



HAL
open science

Un mécanisme de composition de comportements pour agents virtuels

Quentin Reynaud, Vincent Corruble

► **To cite this version:**

Quentin Reynaud, Vincent Corruble. Un mécanisme de composition de comportements pour agents virtuels. 21èmes Journées Francophones des Systèmes Multi-Agents, Jul 2013, Lille, France. hal-00876094

HAL Id: hal-00876094

<https://hal.science/hal-00876094>

Submitted on 23 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.

Un mécanisme de composition de comportements pour agents virtuels

Q.Reynaud^a

quentin.reynaud@lip6.fr

V.Corruble^a

vincent.corruble@lip6.fr

^a Laboratoire d'Informatique de Paris 6,
Université Pierre et Marie Curie, Paris, France

Résumé

L'objectif de cet article est de présenter un mécanisme permettant l'intégration, dans un même processus décisionnel, de plusieurs théories, sous-systèmes ou architectures d'agents intelligents. Chaque composant ainsi intégré a la capacité de proposer des comportements à un module décisionnel chargé de l'intégration, la composition et l'arbitrage. Ce processus est ensuite évalué sur des scénarios assez simples afin de comparer la pertinence, la richesse et la crédibilité du comportement effectif des agents par rapport aux comportements proposés en entrée. Le but de ce mécanisme est, au minimum, de préserver la qualité des entrées, et autant que possible de produire des comportements jugés plus convaincants que chacune des entrées considérées séparément. Les applications sont nombreuses, mais nous présenterons en conclusion le déploiement de ce mécanisme dans un projet collaboratif de simulation urbaine.

Mots-clés : *Couplage dynamique de modèles, Agent virtuel, Génération de comportement*

Abstract

Increasingly complex AI systems are now often faced with the use of a combination of various theories, subsystems or architectures, whose behavior proposals have to be integrated into one behavior generation mechanism. It is the aim of this paper to present such a behavior integration mechanism. It is then evaluated on a number of small scenarios, with the idea of comparing

some single "input behaviors" with the combined "output behavior" in terms of consistency, richness and credibility. The goal of this mechanism is indeed to at least preserve the quality of the inputs, and whenever possible to produce a behavior that is judged more convincing than any of its inputs taken in isolation. Applications are many, but we sketch in a concluding section the current deployment of this mechanism in a collaborative project for urban simulation and its central unifying role in its agent architecture.

Keywords: *Dynamic models combination, virtual agents, behavior generation*

1 Introduction

La simulation multi-agents est un domaine en pleine expansion qui possède de plus en plus d'applications. L'intelligence artificielle joue un rôle central dans l'élaboration d'agents virtuels, en particulier lorsqu'ils simulent des humains dotés d'un large spectre de comportements et de capacités. Selon leurs objectifs et le but de la simulation, ces agents peuvent être très différents les uns des autres. On peut trouver dans l'état de l'art de nombreuses théories ou architectures très pertinentes pour la modélisation d'agents particuliers. Afin de construire un agent, une option est donc de sélectionner la théorie ou le modèle le plus adapté à l'application visée ; une autre option est de trouver un moyen de les combiner entre elles grâce à un mécanisme particulier. Dans ce papier nous explorons cette deuxième direction et nous proposons un mécanisme permettant de combiner dans la

même simulation des comportements proposés par des modules différents, pouvant être basés sur des théories différentes, et de les intégrer dans un unique processus décisionnel. L'objectif d'un tel système est de produire (en sortie) un comportement jugé plus convaincant que chacun des comportements proposés en entrée pris individuellement. Notre priorité n'est pas de prouver la rationalité du comportement de sortie, ni de montrer son optimalité, mais plutôt d'augmenter sa crédibilité et sa richesse dans le but de le rendre plus convaincant [1]. Nous appelons *richesse comportementale* la largeur du spectre des comportements activables pour un agent (variabilité du comportement de sortie), ainsi que le nombre de paramètres considérés en entrée. En ce qui concerne la *crédibilité*, nous nous baserons sur la définition de Coleridge [2], c'est-à-dire la capacité de notre système à « suspendre l'incrédulité d'un observateur ».

Nous décomposons le processus décisionnel en deux phases : la génération des comportements et la sélection du comportement. Dans notre modèle un unique module décisionnel est en charge de la phase de sélection et est conçu pour fonctionner avec un ensemble quelconque de « modules de haut niveau » en entrée. Ces modules d'entrée proposent des comportements potentiels au module décisionnel qui est chargé de la composition et de l'arbitrage.

Nous soutenons dans cet article que ce mécanisme est capable de produire des comportements bénéficiant de l'apport simultané de différents modules d'entrée, certains parmi ceux-ci pouvant être réactifs, d'autres faisant intervenir de la planification, et le comportement résultant vérifie certains critères de cohérence, d'opportunisme, prenant correctement en compte l'environnement, et capable de compromis. De plus l'architecture ne se limite pas à l'utilisation d'un ensemble prédéfini de modules d'entrée, mais autorise l'intégration de nouveaux modules dans le processus décisionnel.

Nous présenterons d'abord quelques travaux connexes, et nous décrirons le formalisme de notre mécanisme décisionnel. Puis nous donnerons des détails sur notre module décisionnel ainsi que le processus de sélection du comportement. Ensuite, nous décrirons quelques résultats expérimentaux permettant une première étape dans la validation du modèle. Finalement, nous concluons et discuterons de quelques perspectives.

1.1 Travaux connexes

Quelques travaux sont particulièrement utiles à mettre en perspective avec le notre. En particulier dans le domaine des architectures agent de nombreux modèles s'intéressent à la composition de comportements.

Le processus décisionnel de PECS [3] utilise quatre modules qui peuvent être comparés à nos modules d'entrées (ou modules de « haut niveau »). Un module physique gère des variables homéostatiques, un module émotionnel est en charge des états émotionnels de l'agent, un module social s'occupe de la coopération entre agents et un module cognitif des croyances. Mais ces modules sont en conflit permanent afin de prendre le contrôle de l'agent, qui ne peut être guidé que par un seul module à la fois (ce qui facilite grandement le processus de sélection de comportement, mais interdit la coopération entre modules).

Le même problème se pose pour les architectures de type « subsomption » [4] : la hiérarchie des composants dicte un ordre de priorité qui exclut toute collaboration entre module.

Polyscheme [5] a pour objectif de simuler la cognition humaine, et utilise de nombreux modules « spécialistes » fonctionnant sur des modèles différents (règles de production, réseaux bayésiens, etc.). Un mécanisme de focalisation sélectionne un thème d'intérêt et chaque spécialiste donne son opinion sur la question. Ensuite, un module décisionnel intègre tous ces avis et sélectionne un comportement. Si ce processus décisionnel

permet bien une étroite collaboration entre modules hétérogènes, il s'en trouve fortement complexifié à cause de l'hétérogénéité des avis rendus par les spécialistes. Ainsi, l'intégration dans le processus décisionnel de nouveaux spécialistes impose la modification de la fonction de décision.

En plus des architectures agent, d'autres approches sont également intéressantes à prendre en compte. En premier lieu la hiérarchie à libre flux [6]. Dans ce type de processus décisionnel, chaque objectif est décomposé en sous-comportements, jusqu'au niveau des actions élémentaires. Le processus décisionnel est retardé au maximum, ce qui permet d'obtenir des comportements de compromis grâce à la combinaison d'actions élémentaires pouvant appartenir à des comportements différents. Notre mécanisme décisionnel est inspiré de ce principe.

Un autre domaine important à mentionner, proche sémantiquement, mais éloigné de part ses objectifs et les techniques associées, est celui de la « composition de comportement », dont l'objectif est de synthétiser un comportement cible grâce à la combinaison d'un ensemble de comportements connus [7]. Cela mène à la création d'un contrôleur cherchant à accomplir un objectif donné en entrée sans qu'un plan menant à cet objectif n'existe préalablement.

2 Formalisation

2.1 Définitions

Bien que le vocabulaire adopté soit classique [8] nous allons préciser ici certains termes clés afin d'éviter toute incompréhension.

Nous appelons *comportement* un ensemble organisé d'actions qu'un agent peut effectuer au sein d'une simulation. Un comportement peut être décomposé en *sous-comportements* (par exemple « dîner » peut être décomposé en « préparer le dîner », « mettre la table », « manger », etc.). Une *action élémentaire* est une action qui ne peut plus être décomposée : ce sont les actions effectivement exécutées

par les agents. Une *action concrète* est l'exécution (ou l'instanciation) d'une action élémentaire dans l'environnement. Par exemple si « manger au restaurant » est une action élémentaire et R_1 un restaurant, « manger dans R_1 » est une action concrète. Un *plan* est une séquence d'actions élémentaires qui mène à l'accomplissement d'un but. *L'instanciation d'un plan* prend le contexte environnemental en compte, c'est une séquence d'actions concrètes (voir Fig. 1).

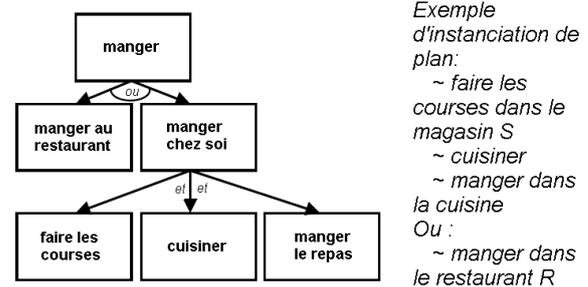


FIG. 1 – Plan et instanciation de plan (S et R sont des emplacements spécifiques dans l'environnement)

Dans notre modèle, un graphe comportemental unique est généré lors du lancement de chaque simulation. Sa définition est laissée à la charge du modélisateur, qui a ainsi la possibilité de l'adapter à ses objectifs. Ce graphe comprend l'ensemble des comportements, et actions réalisables dans la simulation, mais également l'ensemble des plans permettant l'accomplissement de chaque comportement. Dans ce graphe, chaque nœud est un comportement, et chaque feuille une action élémentaire (une action élémentaire est un cas particulier d'un comportement).

2.2 Propositions de comportement

Dans l'architecture proposée dans [10], les modules de haut-niveau prennent en entrée des perceptions de l'environnement et envoient en sortie des propositions de comportement à un module décisionnel chargé de l'intégration, de la décomposition, de la combinaison et de l'arbitrage. Les

modules de haut niveau sont quelconques, et leurs mécanismes internes peuvent donc reposer sur des modèles fortement hétérogènes. Par contre leurs sorties (les propositions de comportement) se font sur un modèle unique, dans un langage commun.

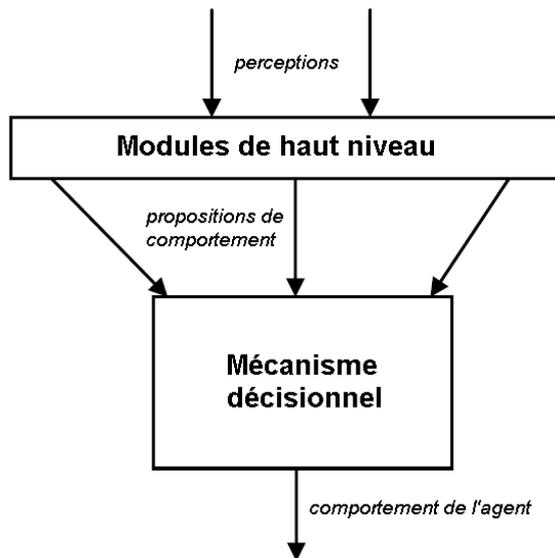


FIG. 2 – Contexte du mécanisme décisionnel

Les modules d'entrée ne doivent pas seulement proposer des comportements potentiels, mais également les prioriser. Etant donné que le module décisionnel doit comparer les différentes possibilités, chaque module de haut-niveau doit ajouter à sa proposition une valeur indiquant l'importance relative du comportement en fonction de ses propres critères. Pour des raisons de simplicité, nous choisissons de sélectionner cette priorité dans l'intervalle $[0,1]$, 0 indiquant l'indifférence et 1 une importance primordiale, du point de vue du module à l'origine de la proposition de comportement, et indépendamment des autres propositions.

2.3 Notations

Ainsi une proposition de comportement se réduit à l'identifiant du comportement désiré plus une priorité. Soit $B = \{B_1, \dots, B_n\}$ l'ensemble des comportements existants, et $P_i = (B_i, p_i)$ une proposition concernant le comportement B_i avec une priorité p_i . Ainsi, $P(A, t) = \{P_1, \dots, P_m\}$ est le vecteur complet des propositions de comportement pour un

agent A au temps t . Ces priorités nous permettent de calculer le niveau de satisfaction de l'agent A : $D_A^t = \sum_{i=1}^n p_i$ avec p_i la priorité de la $i^{\text{ème}}$ proposition. Nous considérons en effet que plus un agent reçoit de propositions de comportements avec de fortes priorités, plus il a de désirs insatisfaits, et donc plus son niveau global de satisfaction est faible.

Notons qu'une action élémentaire est un cas particulier de comportement, ainsi une proposition de comportement peut porter sur un comportement de très haut niveau (i.e. un comportement doté de nombreux niveaux de décomposition en sous-comportements) comme sur une action élémentaire (c'est-à-dire un comportement de très bas niveau).

Les actions sont caractérisées par leur coût par rapport aux ressources considérées dans la simulation (temps, argent, etc.), ainsi que leurs effets (modification de l'inventaire, impact sur l'environnement, etc.). Notons que leurs effets vis-à-vis des modules de haut niveau ne sont connus que des modules eux-même.

Exemple : soit Paul un agent doté de 3 modules d'entrée. M_m un module motivationnel classique [9], M_r un module « réflexe » fonctionnant avec des règles de type perception-action, et M_c un module cognitif gérant des comportements complexes. Lors de son réveil le matin, les différentes propositions de comportement émanent de ces modules pourraient être : « se nourrir », « aller aux toilettes », et « se laver » pour le module motivationnel ; « éteindre le réveil » et « ouvrir les yeux » pour le module de réflexes ; et « gagner de l'argent » pour le module cognitif.

3 Module décisionnel

Ce mécanisme donne de multiples possibilités au modélisateur, mais augmente en contrepartie la complexité du processus décisionnel. Bien que toutes les propositions de comportement partagent le même format, le module décisionnel est confronté au

problème de l'intégration de ces propositions.

3.1 Phase d'intégration

On pourrait penser que le formalisme de sortie partagé par les modules de haut niveau n'est pas suffisant pour comparer directement leurs priorités dans le module décisionnel. En effet, même si l'intervalle est le même (entre 0 et 1), la signification de chaque valeur n'est pas spécifiée. On pourrait également chercher à s'assurer qu'aucun module ne prenne un poids plus important qu'un autre, par exemple en normalisant les priorités sortant de chaque module. De la même manière, on pourrait chercher à vérifier que l'écart type est le même pour chaque module.

Mais nous ne cherchons pas à rendre les différentes priorités directement comparables, tout simplement car elles n'ont pas de raison de l'être. En effet, à quoi sert de chercher à comparer des priorités envoyées par des modules aussi différents qu'un module « instinctif » (une action réflexe), un module « émotionnel » (un désir irrationnel), ou un plan que l'on est en train de suivre ? Le seul moyen de comparer les priorités envoyées par ces différentes sources est de considérer ce choix comme subjectif, propre à l'agent modélisé, et donc de tenir compte de l'individualité de l'agent (qui pourrait être un reflet de sa personnalité).

Ainsi, nous dotons chaque module d'entrée d'un poids indiquant l'importance relative de ce module pour l'agent. Ce poids vient impacter (positivement ou négativement) toutes les priorités sortantes du module correspondant, et nous considérons que tout déséquilibre entre modules fait partie de la modélisation de l'individualité de l'agent. Ces poids peuvent être dynamiques afin de s'adapter à des changements d'état d'esprit de l'agent.

3.2 Algorithme décisionnel

```
Pour un agent a faire
Chaque X pas de temps faire
  Pour chaque Proposition b faire
    Pour chaque Plan p dans b faire
      calculPreferenceMoyenneDuPlan(p,a)
      pour chaque action act dans p
        interrogerEnvironnement(act,Y)
        calculerInstanciationPlan(p)
      pour chaque Instanciation pi faire
        pour chaque critère c faire
          noterInstanciation(pi, c)
    choisirMeilleureInstanciation(a,p)
```

interrogerEnvironnement(act,Y) : nous utilisons un environnement proactif, capable de proposer les Y actions concrètes les plus pertinentes permettant de réaliser l'action act. Pour plus de détails sur cet environnement intelligent, vous pouvez vous référer à [12].

calculerInstanciationPlan(p) : construction de toutes les instanciations de plan grâce à une exploration de toutes les combinaisons possibles des actions concrètes proposées.

choisirMeilleureInstanciation(a,p) : pour chaque plan nous additionnons les notes relatives à chaque critère considéré, après ajustement par le degré d'importance de ce critère pour l'agent a (voir section 4).

calculerCompromis() : la note de compromis d'un plan est calculée en comptant le nombre d'autres propositions de comportement dans laquelle chaque action du plan est utilisée.

noterPlan(p,b,a) : la note du plan est calculée en agrégeant, à travers une fonction sigmoïdale, les paramètres suivants : la priorité de la proposition multipliée par le poids du module origine, la note de la meilleure instanciation, la moyenne des préférences des actions du plan (voir section 4), et la note de compromis.

ChoisirNouveauPlan(a) : le plan préféré est celui ayant la meilleure note. Si ce plan est différent du plan courant, nous ajoutons un bonus d'inertie à la note du plan courant. Cela permet de limiter les problèmes d'oscillations comportementales.

FIG. 3 – Processus de sélection du comportement pour un agent a

Critères : pour l'instant seuls deux critères sont considérés, le temps et le coût, mais il est possible d'en ajouter.

Un agent replanifie son comportement au bout de X pas de temps ou dès qu'il termine une action. En effet, l'exécution d'une action impactant les modules de haut-niveau, les propositions de comportement peuvent être fortement différentes avant et après l'action. La fréquence de re-planification (X) permet

de maîtriser le coût des calculs associés à l'agent, ainsi que de gagner en crédibilité en choisissant une fréquence proche du temps de réaction souhaité pour nos agents.

Exemple : Paul est dans la rue, à midi. Il ne reçoit qu'une seule proposition de comportement : « se nourrir ». Supposons qu'il n'y ait que 2 plans menant à la réalisation de ce comportement, soit « manger au restaurant » soit « manger chez soi » (comportement nécessitant de « faire les courses » et de « faire la cuisine »). La préférence moyenne d'un plan est égale à la moyenne des préférences des actions de ce plan. Le module décisionnel interroge alors l'environnement pour savoir où il peut réaliser chacune des actions présentes dans chacun de ces plans. Il récupère ainsi quelques restaurants et supermarchés (triés selon des critères de distance, coût, préférences, etc.) et calcule l'ensemble des instanciations de plans possibles à partir de ces actions concrètes. Ensuite il sélectionne la meilleure instanciation de chaque plan (selon ses propres critères), regarde s'il existe des actions de compromis (ce n'est le cas ici), puis intègre les différentes priorités, poids, et préférences pour sélectionner la meilleure instanciation (i.e. le meilleur plan, en fonction du contexte et de l'agent).

Ce processus décisionnel peut être confronté à un problème de complexité en cas de multiplication du nombre des modules de haut-niveau. Il est possible de remédier à cela par exemple en réduisant la fréquence de re-planification, ou le nombre d'actions concrètes étudiées pour chaque proposition de comportement.

4 Type d'agents et individualités

Type d'agents

Etant donné que le module décisionnel fonctionne avec un ensemble quelconque de modules en entrée, il est possible de créer des « type d'agents », c'est-à-dire des groupes d'agents ayant des comportements similaires, en modifiant la liste des modules actifs.

Soit $M = \{M_1, \dots, M_n\}$ l'ensemble des modules de haut-niveau et $T = \{T_1, \dots, T_m\}$ l'ensemble des différents types d'agents. Nous appelons M_{T_i} l'ensemble des modules utilisés pour les agents du type T_i .

Individualités

Définir des types d'agents n'est pas suffisant. Afin d'obtenir suffisamment de variabilité comportementale il est nécessaire de pouvoir individualiser les agents. Dans notre modèle nous donnons aux agents des préférences concernant chaque action ou comportement. Nous les dotons également de « degrés d'importance », indiquant l'importance de chaque critère défini. D'autres paramètres entrent également en compte, comme les poids des modules actifs (voir section 3.1), ou les différents bonus utilisés lors de la phase de sélection du comportement (voir section 3.2) : inertie, compromis, etc.

5 Expérimentations

Nous avons réalisé quelques expérimentations pour commencer la validation du modèle.



FIG. 4 – Aperçu de l'environnement

Toutes les expérimentations ont été réalisées dans le même environnement, à savoir une représentation réaliste en 3 dimensions de la rue Soufflot à Paris. Nous avons défini des points d'intérêt à l'intérieur de ce quartier (restaurants, cafés, magasins, etc.) dans lesquels les agents peuvent agir. Cet environnement a été créé dans le cadre du projet Terra Dynamica¹, un projet largement collaboratif visant à la modélisation et la simulation d'humains virtuels en milieu urbain.

¹ <http://www.terradynamica.com/>

Dans nos expérimentations, nous avons utilisé trois modules de haut-niveau assez simples. Un module motivationnel basé sur des modèles éthologiques [9], proposant des comportements liés à des motivations basiques (faim, soif, fatigue, ennui) ; un module de mission qui propose des comportements scriptés ; et un module aléatoire, proposant un comportement aléatoire avec une priorité aléatoire jusqu'à ce qu'il soit exécuté.

Nous avons défini trois types d'agents :

- Des *touristes*, dont le seul module d'entrée actif est le module motivationnel. Les touristes visitent Paris et vaquent à leurs occupations.
- Les *policiers*, eux-aussi n'ont qu'un seul module actif, le module de mission. Les policiers patrouillent dans la zone, et ne se préoccupent de rien d'autre.
- Les *policiers dotés de motivations*, eux, possèdent 2 modules de haut-niveau actifs : le module de mission et le module motivationnel. Ainsi ces agents doivent patrouiller dans le quartier, mais également satisfaire leurs motivations élémentaires.

5.1 Simulations

Impact de la combinaison de comportements

Nous avons lancé 10 simulations de 30 agents (10 touristes, 10 policiers et 10 policiers dotés de motivations). La FIG. 5 montre le temps moyen mis par les agents de chaque type pour accomplir tous leurs buts. Notons que pour les policiers avec motivations, les poids des modules sont identiques.

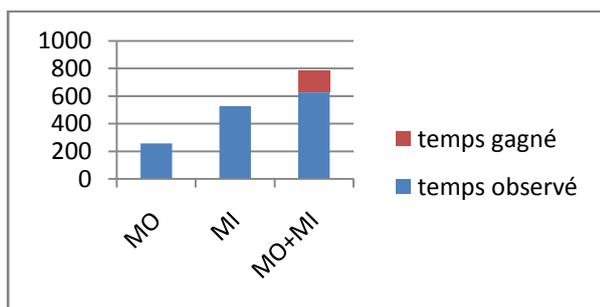


FIG. 5 – Temps nécessaire à l'accomplissement de tous les désirs, en fonction des modules d'entrée actifs (MO : motivations, MI : mission)

Les touristes ont besoin d'environ 250 secondes² pour satisfaire toutes leurs motivations. Ils mangent au restaurant, vont au cinéma, etc. D'un autre côté, les policiers prennent 525 secondes pour faire leur patrouille (c'est-à-dire faire le tour du quartier à pieds). Enfin, les policiers avec motivations mettent autour de 625 secondes pour terminer la patrouille tout en satisfaisant leurs motivations. Le temps mis pour patrouiller est incompressible (il y a un tracé à suivre), ce qui signifie que les policiers avec motivations ont eu besoin de 100 secondes pour satisfaire leurs motivations (environ 40% du temps mis par les touristes). Pour ce faire ils ont dû faire preuve d'opportunisme et profiter du fait que leur patrouille passe proche d'un point d'intérêt pour interrompre leur mission et exécuter une action « motivationnelle ».

Sans ajouter de module de coordination, ni développer de capacité de raisonnement, mais en combinant deux modules d'entrées très simples, le comportement en sortie gagne crédibilité et cohérence.

Impact des poids des modules

Dans cette expérimentation nous nous concentrons sur les policiers avec motivations, et observons les modifications de leur comportement lorsque les poids associés à leurs deux modules changent. Nous avons lancé 10 simulations avec 10 agents de chaque groupe G_1 , G_2 , G_3 , et G_4 , dont les poids sont indiqués dans le tableau ci-dessous.

	Poids du module motivationnel	Poids du module de mission
G_1	0,8	0,2
G_2	0,6	0,4

² Afin de réduire le temps de simulation, nous avons choisi des durées d'actions non réalistes.

G ₃	0,5	0,5
G ₄	0,2	0,8

FIG. 6 – Poids des modules

La FIG. 7 indique les pourcentages d'agents terminant leur patrouille avant d'avoir satisfait toutes leurs motivations, et inversement. Comme nous pouvons le constater, le poids des modules impacte correctement les comportements (plus un module a un poids fort, plus les comportements liés à ce module sont prioritaires). Si des agents sont dotés d'un module, quel qu'il soit, dont le poids est de 0,8, ils termineront les directives de ce module en premier. Nous constatons également que lorsque les deux modules ont le même poids, la majorité des agents préfère terminer la patrouille en priorité. A priori cela peut sembler problématique, voire incohérent. En effet, si des agents ont le même niveau d'intérêt pour deux choses différentes, ils devraient choisir entre elles de manière équilibrée. Mais en réalité, il n'y a aucune raison d'obtenir un quelconque équilibre ici. Les poids des modules n'impactent que les priorités sortantes de ces modules. Les paramètres internes des modules de haut-niveau ne peuvent pas être pris en compte (puisque nous avons justement choisi de les rendre opaques), pas plus que les paramètres intervenant lors du processus décisionnel (qui dépendent du contexte, des préférences des agents, des caractéristiques des actions, etc.). Dans ce cas concret, une des raisons pour lesquelles les agents préfèrent la mission aux motivations c'est que les actions liées à la patrouille sont gratuites, contrairement aux actions liées aux motivations, qui sont la plupart du temps payantes !

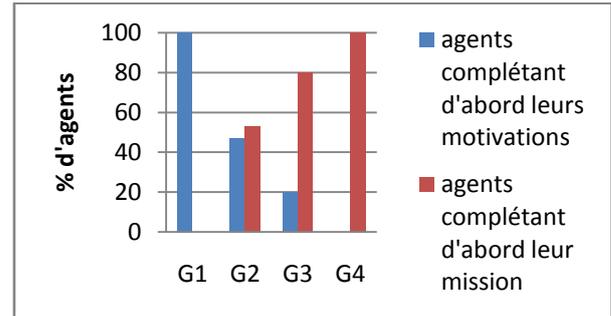


FIG. 7 – Impact du poids des modules

Par contre, si c'est important pour le modélisateur d'obtenir des comportements plus équilibrés, il est possible d'ajuster les poids grâce à une phase d'apprentissage (qui pourrait se servir du niveau de satisfaction comme fonction de récompense).

Impact du bruit résultant d'une entrée aléatoire

Nous avons lancé une dernière série d'expérimentations afin d'évaluer l'impact d'un module aléatoire (et donc de « mauvaise qualité ») sur le comportement des agents. Nous avons ainsi créé quatre groupes d'agents (A, B, C, D) possédant les trois modules de haut niveau (motivation, mission, et aléatoire) avec des poids différents (voir FIG. 8).

	Poids du module motivationnel	Poids du module de mission	Poids du module aléatoire
A	0,5	0,5	0
B	0,4	0,4	0,2
C	1/3	1/3	1/3
D	0,25	0,25	0,5

FIG. 8 – Poids des modules de motivation, mission et aléatoire

Nous avons lancé 10 simulations de 10 agents de chaque groupe et comparé les temps et coûts moyens de leurs comportements (FIG. 9).

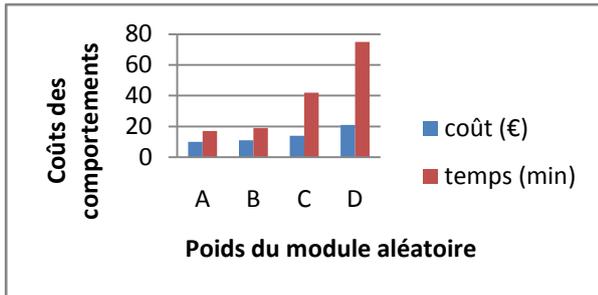


FIG. 9 – Impact d'un module aléatoire

Les agents des groupes A et B ont des résultats très similaires. Lorsque le poids du module aléatoire est faible, il n'impacte que très peu le comportement. Par ailleurs, le coût moyen augmente faiblement avec l'augmentation du poids du module aléatoire (coût multiplié par 2 entre les groupes A et D). En étudiant en détails le comportement des agents du groupe D, on constate qu'ils ont réalisé environ deux fois plus d'actions que ceux des groupes A et B. Un coût double est donc tout à fait normal.

Par contre, le temps moyen augmente de manière beaucoup plus importante : multiplié par environ 4,5 entre les groupes A et D. Cela s'explique par le fait que le module aléatoire ne prend pas correctement l'environnement en compte, il peut ainsi proposer des actions réalisables uniquement dans des lieux peu accessibles. De plus, l'introduction d'un module aussi imprévisible empêche la planification de fonctionner correctement, et aboutit à une très forte augmentation du nombre de déplacements inutiles (ainsi qu'à l'apparition d'oscillations comportementales).

5.2 Discussion

Ces expérimentations permettent une évaluation préliminaire de notre modèle. Notre processus décisionnel autorise la modélisation d'agents réactifs, opportunistes, capables de replanifier leur comportement et d'opter pour des actions de compromis.

Nous avons montré que nous sommes capables de gagner en crédibilité grâce à un mécanisme de composition comportementale, que le poids des modules influence de

manière correcte les comportements, et que la combinaison des modules d'entrée permet de réduire l'impact d'un module « bruité ».

Un résultat surprenant de la dernière expérimentation est que le comportement des agents du groupe C a été parfois jugé plus crédible que celui des agents du groupe A (voir FIG. 8). Cela est dû au fait que des comportements sous-optimaux sont facilement interprétables par un observateur extérieur comme des erreurs typiquement humaines (oubli, fausse manipulation, étourderie, etc.), voire comme des comportements résultants d'émotions.

6 Conclusion et perspectives

Dans cet article nous avons présenté un mécanisme permettant la composition de comportements, qui préserve la qualité des entrées et autorise un gain de richesse et cohérence en augmentant la profondeur des comportements. De plus amples expérimentations seront nécessaires afin de valider et mesurer l'impact positif de cette composition sur la crédibilité des comportements.

Ce mécanisme est actuellement en train d'être déployé [10] au sein du projet Terra Dynamica visant à l'élaboration d'un ensemble d'outils d'IA permettant la modélisation et la simulation d'humains virtuels en environnement urbain. Ce projet a de nombreuses applications notamment dans le jeu vidéo, la sécurité, le transport et l'urbanisme.

En termes de perspectives, nous avons réfléchi à la création de différents niveaux de complexité dans la sélection du comportement, afin d'adapter la profondeur du comportement d'un agent selon son importance relative dans le scénario (et économiser du temps du calcul en rendant les agents jugés peu importants moins gourmands lors des phases de génération comportementale).

D'un autre côté, si les expérimentations nous

encouragent à penser qu'avec des modules d'entrées très simples, les comportements des agents peuvent être jugés cohérents, voire crédibles, il manque tout de même des éléments pour obtenir des comportements vraiment réalistes. En effet, tous les modules d'entrée qui ont été testés sont « réactifs » : les comportements proposés en sortie sont issus de la fluctuation de variables internes, à causes de perceptions ou de stimuli. Les comportements se font en réponse à des informations venant de l'environnement. Pour modéliser des humains virtuels réalistes il y a besoin de capacités supplémentaires, notamment celles permettant la manipulation d'informations incertaines, ou la projection dans le futur. C'est-à-dire qu'il manque à nos agents actuels des modules d'entrée plus « cognitifs » [11], sur lesquels nous travaillons actuellement.

Remerciements

Nous remercions les financeurs du projet Terra Dynamica : la Direction générale de la compétitivité de l'industrie et des services, le Conseil Régional d'Aquitaine, le Conseil Régional d'Ile de France, la Mairie de Paris, le Conseil général des Yvelines, le Conseil général de Seine Saint-Denis, et le Conseil général du Val d'Oise ; ainsi que l'ensemble des partenaires : BeTomorrow, DAVI, le Conservatoire des Arts et Métiers, l'Institut français des sciences et technologies des transports, de l'aménagement et des réseaux, Kylotonn Games, Star-Apic, Thales Services, Thales Training & Simulation, l'Université Pierre et Marie Curie/LIP6, et l'Université Paris 8/CITU.

Références

- [1] P. Sengers, "Designing comprehensible agents," in *Proceedings of the 16th international joint conference on Artificial intelligence - Volume 2*, 1999, pp. 1227–1232.
- [2] S. T. Coleridge, *Biographia Literaria*. 1817.
- [3] B. Schmidt, "Human factors in complex systems: The modelling of human behaviour" in *Simulation in wider Europe, 19th European Conference on Modelling and Simulation*, 2005, pp. 5–14.
- [4] R. A. Brooks, "A Robust Layered Control System For a Mobile Robot," Massachusetts Institute of Technology, Cambridge, MA, USA, 1985.
- [5] N. L. Cassimatis, "Integrating Cognitive Models Based on Different Computational Methods," in *Proceedings of the Twenty-Seventh Annual Conference of the Cognitive Science Society*, 2005.
- [6] T. Tyrrell, "Computational Mechanisms for Action Selection," University of Edinburgh, 1993.
- [7] N. Yadav and S. Sardina, "Decision theoretic behavior composition," in *The 10th International Conference on Autonomous Agents and Multiagent Systems*, 2011.
- [8] S. J. Russell and P. Norvig, *Artificial intelligence: a modern approach*. Prentice hall, 2010.
- [9] E. de Sevin, "An action selection architecture for autonomous virtual humans in persistent worlds," Ecole polytechnique fédérale de Lausanne, 2006.
- [10] E. de Sevin, Q. Reynaud, and V. Corruble, "FlexMex: Flexible Multi-Expert Meta-Architecture for Virtual Agents," in *Advances in Cognitive Systems (ACS'12)*, 2012.
- [11] Q. Reynaud, E. de Sevin, J.-Y. Donnart, and V. Corruble, "A cognitive module in a decision-making architecture for agents in urban simulations," in *Workshop on Cognitive Agents in Virtual Environments (CAVE) Autonomous Agents and Multi-Agent Systems (AAMAS'12)*, 2012.
- [12] Harkouken-Saiah Kenza, Sabouret Nicolas, and Donnart Jean-Yves, "Modélisation de l'information sémantique pour la simulation d'un environnement virtuel sémantique urbain," in *23 ème Journées Francophones d'Ingénierie des Connaissances*, 2012.