwestern University*.

[Fox and Long, 2003] Fox, M. and Long, D. (2003). Pddl2.1 : An extension to pddl for expressing temporal planning domains. *Journal of Artificial Intelligence Research*, 20 :61 – 124.

[Gamma et al., 1995] Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1995). *Design Patterns : Elements of Reusable Object-Oriented Software*. Addison-Wesley.

[Ganier et al., 2013] Ganier, F., Hoareau, C., and Devillers, F. (2013). Évaluation des performances et de la charge de travail induits par l'apprentissage de procédures de maintenance en environnement virtuel. *Le Travail Humain*, 76(4) :335 – 363.

[Gosset, 1908] Gosset, W. S. (1908). The probable error of a mean. *Biometrika*, 6(1) :1 – 25.

[Grimaldo et al., 2006a] Grimaldo, F., Barber, F., and Lozano, M. (2006a). An ontology-based approach for ive+va. In *Procedings of IVEVA International Conference*.

[Grimaldo et al., 2006b] Grimaldo, F., Barber, F., Lozano, M., and Orduña, J. (2006b). Semantic virtual environments for interactive planning agents. In *Procedings of International Digital Games Conference (iDiG'06)*.

[Grimaldo et al., 2008] Grimaldo, F., Lozano, M., Barber, F., and Vigueras, G. (2008). Simulating socially intelligent agents in semantic virtual environments. *Knowl. Eng. Rev.*, 23(4) :369 – 388.

[Gruber, 1993] Gruber, T. R. (1993). Toward principles for the design of ontologies used for knowledge sharing. In *Formal ontology in conceptual analysis and knowledge representation*. Kluwer Academic Publishers.

[Guarino and Giaretta, 1995] Guarino, N. and Giaretta, P. (1995). Ontologies and knowledge bases : Towards a terminological clarification. In *Towards very Large Knowledge bases : Knowledge Building and Knowledge sharing*, pages 25 – 32. IOS Press.

[Guessoum, 1996] Guessoum, Z. (1996). *Un environnement opérationnel de conception et de réalisation de systèmes multi-agents*. PhD thesis, Université Pierre et Marie Curie, Paris (France).

[Gutierrez et al., 2005] Gutierrez, M., F., V., and Thalmann, D. (2005). Semantics-based representation of virtual environments. *International Journal of Computer Applications in Technology*, 23(2-3-4) :229 – 238.

[Guye-Vuilléme and Thalmann, 2000] Guye-Vuilléme, A. and Thalmann, D. (2000). Requirements for an architecture for believable social agents. In *Proceedings of the Fourth International Conference on Autonomous Agents*, AGENTS '00, pages 48 – 49. ACM Press.

[Haase et al., 2004] Haase, P., Broekstra, J., Ehrig, M., Menken, M., Mika, P., Olko, M., Plechawski, M., Pyszlak, P., Schnizler, B., Siebes, R., Staab, S., and Tempich, C. (2004). Bibster – a semantics-based bibliographic peer-to-peer system. In *Proceedings of 3rd International Semantic Web Conference (ISWC'04)*.

[Harkouken-Saiah et al., 2011] Harkouken-Saiah, K., Simo-Kanmeugne, P., Sabouret, N., and Beynier, A. (2011). Simulation des déplacements en milieu urbain á l'aide d'un module sémantique. In *La 10ème Rencontre des Jeunes Chercheurs en Intelligence Artificielle*.

[Harmon et al., 1996] Harmon, R., Patterson, W., Ribarsky, W., and Bolter, J. (1996). The virtual annotation system. In *Proceedings of the IEEE : Virtual Reality Annual International Symposium*, pages 239 – 245.

[Hook, 2005] Hook, K. (2005). User-centred design and evaluation of affective interfaces. In Ruttkay, Z. and Pelachaud, C., editors, *From Brows to Trust*, volume 7 of *Human-Computer Interaction Series*, pages 127 – 160. Springer Netherlands.

[Huhns and Singh, 1997] Huhns, M. and Singh, M. (1997). Ontologies for agents. *Internet Computing, IEEE*, 1(6) :81 – 83.

[Ibáñez and Delgado-Mata, 2006] Ibáñez, J. and Delgado-Mata, C. (2006). A basic semantic common level for virtual environments. *International Journal of Virtual Reality (IJVRÓ6)*, (5(3)) :25 – 32.

[Ibáñez and Delgado-Mata, 2011] Ibáñez, J. and Delgado-Mata, C. (2011). Lessons from research on interaction with virtual environments. *Journal of Network and Computer Applications*, 34 :268 – 281.

[Ingrand et al., 1992] Ingrand, F. F., Georgeff, M. P., and Rao, A. S. (1992). An architecture for real-time reasoning and system control. *IEEE expert*, 7(6) :34 – 44.

[Isbister and Doyle, 2002] Isbister, K. and Doyle, P. (2002). Design and evaluation of embodied conversational agents : A proposed taxonomy. In Marriott, A., Pelachaud, C., Rist, T., Ruttkay, Z., and Vilhjalmsson, H., editors, *Proceedings of International Conference on Autonomous Agents and Multi-Agent Systems – Workshop : Embodied Conversatinal Agents*, AAMAS'02.

[Isbister and Höök, 2005] Isbister, K. and Höök, K. (2005). Evaluating affective interfaces : Innovative approaches. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '05, pages 2119 – 2119. ACM Press.

[Jeong et al., 2009] Jeong, B., Cho, H., and Lee, C. (2009). On the functional quality of service (fqos) to discover and compose interoperable web services. In *Expert Syst. Appl.*, volume 36, pages 5411 – 5418.

[Kallmann, 2001] Kallmann, M. (2001). *Object Interaction in Real-Time Virtual Environments*. PhD thesis, Ecole Polytechnique Fédérale de Lausanne. Type : Thèse de Doctorat.

[Kallmann and Thalmann, 1999] Kallmann, M. and Thalmann, D. (1999). Direct 3d interaction with smart objects. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology (VRST '99)*, pages 124–130.

[Kalogerakis et al., 2006] Kalogerakis, E., Christodoulakis, S., and Moumoutzis, N. (2006). Coupling ontologies with graphics content for knowledge driven visualization. In *Proceedings of the IEEE conference on Virtual Reality*, pages 43 – 50. IEEE Computer Society.

[Kalra et al., 1998] Kalra, P., Magnenat-Thalmann, N., Moccozet, L., Sannier, G., Aubel, A., and Thalmann, D. (1998). Real-time animation of realistic virtual humans. *Computer Graphics and Applications, IEEE*, 18(5) :42 – 56.

[Kapahnke et al., 2010] Kapahnke, P., Liedtke, P., Nesbigall, S., Warwas, S., and Klusch, M. (2010). Isreal : An open platform for semantic-based 3d simulations in the 3d internet. In *Procedings of 9th International Semantic Web Conference (ISWC'10)*.

[Kleinermann et al., 2005] Kleinermann, F., De Troyer, O., Mansouri, H., Romero, R., Pellens, B., and Bille, W. (2005). Designing semantic virtual reality applications. In *Proceedings of the 2nd INTUITION International Workshop*, pages 5 – 10.

[Klusch et al., 2005] Klusch, M., Gerber, A., and Schmidt, M. (2005). Semantic web service composition planning with owls-xplan. In Press, A., editor, *Proceedings of the 1st Intl. AAAI Fall Symposium on Agents and the Semantic Web*.

[Laaksolahti et al., 2001] Laaksolahti, J., Persson, P., and Palo, C. (2001). Evaluating believability in an interactive narrative. In *Proceedings of the International Conference on Intelligent Agent Technology*, IAT '01, pages 490 – 497.

[Laird and Duchi, 2000] Laird, J. E. and Duchi, J. C. (2000). Creating human-like synthetic characters with multiple skill levels : A case study using the soar quakebot. In *Proceedings of the AAAI. Fall Symposium : Simulating Human Agents*, AAAI '00, pages 54 – 58.

[Lassila and Swick, 1999] Lassila, O. and Swick, R. R. (1999). Resource description framework : Model and syntax specification. W3C Recommendation, World Wide Web Consortium.

[Lester and Stone, 1997] Lester, J. C. and Stone, B. A. (1997). Increasing believability in animated pedagogical agents. In *Proceedings of the First International Conference on Autonomous Agents*, ICAA '97, pages 16 – 21. ACM Press.

[Li et al., 2005] Li, T.-y., Liao, M.-y., and Tao, P.-c. (2005). Imnet : An experimental testbed for extensible multi-user virtual environment systems. In *ICCSA 2005, LNCS 3480, O. Gervasi et al. (Eds)*, pages 957 – 966.

[Linturi et al., 2000] Linturi, R., Koivunen, M.-R., and Sulkanen, J. (2000). Helsinki arena 2000 - augmenting a real city to a virtual one. In Ishida, T. and Isbister, K., editors, *Digital Cities*, volume 1765 of *Lecture Notes in Computer Science*, pages 83 – 96. Springer Berlin Heidelberg.

[Loughlin and Hughes, 1994] Loughlin, M. and Hughes, J. (1994). Anannotation system for 3d fluid flow visualisation. In *Proceedings of the IEEE conference on visualisation*, pages 273 – 279.

[Loyall, 1997] Loyall, B. A. (1997). *Believable Agents : Building Interactive Personalities*. PhD thesis, Carnegie Mellon University. Type : Thèse de Doctorat.

[Maass and Dollner, 2006] Maass, S. and Dollner, J. (2006). Dynamic annotation of interactive environments using object-integrated billboards. In *14th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG'06)*, pages 327 – 334.

[Mac Namee, 2004] Mac Namee, B. (2004). *Proactive Persistent Agents : Using Situational Intelligence to Create Support Characters in Character-Centric Computer Games.* PhD thesis, University of Dublin. Type : Thèse de Doctorat.

[Magnin, 1996] Magnin, L. (1996). *Modélisation et simulation de l'environnement dans les systèmes multi-agents.* PhD thesis, Université Pierre et Marie Curie, Paris (France).

[Marion et al., 2007] Marion, N., Septseault, C., Boudinot, A., and Querrec, R. (2007). Gaspar : Aviation management on an aircraft carrier using virtual reality. In *Procedings of The International Conference on Cyberworlds (CW '07)*, pages 15 – 22.

[Martin et al., 2004] Martin, D., Burstein, M., Hobbs, J., Lassila, O., McDermott, D., McIlraith, S., Narayanan, S., Paolucci, M., Parsia, B., Payne, T., Sirin, E., Srinivasan, N., and Sycara, K. (2004). Owl-s : Semantic markup for web services. In *W3C*, http ://www.w3.org/Submission/OWL-S/.

[Metoyer and Hodgins, 2000] Metoyer, R. and Hodgins, J. K. (2000). Animating athletic motion planning by example. In *In Proceedings of Graphics Interface*, pages 61 – 68.

[Muller et al., 2004] Muller, H., Kenny, E., and Sternberg, P. (2004). Textpresso : An ontology-based information retrieval and extraction system for biological literature. In *PLoS Biol 2.*

[Neal Reilly, 1996] Neal Reilly, S. (1996). *Believable Social and Emotional Agents.* PhD thesis, Carnegie Mellon University. Type : Thèse de Doctorat.

[Ortony, 2003] Ortony, A. (2003). On making believable emotional agents believable. *Emotions in Humans and Artifacts*, pages 189 – 212.

[Otto, 2005] Otto, K. (2005). The semantics of multi-user virtual environments. In *Procedings of Workshop Towards Semantic Virtual Environments*, pages 47 – 56.

[Pelachaud and Bilvi, 2003] Pelachaud, C. and Bilvi, M. (2003). Computational model of believable conversational agents. In Huget, M.-P., editor, *Communication in Multiagent Systems*, volume 2650 of *Lecture Notes in Computer Science*, pages 300 – 317. Springer Berlin Heidelberg.

[Pellens et al., 2005] Pellens, B., De Troyer, O., Bille, W., and Kleinermann, F. (2005). Conceptual modeling of object behavior in a virtual environment. Springer-Verlag.

[Perlin and Goldberg, 1996] Perlin, K. and Goldberg, A. (1996). Improv : A system for scripting interactive actors in virtual worlds. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, pages 205 – 216.

[Pirard, 2010] Pirard, F. (2010). Pour une évaluation participative de la qualité effective de l'accueil. pages 37 – 68.

[Plexousakis and Kritikos, 2007] Plexousakis, K. and Kritikos, D. (2007). Extending owl for qos-based web service description and discovery. In *Proceedings of the Workshop on Service Matchmaking and Resource Retrieval in the Semantic Web (SMR2'07).*

[Poulet, 2013] Poulet, C. (2013). *Coordination dans les systèmes multi-agents :Le problème de la patrouille en système ouvert.* PhD thesis, Université Pierre et Marie Curie (UPMC). Type : Thèse de Doctorat.

[Querrec, 2002] Querrec, R. (2002). *Les Systèmes Multi-Agents pour les Environnements Virtuels de Formation : Application à la sécurité civile.* PhD thesis, Université de Bretagne Occidentale, Brest (France).

[Querrec, 2010] Querrec, R. (2010). *Apprentissage de procédures en environnements virtuels. Habilitation à diriger des recherches.* PhD thesis, Université de Bretagne Occidentale, Brest (France).

[Reynaud, 2014] Reynaud, Q. (2014). *Architecture cognitive d'agents situés.* PhD thesis, Université Pierre et Marie Curie (UPMC). Type : Thèse de Doctorat.

[Reynaud et al., 2012] Reynaud, Q., De Sevin, E., Donnart, J.-Y., and Corruble, V. (2012). A cognitive module in a decision-making architecture for agents in urban simulations. In *AAMAS 2012 Workshop on Cognitive Agents in Virtual Environments (CAVE)*, number 1, pages 1 – 15.

[Reynaud et al., 2014] Reynaud, Q., Donnart, J.-Y., and Corruble, V. (2014). Evaluating the efficiency and belivability of virtual agents with anticipatory abilities. In *AAMAS 2014 An Autonomous Agents and Multi-Agents Systems, Procedings)*, number 1, pages 1533 – 1534.

[Ruttkay et al., 2002] Ruttkay, Z., Dormann, C., and Noot, H. (2002). Evaluating ecas - what and how ? In Marriott, A., Pelachaud, C., Rist, T., Ruttkay, Z., and VilhjÃ¡lmsson, H., editors, *Procedings of the International Conference on Autonomous Agents and Multi-Agent Systems – Workshop : Embodied Conversatinal Agents*, AAMAS'02.

[Sai et al., 2001] Sai, Y., Yao, Y. Y., and Zhong, N. (2001). Data analysis and mining in ordered information tables. In *Proceedings of The IEEE Int'l Conf. on Data Mining*, pages 497 – 504.

[Salembier and Sikora, 2002] Salembier, P. and Sikora, T. (2002). *Introduction to MPEG-7 : Multimedia Content Description Interface*. John Wiley & Sons, Inc., New York, NY, USA.

[Sengers, 1999] Sengers, P. (1999). Designing comprehensible agents. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, IJCAI '99, pages 1227 – 1232. Morgan Kaufmann Publishers Inc.

[Septseault, 2007] Septseault, C. (2007). *Représentation d'environnements virtuels informés et de leur dynamique par un personnage autonome en vue d'une crédibilité comportementale*. PhD thesis, Université de Bretagne Occidentale, Brest (France).

[Simo Kanmeugne, 2014] Simo Kanmeugne, P. (2014). *Simulation crédible des déplacements de piétons en temps réel : modèle microscopique à influence macroscopique*. PhD thesis, Université Pierre et Marie Curie (UPMC). Type : Thèse de Doctorat.

[Simo Kanmeugne and Beynier, 2014] Simo Kanmeugne, P. and Beynier, A. (2014). Simulating Autonomous Pedestrians Navigation : A Generic Multi-Agent Model to Couple Individual and Collective Dynamics.

[Soto and Allongue, 1997] Soto, M. and Allongue, S. (1997). A semantic approach of virtual worlds interoperability. In *WETICE*, pages 173 – 178.

[Soto and Allongue, 1998] Soto, M. and Allongue, S. (1998). Interoperable networked virtual environments. In *ICMCS*, pages 222 – 225.

[Soto and Allongue, 2002] Soto, M. and Allongue, S. (2002). Modeling methods for reusable and interoperable virtual entities in multimedia virtual worlds. *Multimedia Tools and Applications*, 16(1-2) :161 – 177.

[Stanney, 2002] Stanney, K. M. (2002). *Handbook of virtual environments : design, implementation, and applications*. Lawrence Erlbaum Associates.

[Steels, 1994] Steels, L. (1994). The artificial life roots of artificial intelligence. *Artificial Life Journal*, 1(1) :75 – 110.

[Tansu et al., 2007] Tansu, A., C., B., and Evangelos, K. (2007). Towards 3d internet : Why, what, and how ? In *Procedings of IEEE Intl. Conference on Cyberworlds (CW'07)*, pages 95 – 99.

[Thomas and Johnston, 1981] Thomas, F. and Johnston, O. (1981). *The Illusion of Life : Disney Animation*. Abbeville Press, Disney Editions. New York.

[Tondello and Siqueira, 2008] Tondello, F. G. and Siqueira, F. (2008). The qos-mo ontology for semantic qos modeling. In *Proceedings of the ACM Symposium on Applied Computing (SAC'08)*.

[Tsarkov and Horrocks, 2006] Tsarkov, D. and Horrocks, I. (2006). Fact++ description logic reasoner : System description. In Furbach, U. and Shankar, N., editors, *Automated Reasoning*, volume 4130 of *Lecture Notes in Computer Science*, pages 292 – 297. Springer Berlin Heidelberg.

[Tutenel et al., 2008] Tutenel, T., Bidarra, R., Smelik, R. M., and De Kraker, K. J. (2008). The role of semantics in games and simulations. *Comput. Entertain*, 6 :1 – 35.

[Tutenel et al., 2009] Tutenel, T., Smelik, R. M., Bidarra, R., and De Kraker, K. J. (2009). Using semantics to improve the design of game worlds. In *Procedings of the 15th Artificial Intelligence and Interactive Digital Entertainment Conference (AIIDE'09)*, pages 100 – 105.

[Van Oijen and Dignum, 2011] Van Oijen, J. and Dignum, F. (2011). Scalable perception for bdi-agents embodied in virtual environments. In *International Conferences on Web Intelligence and Intelligent Agent Technology*, volume 2, pages 46 – 53. IEEE Computer Society.

[Vanacken et al., 2008] Vanacken, L., De Boeck, J., Raymaekers, C., and Karin, C. (2008). Using relations between concepts during interaction modelling for virtual environments. In *Software Engineering and architectures for Realtime Interactive Systems*.

[Vanacken et al., 2007] Vanacken, L., Raymaekers, C., and Coninx, K. (2007). Introducing semantic information during conceptual modelling of interaction for virtual environments. In *Proceedings of the 2007 Workshop on Multimodal Interfaces in Semantic Interaction*, WMISI '07, pages 17–24.

[Vu, 2003] Vu, L. H. e. a. (2003). A search engine for qos-enabled discovery of semantic web services. In *Proceedings of International Journal of Business Process Integration and Management (IJBPIM'03)*.

[Yessad and Boufaida, 2011] Yessad, L. and Boufaida, Z. (2011). A qos ontology-based component selection. In *Proceedings of the International Journal on Soft Computing (IJSC'11)*.

[Yong et al., 2006] Yong, L., Congfu, X., Zhigeng, P., and Yunhe, P. (2006). Semantic modeling for ancient architecture of digital heritage. *Computers & Graphics*, 30(5) :800 – 814.

[Zhang et al., 2012] Zhang, Q., Wang, X., Xie, K., and Yang, F. (2012). Research on component quality model. In *Proceedings of the International Conference on Computing, Measurement, Control and Sensor Network (CMCSN'12)*.

[Zhou et al., 2004] Zhou, C., Chia, L. T., and Lee, B. S. (2004). Daml-qos ontology for web services. In *Proceeding of the International Conference on Web Services (ICWS'04)*.

[Zimeo and Giallonardo, 2007] Zimeo, E. and Giallonardo, E. (2007). More semantics in qos matching. In *Proceedings of the IEEE International Conference on Service-Oriented Computing and Applications (SOCA'07)*.