Serious Games et SMA - Application à un supermarché virtuel

Philippe Mathieu, David Panzoli, Sébastien Picault

▶ **To cite this version:**

Philippe Mathieu, David Panzoli, Sébastien Picault. Serious Games et SMA - Application à un supermarché virtuel. 19e Journées francophones sur les systèmes multi-agents(JFSMA'2011), Oct 2011, Valenciennes, France. pp.181-190. hal-00826434

HAL Id: hal-00826434
https://hal.science/hal-00826434

Submitted on 29 Oct 2013

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

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

Serious Games et SMA - Application à un supermarché virtuel

P. Mathieu philippe.mathieu@lifl.fr
D. Panzoli david.panzoli@lifl.fr
S. Picault sebastien.picault@lifl.fr

Laboratoire d'Informatique Fondamentale de Lille, UMR CNRS 8021
Université Lille 1, France

Résumé

La pertinence des systèmes multi-agents (SMA) a été démontrée à de nombreuses reprises dans la conception de simulations informatiques ou de jeux vidéo où un certain nombre d'entités autonomes évoluent dans un environnement complexe et dynamique. Les Serious Games *(SG) représentent une discipline nouvelle, à la frontière de la simulation et du jeu. Nous pensons qu'une catégorie de SG, ayant pour vocation l'immersion de l'apprenant dans un environnement 3d, représente un banc de test particulièrement intéressant pour les SMA car ils introduisent des problématiques nouvelles et stimulantes pour la communauté. Dans cet article, nous explorons les défis lancés à l'approche SMA par ces SG immersifs. Particulièrement, nous démontrons que l'approche multi-agents orientée interactions* IODA*, dont l'aptitude à faciliter la conception de simulations a déjà été établie, permet de répondre efficacement à ces nouvelles problématiques. Nous illustrons notre argumentaire en nous basant sur un projet de SG développé dans notre équipe.*

Mots-clés : Serious Game*, Système Multi-Agents, Adaptativité, Interactions*

Abstract

In this paper, we claim that immersive 3d Serious Games (SG) represent an interesting testbed for multi-agent systems to prove their ability to model compelling and user-enticing simulations. We explore the different challenges arising from the use of an interaction-oriented methodology such as IODA *for modelling a realistic population of autonomous virtual characters able to behave and interact with the learner in an adaptive fashion. Our argumentation is illustrated with many examples from our own experience in the development of a SG application.*

Keywords: *Serious Game, Multi-Agent System, Adaptivity, Interactions*

1 Introduction

Les simulations multi-agents (MA) s'appliquent avec succès à de nombreux domaines, mais surtout sont capables de répondre à des objectifs variés : la compréhension des mécanismes sous-jacents à des phénomènes naturels (de la biologie [6], des écosystèmes [4]) ou humains (l'animation de mondes artificiels [3], la simulation de comportements sociaux [7] ou la finance [1]) ; la prédiction de comportements (gestion de catastrophes [16], simulateurs de conduite Scanner [13] et ArchiSim[8]) ; ou encore, à vocation ludique ou artistique (animation, jeux vidéos). [1]

Issus du *Kriegspiel*, les *Serious Games* (SG) ont connu un essor très net ces dernières années, et s'inscrivent à la confluence de plusieurs objectifs : ludique bien sûr, mais aussi éducatif ou informatif, en assurant en général une forme d'entraînement dans un contexte réaliste.

Or, bien que ces fonctions recoupent assez largement les emplois de la simulation MA, cette dernière reste peu ou pas employée pour la réalisation de SG. Notre propos ici est d'abord de montrer en quoi un certain type de SG peut constituer un banc d'essai privilégié pour la mise à l'épreuve des techniques de simulation. Nous souhaitons également montrer des pistes concrètes pour s'attaquer à leur réalisation, notamment à travers une approche orientée interactions qui est particulièrement adaptée. Nous illustrerons notre propos à travers des exemples tirés d'un SG réalisé dans l'équipe SMAC, le projet FORMAT-STORE.

2 État de l'art

Un *serious game* (SG) est un jeu destiné à la formation d'un public d'apprenants (également

1. http ://www.massivesoftware.com/

considérés comme joueurs [2]) à un métier ou leur sensibilisation à un thème particulier (patrimoine culturel, écologie, sécurité, etc). Le SG a pour ambition d'incorporer dans une simulation certains aspects propres au jeu, dans le but de favoriser l'engagement de l'apprenant, et ainsi renforcer l'impact du SG. Nous nous intéressons à la catégorie des SG immersifs, dans lequel l'immersion du joueur est considérée comme la condition primordiale à cet engagement. Comme illustré en gras dans la figure 1, l'immersion est assurée d'une part par le réalisme visuel du jeu et d'autre part par la scénarisation des interactions entre le joueur et les personnages non-joueurs (PNJ).

Dans certains SG immersifs [2, 15], l'accent est placé sur le réalisme visuel. L'apprenant se déplace librement dans un environnement 3d peuplé de personnages virtuels autonomes se comportant de manière réaliste et cohérente. Du fait de la difficulté de scénariser les interactions dans ce contexte de liberté, les PNJ se contentent d'offrir des informations factuelles, comme des indications ou des objectifs de quête.

D'autres SG font primer la scénarisation sur le réalisme visuel. La liberté de l'apprenant et la complexité du comportement des PNJ sont ainsi réduites au profit de dialogues scriptés dans un environnement contrôlé. Le projet *Banque Cantonale Vaudoise*, développé par Daesign [3] pour la banque éponyme, place l'apprenant dans une situation de dialogue interactif pour le former au placement de produits financiers pour des clients revêtant divers profils. *The Sales Game* de PIXELearning [4] offre une scénarisation plus riche où l'apprenant apprend à construire son réseau professionnel ou enrichir sa base de clients. *Knowledge Drive*, développé par Caspian Learning [5] pour Volvo Car UK, entraîne le vendeur à construire un argumentaire de vente à partir d'une discussion avec un client. Bien que l'apprenant soit laissé libre de se déplacer dans une restitution d'un *showroom* accompagné du client, le comportement de ce dernier est extrêmement simpliste.

Idéalement, intégrer une simulation multi-agents des PNJ devrait permettre d'éviter de transiger entre réalisme et scénarisation. Dans la pratique cependant, c'est peu souvent le cas car un SG est développé par un studio de jeux vidéo pour des experts qui fournissent le contenu pédagogique, et aucun de ces deux acteurs ne possède les compétences pour appréhender l'aspect multi-agents, voire l'aspect agent. Réciproquement, nous allons voir que les problématiques de jeu et de pédagogie induisent des contraintes importantes sur la façon de concevoir et de réaliser la partie simulation et contribuent à faire des SG un banc d'essai novateur pour la mise à l'épreuve des méthodologies et des techniques de simulation MA.

3 Spécificités des SG du point de vue SMA

Fondamentalement, la conception d'une simulation MA nécessite la définition d'un modèle décrivant les connaissances et les hypothèses devant être incorporées dans la simulation, puis la spécification des comportements locaux et des propriétés des agents de la simulation permettant de les représenter efficacement. Pour rendre possible l'utilisation de la simulation MA dans un contexte de SG, un certain nombre de contraintes liées aux aspects jeu et pédagogie doivent être prises en compte. La figure 1 recense l'ensemble de ces problématiques, en plaçant l'accent sur leur interdépendance.

Un SG est avant tout un jeu, ce qui se traduit par la nécessité d'intégrer l'apprenant en tant qu'acteur, ainsi que la volonté de lui proposer un environnement immersif. Dans la conception de la simulation MA, ces deux contraintes s'expriment à travers trois problématiques :

1. La simulation doit être participative en prenant en compte l'élément humain dans la gestion des agents. L'humain interagit avec les agents, au même titre qu'ils interagissent entre eux, mais pas nécessairement selon les mêmes modalités et la même temporalité. En outre, ses interventions sont imprévisibles. Réciproquement, il est souvent nécessaire de pouvoir lancer le jeu sans joueur humain (pour éditer des documents à usage pédagogique ou vérifier la validité du comportement des PNJ) ou au contraire avec plusieurs humains (compétition ou travail collaboratif). Cela ne doit pas nécessiter de changement majeur et le joueur humain doit donc être appréhendé comme interchangeable avec un agent autonome.
2. Le comportement des agents doit être réaliste et cohérent pour favoriser l'immersion

2. Nous employons les termes "joueur" et "apprenant" comme synonymes dans le reste de cet article.
3. http ://www.daesign.com
4. http ://www.pixelearning.com/services-the_sales_game.htm
5. http ://www.caspianlearning.co.uk/

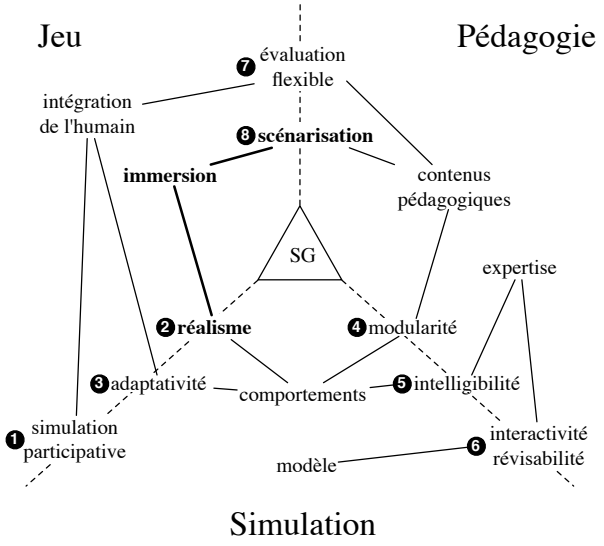

FIGURE 1 – Interdépendance des problèmes soulevés par un *serious game* immersif. Les trois composantes d'un tel SG (pédagogie, jeu, réalisme) prises séparément engendrent leurs propres problématiques, mais la résolution de ces dernières est considérablement compliquée par le nécessaire équilibre qui doit être établi entre ces trois objectifs. Ainsi par exemple, l'immersion du joueur passe par un soucis de réalisme, qui fait appel à une modélisation appropriée du comportement des PNJ (problème de simulation), mais aussi par la mise en scène des contenus pédagogiques à travers des agents, des comportements et un environnement spécifiques.

du joueur dans le jeu, c'est à dire le degré avec lequel le joueur adhère à la simulation.

3. Le comportement des agents doit être adaptatif car l'environnement est extrêmement dynamique, du fait de leurs actions respectives mais également de la présence du joueur et de son imprévisibilité.

Les aspects pédagogiques sont caractérisés par deux enjeux. Les experts doivent pouvoir exprimer des recommandations quant aux caractéristiques des PNJ (profils) ou leur comportement (plans, capacités d'interactions). Parallèlement, ils définissent des objets pédagogiques (des informations, des dialogues) qui doivent pouvoir être intégrés dans ces comportements. Ces enjeux se traduisent par les problématiques suivantes :

4. L'architecture comportementale des agents doit être modulaire pour regrouper et arbitrer au sein d'un système unique des capacités de complexités différentes : se déplacer, interagir, raisonner, communiquer, etc.
5. Les comportements doivent être suffisamment intelligibles pour que l'expert puisse s'assurer de la bonne intégration de ses recommandations, voire y participer.
6. L'évaluation de la simulation ne pouvant se faire que de manière expérimentale (par l'observation macroscopique de la simulation) et incrémentale (par modifications successives), il est nécessaire que le modèle de la simulation soit aisément révisable, idéalement par le biais d'un processus de conception interactif.

Enfin, deux dernières problématiques sont issues de la combinaison des aspects liés à la pédagogie et au jeu.

7. Dans un SG, l'apprenant est évalué durant la session de jeu, quelle que soit la manière dont le résultat lui est communiqué (par un score en cours de session, par un bilan en fin de session ou de manière hybride). Dans le but d'accommoder la grande diversité des situations possibles, cette évaluation doit donc revêtir un caractère flexible en étant proposée à des temps différents, et sous des formes différentes.
8. Dans une volonté d'immersion de l'apprenant, la présentation des contenus pédagogiques doit être scénarisée. Bien que les approches centrées agents facilitent leur contextualisation, en permettant aux agents d'incarner des situations pédagogiques, leur scénarisation (au sens strict, par utilisation d'un script) est rendue délicate par l'autonomie des agents dans un SMA.

Individuellement, la prise en compte de chaque problématique introduite par le SG ne présente pas d'écueil particulier, comme le résume le tableau 1. Mais c'est bien la conjonction dans une même application de toutes ces problématiques qui rend l'exercice difficile. En ce sens, le SG représente un banc d'essai stimulant pour la mise à l'épreuve des techniques et des outils de la communauté simulation MA. Pour notre part, nous pensons qu'une méthodologie orientée interactions telle que IODA, dont nous rappelons les principes dans la section suivante, offre une bonne réponse aux défis issus du SG : nous allons l'illustrer en expliquant comment un SG concret (le projet FORMAT-STORE) est réalisé au moyen de IODA (cf. section 5).

4 IODA : Une méthodologie de conception orientée interactions

IODA est née en 2001 [14] de la volonté de proposer une méthodologie multi-agents à la fois

#	Problématique *serious game*	Illustration SMA (méthodes et exemples)
1	Simulation participative	Immersion d'un humain [8] ou situations à problèmes présentées à un groupe [16]
2	Réalisme	Simulations explicatives [1, 4, 6] Simulations immersives [8] Comportements diversifiés [13]
3	Adaptativité	Environnements dynamiques [5, 16]
4	Comportements modulaires	Bibliothèque d'interactions indépendante des agents [12]
5	Comportements intelligibles	Règles (interactions [12] ou normes [13])
6	Modèle révisable	Homogénéité des entités/comportements [11] Conception incrémentale [7]
7	Évaluation de l'apprenant	Observations des réactions de participants humain [8, 16]
8	Scénarisation	Mise en scène de problèmes [16]

TABLE 1 – Tableau résumant les problématiques introduites par un SG immersif, et quelques exemples de techniques ou d'applications SMA qui répondent à quelques-unes de ces problématiques isolément.

simple à utiliser pour le concepteur et puissante en terme de complexité des simulations réalisables.

IODA [10, 12] est une méthodologie de conception orientée interactions (les méthodologies de type IOP [18] se concentrent prioritairement sur les interactions entre les agents, à l'inverse des méthodologies type AOP [17] qui s'intéressent au comportement intrinsèque de l'agent) dont l'originalité est d'accorder une tangibilité logicielle aux interactions et de proposer une conception unifiée de la simulation MA. IODA est fondée sur un argument provenant d'observations expérimentales. Le processus de conception d'une simulation requiert la formulation d'hypothèses, indépendamment du phénomène simulé. Dans un contexte multi-agents, lorsque le phénomène considéré est le résultat évident des interactions d'une multitude d'agents, la description de ces dernières constitue l'unique hypothèse objective pouvant être formulée. L'ensemble des processus internes mis en œuvre par les agents ne peut qu'être deviné. Dans le but d'éviter l'introduction d'un biais trop tôt dans le processus de conception, une approche prudente consiste à établir les bases du modèle sur des caractéristiques observables, idéalement formulées par les experts du domaine, puis à enrichir le modèle avec la préoccupation constante de retarder le plus possible l'introduction de nouvelles hypothèses. IODA suit ce principe en proposant un certain nombre de recommandations.

4.1 Tout est agent

Le point de départ dans la conception d'une simulation MA consiste à identifier les agents participant à la simulation. Un agent dans un SMA est caractérisé par un degré minimal d'autonomie, se traduisant par la capacité à déclencher une action ou une interaction de manière autonome. Généralement, les personnages "vivants" dans une simulation sont considérés comme des agents alors que les objets "inanimés" appartenant à l'environnement (arbres, meubles, etc.) ne le sont pas.

La première recommendation de IODA consiste à considérer toute entité participant à la simulation comme un agent [11]. La première étape de la conception consiste donc à lister les familles d'agents, de manière à ce que chaque agent appartienne strictement à une famille, et que tous les agents soient représentés.

4.2 L'interaction réifiée

De la même manière que chaque entité dans la simulation est représentée par un agent, chaque comportement peut être décrit dans IODA par une interaction.

Contrairement à d'autres approches SMA, où l'interaction est exprimée implicitement dans le comportement des agents, chaque interaction dans IODA est une composante logicielle tangible qui occupe une place centrale dans la conception. Une interaction est une règle impliquant deux agents : l'agent source exécute l'interaction alors que l'agent cible la subit. L'interaction est composée de deux parties : précondition et action, chacune reposant sur des primitives de perception et d'action et étant donc indépendante de l'implémentation concrète des agents (on parle de réification des interactions).

En conséquence, IODA exhibe deux caractéristiques uniques. D'abord, les interactions peuvent être représentées indépendamment des agents, sous forme de librairies d'interactions par exemple. Les interactions sont réutilisables d'une famille d'agents à une autre, et d'une simulation à une autre. Elles sont allouées aux agents de manière transparente et intuitive (cf. la section 4.3 consacrée à la matrice des interactions). L'autre avantage de la réification des

interactions consiste en l'utilisation d'un processus générique et d'une boucle unique pour le traitement des agents, et ce indépendamment de leur nature (cf. la section 4.4 consacrée à la sélection de l'action).

4.3 Matrice des interactions

Une fois les familles d'agents identifiées, l'étape suivante dans la méthodologie IODA consiste en la description de leurs interactions. Ceci implique également leur allocation dans une matrice des interactions.

Le tableau 2 présente un exemple de matrice des interactions. Les familles d'agents sont listées en abscisse et en ordonnée. Les interactions sont ensuite affectées de manière à identifier visuellement les familles d'agents source et cible. Une interaction se lit `nomInteraction(dist, prt)`. Lorsque l'agent cible de l'interaction est l'agent source lui-même, l'interaction est appelée dégénérée et se lit `nomInteraction(prt)`. Une interaction possède les caractéristiques suivantes :

- La distance `dist` définit la distance minimale entre l'agent source et l'agent cible pour que l'interaction puisse être considérée. Les agents étant situés dans l'environnement, la distance est donnée par une métrique préalablement choisie.
- La priorité `prt` est utilisée pour trier les différentes interactions réalisables, lorsque, pour un agent donné, il en existe plusieurs en compétition à un instant donné.

IODA permet également, à travers une seconde matrice appelée la matrice de mise à jour, la prise en compte des changements d'états qui ne relèvent pas d'une interaction particulière (vieillissement d'un agent, usure, etc).

La matrice des interactions présente l'avantage d'être lue très naturellement par un non-informaticien. Malgré son apparente simplicité, elle décrit le comportement de chaque agent de manière exhaustive. En outre, exploitée par le mécanisme de sélection de l'action présenté dans la section suivante, elle est fonctionnellement équivalente à tout type de représentation du comportement.

4.4 Sélection de l'action

L'expression de la matrice des interactions en un comportement pour chaque agent est opérée par un mécanisme de sélection de l'action.

Traditionnellement, la sélection de l'action consiste en une boucle interne perception→décision→action, opérée par l'agent lui-même selon le principe de l'autonomie comportementale. Évaluer la perception d'un agent consiste à recenser les agents dans son voisinage local. Les agents étant situés dans un environnement, le voisinage est donné par une métrique prédéfinie dans l'environnement. Des métriques possibles peuvent être la distance dans un espace euclidien ou les accointances dans un espace social. Le processus de décision permet, à partir de la perception, de décider d'une action appropriée. Celui-ci est plus ou moins complexe selon la nature de l'agent. Un agent réactif prend une décision relativement triviale alors qu'un agent cognitif manipule des buts internes et planifie de manière proactive des plans menant à ces buts. L'action sélectionnée est ensuite exécutée.

Dans le cas de IODA, l'approche orientée interactions simplifie grandement ce processus. À chaque pas de temps, ce dernier étant discrétisé dans IODA, l'agent considère l'ensemble des interactions possibles à partir de la matrice des interactions et des agents dans son voisinage. Chaque interaction est ensuite qualifiée de réalisable si la distance entre lui-même et l'agent cible est inférieure à la garde de distance de l'interaction et si les préconditions sont vérifiées. S'il existe plusieurs interactions réalisables à l'issue de ce processus, l'agent exécute celle de priorité la plus haute. De fait, les agents dans la méthodologie IODA sont tous considérés comme réactifs.

En optant pour une méthodologie orientée interactions et en se libérant de l'hétérogénéité des processus de sélection de l'action des agents, IODA propose un mécanisme de gestion du comportement extrêmement simple. Toutefois, celui-ci permet d'obtenir des comportements extrêmement complexes, comme illustré dans la section 5.3

5 Mise en œuvre d'un SG à travers l'approche orientée interactions

Dans cette section, nous examinons l'application de la méthodologie IODA à la réalisation d'un projet de SG, mené en collaboration avec deux partenaires : Idées-3Com, un studio de développement de jeux vidéo, et l'école de commerce ENACO. Nous détaillons de quelle ma-

nière IODA répond aux problématiques définies dans la section 2 et particulièrement comment sont simulés les comportements réalistes et cohérents d'agents humanoïdes.

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

FORMAT-STORE a pour ambition de proposer une mise en contexte des situations-problèmes, permettant à l'apprenant d'appliquer dans un magasin virtuel les connaissances acquises de manière traditionnelle sur la plate-forme, et d'expérimenter diverses façons de traiter un cas. En outre, il permet en parallèle l'apprentissage de compétences liées à la gestion du temps ou la priorisation des tâches. L'utilisation d'un SMA est motivée par la volonté de proposer à l'apprenant une expérience immersive dans laquelle des clients virtuels aux profils différents simulent l'activité et les problèmes de clients réels. Par le biais de son avatar, l'apprenant assume le rôle d'un vendeur dans le magasin (voir figure 2).

FIGURE 2 – Dans FORMAT-STORE, l'apprenant évolue dans un magasin à travers un avatar personnalisable qu'il contrôle avec les flèches de son clavier. Par le biais de son avatar, il peut aussi interagir avec les clients et les articles en utilisant la souris.

Dès son entrée dans le magasin, chaque client se voit attribuer une liste d'articles à acheter. De manière autonome, il se déplace dans le magasin à la recherche de ces articles (cf. section 5.3), qu'il place ensuite dans son panier. Une ou plusieurs situations-problèmes peuvent également être affectées à un client (cf. section 5.2), ce qui se traduit par la nécessité de trouver le vendeur puis d'engager un dialogue. En situation conversationnelle, comme l'illustre la figure 3, l'apprenant doit sélectionner les réponses correctes aux questions des clients virtuels. À la fin du dialogue, et indépendamment des réponses du joueur, le client retourne à son activité. D'autres situations-problèmes peuvent ainsi être cumulées durant l'activité du client, ayant pour cause des perturbations introduites volontairement (une allée peut être encombrée) ou non (un article peut se trouver en rupture, leur quantité étant limitée).

FIGURE 3 – Les dialogues des situations-problèmes sont mis en scène lors d'interactions conversationnelles, par le biais d'une interface spécifique.

5.1 Intégration de l'expert dans la conception

Dans la section 4, nous avons relevé l'importance d'intégrer les experts dans le processus de conception de la simulation. La méthodologie IODA, à travers différents aspects, répond parfaitement à cette problématique.

Comme préconisé dans la section 4.1, toutes les entités participant à la simulation sont représentées par une famille d'agents. On retrouve donc le `Vendeur`, les `Clients`, mais aussi les `Articles`, les `Panneaux` qui servent à l'orientation des clients et la `Porte` par laquelle les agents entrent et sortent (voir tableau 2 pour la liste complète des agents). Pour l'expert, participer à la première étape de modélisation consiste donc simplement à recenser

toutes les entités jouant un rôle dans la simulation, que ce dernier soit actif ou passif.

De la même manière, la formulation du comportement des agents sous forme d'interactions est très naturelle pour l'expert, et son placement dans la matrice (cf. section 4.3) très intuitif. Prenons l'exemple d'une requête de l'expert concernant l'orientation des clients : « Les clients localisent la position des articles qui les intéressent en lisant les panneaux d'informations situés dans le magasin, et se rendent ensuite vers le plus proche ». La traduction de cette requête en interaction est quasi triviale : Une première interaction `Informer` est positionnée dans la matrice à l'intersection du `Panneau` (source) et du `Client` (cible). Une deuxième interaction dégénérée `Aller vers` est placée dans la cellule adéquate de la ligne du `Client`. Pour chaque interaction, les conditions d'exécution sont uniquement déterminées par les préconditions et la distance. Par exemple, un `Client` sera informé par tout panneau à une distance inférieure ou égale à 10, à la condition que l'article désigné par le panneau soit sur sa liste de courses. Il sera capable d'`Aller vers` tout article à la condition qu'il en connaisse la position.

L'intégralité du comportement de chaque agent est ainsi représenté de manière extrêmement *intelligible (5)*. Pour chaque agent, sa ligne dans la matrice regroupe les interactions qu'il peut exécuter (respectivement sa colonne celles qu'il peut subir). Par exemple, un `Client` peut (par ordre de priorité décroissante) `Repérer` un vendeur, `Sortir` du magasin, `Sortir` de la file d'attente, `Payer` à la caisse, `Se placer` et `Entrer` dans la file d'attente, `Prendre` un article, `Aller vers` une position donnée dans le magasin ou `Errer`. Le `Client` est informé par un `Panneau`, un `Article`, la `Caisse` ou la `Porte` par la même interaction, compte tenu du caractère polymorphe des interactions, lié à leur réification.

La matrice des interactions constitue l'unique modèle des comportements de la simulation, et il est très simplement *révisable (6)* grâce à deux modules additionnels de IODA : JEDI et JEDI-Builder. JEDI est une API Java qui assure une implémentation rigoureuse d'un modèle conceptuel IODA. JEDI-Builder est un applet Java qui assure deux rôles : fournir une interface intuitive permettant la création ou la modification d'un modèle IODA directement à la souris, et traduire rapidement et automatiquement un modèle IODA en application JEDI. Grâce à ces outils, les paramètres de chaque interaction, à savoir les familles d'agents source et cible, les préconditions, la distance et la priorité, peuvent être positionnés ou modifiés très facilement par action sur la matrice, traduits automatiquement, et testés immédiatement.

Bien que la conception et la révision d'un modèle IODA ne requière pas de compétences spécifiques en programmation informatique, l'implémentation des parties condition et action de chaque interaction nécessite tout de même l'intervention d'un informaticien, lorsque celles-ci ne font pas déjà partie d'une libraire existante (ce qui est souvent le cas puisque JEDI inclue nativement un ensemble d'interactions génériques).

5.2 Intégration des contenus pédagogiques

L'implémentation d'une interaction peut prendre plusieurs formes, de la plus triviale à la plus complexe. Par exemple, dans FORMAT-STORE, le joueur (dans le rôle d'un `Vendeur`) peut réapprovisionner un produit ou dialoguer avec un `Client`. Alors que l'interaction `Réapprovisionner` consiste simplement à augmenter la quantité de l'article cible, l'interaction `Dialoguer` implique une suite complexe d'opérations comprenant l'utilisation d'une interface spécifique, la capture des répliques sélectionnées par le joueur, l'attribution d'un score et l'affichage d'une notification. De fait, la *modularité des comportements (4)* est assurée par la possibilité de définir des interactions de complexité différente.

De la même manière, la problématique de la *scénarisation (8)*, consistant à intégrer les contenus pédagogiques dans le comportement des PNJ, est rendue accessible par la méthodologie centrée interactions. Dans FORMAT-STORE, les situations-problèmes qui constituent le cœur de la pédagogie, se présentent sous la forme de dialogues contextualisés entre l'apprenant et un client hypothétique. Nous présentons une implémentation volontairement simple, n'ayant pas vocation à doter les PNJ de propriétés propres aux ACA, dans laquelle chaque dialogue est représenté par une interaction entre un `Client` et le `Vendeur`, et son contexte par les préconditions de l'interaction (par exemple : un `Client` ne peut pas accéder à une allée, un `Client` ne peut pas prendre un `Article`), ainsi éventuellement qu'une autre interaction (par exemple, l'`Article Génère une tache` qui obstrue l'allée, l'`Article Périme`).

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

TABLE 2 – La matrice des interactions détaille les interactions possibles entre familles d'agents. Chaque interaction est positionnée dans une cellule dont la ligne représente la famille source et la colonne la famille cible (les interactions dégénérées sont placées dans la colonne ∅). Chaque interaction est associée à une priorité (du point de vue de la source) ainsi qu'à une garde de distance (sauf interactions dégénérées).

Bien que les interactions complexes comme les dialogues semblent au premier abord les plus importantes pour assurer le réalisme du comportement, l'intérêt même d'une simulation MA repose sur la multiplicité des interactions simples, comme expliqué dans la section suivante.

5.3 Comportements réalistes

Au même titre que la scénarisation, le réalisme du comportement des agents autonomes constitue le deuxième aspect de l'immersion de l'apprenant dans le *serious game*.

La vision orientée interactions proposée par IODA offre une implémentation originale du concept d'affordance. Dans [9], Gibson établit que les possibilités d'interaction (les fonctions) d'un objet dans l'environnement sont principalement suggérées par l'objet lui-même (sa forme, sa position, etc). Dans les simulations de l'animation comportementale ou les jeux vidéo où un personnage virtuel doit interagir avec les objets de son environnement, une interprétation de cette théorie est traditionnellement employée. Elle consiste à attacher à chaque objet d'une part les algorithmes de l'interaction (de cette manière l'objet lui-même décrit au PNJ comment il devrait être manipulé, dans quelles circonstances, et quel est le résultat de la manipulation) ainsi que les animations. Par exemple, un PNJ voulant ouvrir une porte est informé par la porte elle-même de la position à laquelle il doit se tenir, de l'animation qu'il doit jouer et de l'état final de la porte. Ce mécanisme permet avant tout d'éviter tout défaut dans l'animation en ajustant avec précaution les positions des entités en interaction et en synchronisant leurs animations respectives. Plus important, il permet de libérer le PNJ de ces connaissances pour se concentrer sur ses objectifs internes et la recherche dans l'environnement d'objets lui permettant de les satisfaire. Traditionnellement cependant, le PNJ reste à l'origine de l'interaction, et toute interaction exécutée est nécessairement le résultat d'un processus de décision souvent complexe et chronophage exécuté par un mécanisme de sélection de l'action cognitif.

L'idée que les capacités cognitives de l'agent puissent également être distribuées au sein des possibilités d'interaction offertes à l'agent constitue un postulat de l'approche SMA, et plus particulièrement de l'approche IODA qui considère qu'un comportement intelligent peut être exprimé par un agent réactif. Cette argument est proche de l'idée développée par Brooks dans [5], postulant que l'intelligence n'est pas le résultat d'une planification ou d'un raisonnement (à l'instar d'un agent mettant en œuvre un script ou un système expert) mais au contraire que l'apparence de processus cognitifs est le résultat du comportement réactif d'un agent dans un environnement dynamique.

Pour illustrer ce principe, prenons l'exemple de l'activité d'un client dans FORMAT-STORE. Un client possède un profil (parmi 10 profils initialement fournis, correspondant à 5 catégories d'âge, multipliés par 2 sexes) ainsi qu'une liste d'articles à acheter (le nombre d'articles étant fonction du profil). Un client virtuel entrant dans le magasin considère toutes les interactions pos-

sibles (décrites dans le tableau 2) par ordre de priorité décroissante jusqu'à en trouver une réalisable (cf. section 4.4). `Sortir` du magasin du magasin n'est possible qu'après avoir réglé ses articles. `Payer` nécessite préalablement d'occuper la première place dans la file d'attente. `Entrer` dans la file d'attente n'est possible que lorsque tous les articles de la liste ont été récoltés. `Prendre` un article suppose d'être à portée de celui-ci. `Aller vers` un article s'appuie sur des algorithmes de *pathfinding* et de navigation, mais n'est possible que lorsque la position de cet article est connue. Le client est `Informé` de la position des articles par les panneaux d'affichage situés dans le magasin. Les panneaux ne peuvent être lus que lorsque le client `Erre` dans le magasin (ou se déplace vers un autre article).

Alors qu'il évolue dans le magasin, le client semble suivre un plan, à l'instar d'un agent cognitif. Pourtant, chaque interaction est indépendante de la suivante et leur séquence découle seulement des priorités qui ont été établies entre elles. Non seulement ce mécanisme permet l'obtention de *comportements réalistes (2)*, mais il garanti aussi leur adaptativité et leur variabilité [10].

Le *comportement adaptatif (3)* du client est exprimé à travers sa robustesse aux perturbations de l'environnement. La capacité des clients à trouver les articles et remplir leur panier ne dépend pas d'une carte prédéfinie du magasin. Les produits peuvent être déplacés, retirés, les allées obstruées, sans que leur capacité à remplir leur objectif ne soit altérée. De même, la faculté des clients à trouver le vendeur dans le magasin, dès qu'un problème survient, et malgré les déplacements imprévisibles du joueur, constitue un autre exemple d'adaptativité.

Du fait de l'adaptativité du comportement des clients, nous défendons l'idée que la différenciation comportementale peut être obtenue sans se reposer uniquement sur l'aléatoire de certaines interactions, telle que `Errer`. Dans les deux exemples du paragraphe précédent, la variabilité des comportements est illustrée par la possibilité d'affecter des listes d'articles différentes à chaque client, ce qui se traduit par un parcours indéterminable du magasin par le client, et la possibilité d'introduire des perturbations dans le magasin durant la session de jeu, de manière aléatoire et sans possibilité d'anticiper le nombre de clients impactés, occasionnant des réactions variées.

L'adaptativité des PNJ et la variabilité de leur comportement sont deux propriétés essentielles lorsque l'élément humain joue un rôle actif dans la simulation.

5.4 L'humain dans la boucle

Dans une *simulation participative (1)*, les agents autonomes cohabitent avec le joueur, dont la modalité et la temporalité des actions sont différentes. En plus de la robustesse du comportement des agents autonomes, l'enjeu consiste à proposer une intégration non bloquante du joueur.

Dans FORMAT-STORE, comme illustré dans le tableau 2, le `vendeur` est présent dans la liste des agents bien qu'il soit l'avatar de l'apprenant dans le jeu. Le contrôle exercé par le joueur est répercuté sur l'agent, qui se déplace ou agit en conséquence, mais en l'absence d'action de la part du joueur, la simulation suit son cours normalement. Une autre conséquence de ce fonctionnement permet à l'agent contrôlé par le joueur d'être substitué par n'importe quel autre agent dans le jeu. Concrètement, si les objectifs pédagogiques en exprimaient la contrainte, il serait aisé de placer le joueur dans la peau d'un `Client` et de le voir confronté par exemple à un `Vendeur` autonome. D'ailleurs, pour tester les comportements des PNJ, nous lançons la simulation MA sans la présence du joueur.

Une autre problématique inhérente à la présence de l'apprenant dans la simulation consiste à proposer une *évaluation flexible (7)* de sa performance. L'évaluation de l'apprenant prend en compte deux variables, pour chaque client. Premièrement, en raison de notre choix dans la scénarisation, l'apprenant est évalué immédiatement à l'issue de chaque dialogue avec un client, et en fonction de ses réponses. La variabilité des comportements des agents, décrite dans la section précédente, garantit que l'apprenant n'est jamais évalué de la même manière (les problèmes des clients sont nombreux et éventuellement soulevés durant leur évolution dans le magasin au gré des problèmes qui surviennent), et jamais au même moment (car chaque rencontre entre l'apprenant et un client est imprévisible). Deuxièmement, et indépendamment, le score du joueur prend aussi en compte le niveau de satisfaction de chaque client qui sort du magasin. La satisfaction d'un client décroît par exemple lorsqu'un article désiré est manquant ou une allée obstruée. Dans ce cas également, le nombre de clients affectés par chaque problème dans le magasin, ainsi que l'impact du problème sur le

client (qui dépend de la tolérance associée au profil du client) sont imprévisibles.

6 Conclusion et perspectives

Nous avons montré dans cet article que les *Serious Games* soulèvent des problématiques de jeu, de pédagogie et de simulation étroitement couplées, qui en font un défi aux approches utilisées classiquement pour la simulation MA. Si chacune des problématiques d'un SG est souvent présente dans une simulation à base d'agents, leur combinaison en revanche met à l'épreuve les qualités de modularité, d'expressivité, de reformulation de modèle, d'intégration dynamique de l'humain, dont font preuve ou non les méthodes de simulation. À travers notre propre expérience de réalisation d'un SG, nous avons illustré ces difficultés et montré comment l'approche orientée interactions (IODA), conçue pour des simulations plus classiques, est particulièrement adaptée pour répondre aux problématiques SG, grâce notamment à une séparation des entités et des comportements ainsi qu'une implémentation originale de la théorie des affordances. Nous avons montré également comment prendre en compte un joueur humain *"plug and play"*, pouvant être remplacé par des agents autonomes ou vice-versa. Cette versatilité suggère que les SG immersifs pourraient constituer une forme de "Test de Turing" pour les simulations MA et la définition de comportements réalistes.

Remerciements

Le projet FORMAT-STORE a été financé par le ministère de l'Économie, des Finances et de l'Industrie, dans le cadre de l'appel à projet "Serious Game" de 2009, en collaboration avec les sociétés Learn'Ingenierie et Idées-3Com. Les images du jeu sont la propriété de Idées-3Com. Les auteurs remercient également Jean-Baptiste Leroy pour son implication dans le projet.

Références

[1] W. B. Arthur, J. H. Holland, B. LeBaron, R. Palmer, and P. Tayler. Asset pricing under endougenous expectations in an artificial stock market. *Economic Notes*, 26 :297–330, 1997.

[2] F. Bellotti, R. Berta, A. De Gloria, and L. Primavera. Enhancing the educational value of video games. *Comput. Entertain.*, 7 :23 :1–23 :18, June 2009.

[3] E. Bonabeau. Agent-based modeling : methods and techniques for simulating human systems. In *Proc. National Academy of Sciences*, volume 99, pages 7280–7287, 2001.

[4] F. Bousquet and C. Le Page. multi-agent simulations and ecosystem management : a review. *Ecological Modelling*, 176(3-4) :313–332, 2004.

[5] R. A. Brooks. Intelligence without reason. In *proceedings of the 1991 International Joint Conference on Artificial Intelligence*, pages 569–595, 1991.

[6] G. Desmeulles, S. Bonneaud, P. Redou, V. Rodin, and J. Tisseau. In-virtuo experiments based on the multi-interaction system framework : the RéISCOP meta-model. *CMES, Computer Modeling in Engineering & Sciences*, 2009.

[7] J. M. Epstein and R. Axtell. *GrowingArtificial Societies : Social Science from the Bottom Up*. Brookings Institution Press, Washington, 1996.

[8] S. Espié. Vehicle driven simulator versus traffic-driven simulator : the INRETS approach. In *Proceedings of driving simulation conference (DSC'99)*, Paris, France, 1999.

[9] J. J. Gibson. *The ecological approach to visual perception*. Hillsdale ; New Jersey ; London, 1979.

[10] Y. Kubera, P. Mathieu, and S. Picault. Interaction-oriented agent simulations : From theory to implementation. In *Proceedings of the 18th European Conference on Artificial Intelligence (ECAI'08)*, pages 383–387. IOS Press, 2008.

[11] Y. Kubera, P. Mathieu, and S. Picault. Everything can be agent ! In *Proceedings of the ninth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'2010)*, pages 1547–1548, 2010.

[12] Y. Kubera, P. Mathieu, and S. Picault. IODA : an interaction-oriented approach for multi-agent based simulations. *Autonomous Agents and Multi-Agent Systems*, pages 1–41, 2011. 10.1007/s10458-010-9164-z.

[13] B. Lacroix, P. Mathieu, and A. Kemeny. The use of norms violations to model agents behavioral variety. In *Coordination, Organizations, Institutions and Norms in Agent Systems IV*, volume 5428 of *Lecture Notes on Artificial Intelligence*, pages 220–234. Springer, 2009.

[14] P. Mathieu, J-C. Routier, and P. Urro. Un modèle de simulation agent basé sur les interactions. In *Actes des Premières Journées Francophones sur les Modèles Formels de l'Interaction (MFI'01)*, pages 407–417, 2001.

[15] D. Panzoli, C. Peters, I. Dunwell, S. Sanchez, P. Petridis, A. Protopsaltis, V. Scesa, and S. de Freitas. Levels of interaction : A user-guided experience in large-scale virtual environments. *Games and Virtual Worlds for Serious Applications, Conference in*, 0 :87–90, 2010.

[16] R. Querrec, P. Reignier, and P. Chevaillier. Humans and autonomous agents interactions in a virtual environment for fire figthing training. *Virtual Reality International Conference*, pages 57–63, 2001.

[17] Y. Shoham. Agent oriented programming. *Journal of Artificial Intelligence*, 60(1) :51–92, 1993.

[18] M. Singh. Conceptual modeling for multiagent systems : Applying interaction-oriented programming. In *Conceptual Modeling*, volume 1565 of *Lecture Notes in Computer Science*, pages 195–210. Springer Berlin / Heidelberg, 1999.