



Méta-modèle d'agent pour la génération de comportements variables induits par des modèles cognitifs modulaires

Tristan de Blauwe, Nicolas Sabouret, Domitile Lourdeaux

► To cite this version:

Tristan de Blauwe, Nicolas Sabouret, Domitile Lourdeaux. Méta-modèle d'agent pour la génération de comportements variables induits par des modèles cognitifs modulaires. 30th Journées Francophones sur les Systèmes Multi-Agents (JFSMA 2022), Jun 2022, Saint-Etienne, France. pp.103-112. hal-03752153

HAL Id: hal-03752153

<https://hal.science/hal-03752153>

Submitted on 16 Aug 2022

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.

Méta-modèle d'agent pour la génération de comportements variables induits par des modèles cognitifs modulaires

T. de Blauwe^a
tristan.de-blauwe@hds.utc.fr

N. Sabouret^b
nicolas.sabouret@universite-paris-saclay.fr

D. Lourdeaux^a
domitile.lourdeaux@hds.utc.fr

^a Alliance Sorbonne Université,
UTC Heudiasyc, CNRS, UMR 7253, Compiègne, France

^bLaboratoire Interdisciplinaire des Sciences du Numérique,
LISN, CNRS, UMR 9015, Université Paris-Saclay, France

Résumé

Cet article se place dans le contexte du développement d'agents virtuels pour la formation professionnelle. Nous proposons un méta-modèle permettant l'intégration de modèles cognitifs dans un modèle d'agent, avec un minimum d'hypothèses. La définition d'un agent se fait de manière modulaire à partir d'opérations de base définissant le modèle d'agent et d'un ensemble de processus cognitifs qui complètent ou modifient ces opérations. Nous présentons une première évaluation avec un modèle d'agent et trois modèles cognitifs. Nous montrons que les modèles cognitifs impactent le comportement de l'agent de manière cohérente par rapport aux propriétés des modèles cognitifs introduits.

Mots-clés : Méta-Modèle d'agent, Agent cognitif, modèles cognitifs, comportements

Abstract

This paper's context is the development of virtual characters for training purposes. We propose a meta-model to integrate cognitive models in an agent model, with a minimum of assumptions. The definition of an agent is done in a modular way from basic operations defining the agent model and from a set of cognitive processes that complete or modify these operations. We present a first experimentation with one agent model and three cognitive models. We show that the cognitive models impact the agent's behaviour in a way that is consistent with the properties of the introduced cognitive models.

Keywords: Agent meta-model, cognitive agent, cognitive models, behaviours

1 Introduction

Les Personnages Virtuels Autonomes (PVAs) sont des agents évoluant au sein d'un Environne-

ment Virtuel (EV). Ils interagissent avec leur environnement, ainsi qu'avec l'utilisateur, généralement représenté par un avatar. Une application de ces PVAs consiste à simuler des opérateurs humains, de divers métiers, dans un contexte d'entraînement professionnel [2, 14, 21]. Les agents doivent alors exprimer des comportements complexes, i.e cognitifs, émotionnels et/ou sociaux, afin de reproduire les comportements observés. Le comportement des PVAs est observable au travers des actions qu'ils réalisent sur les éléments de l'environnement.

Ces actions sont le produit d'un modèle d'agent, défini par un ensemble d'*opérations* c'est-à-dire de mécanismes de traitements d'informations, allant de la perception à la réalisation d'actions, suite à une prise de décision, en passant par la communication avec les autres agents [1].

Pour mettre en oeuvre différents comportements, les informaticiens implémentent, au sein d'un modèle d'agent, un ou plusieurs modèles cognitifs, généralement inspirés des Sciences Humaines et Sociales (SHS). Ces modèles définissent la manière dont les opérations déterminent les actions et les communications à effectuer en fonction des perceptions. Par exemple, l'architecture BEN [3] intègre plusieurs modèles cognitifs au sein d'une architecture d'agents BDI [4]. Elle lie des modèles cognitifs, comme OCC [20] pour les émotions, au modèle de personnalité OCEAN [5] afin d'obtenir des comportements cognitifs, affectifs et sociaux. Une autre architecture comme EMA [12] intègre une composante émotionnelle avec la théorie de l'évaluation cognitive des émotions de Smith et Lazarus [22], au sein de l'architecture d'agents SOAR [16].

Cependant, dans un contexte de simulation, les comportements et les modèles cognitifs des agents dépendent fortement du domaine d'ap-

plication et des objectifs de la simulation (qu'il s'agisse d'un scénario de formation, d'un jeu, etc.). Concevoir une architecture de PVAs générique est donc difficile. Tout d'abord, on ne connaît pas a priori les modèles cognitifs à intégrer et à combiner, ainsi que les comportements à générer. Qui plus est, les différentes combinaisons de modèles impactent le comportement à différents niveaux, mais le résultat doit respecter deux propriétés. Premièrement, la *sensibilité* des comportements aux modèles cognitifs, c'est-à-dire le fait que l'intégration des modèles impactent le comportement d'un agent de manière cohérente vis-à-vis des propriétés de ces modèles. Deuxièmement, l'*intelligibilité* des comportements, c'est-à-dire que l'utilisateur de la simulation doit pouvoir qualifier les séquences d'actions qui traduisent l'expression d'un comportement (par exemple : « l'agent semble stressé » ou « l'agent respecte la réglementation »).

Ainsi, notre objectif est de concevoir un méta-modèle qui répond à cette double difficulté : permettre l'intégration de modèles cognitifs dans un modèle d'agent, avec un minimum d'hypothèses sur le modèle d'agent et sur le modèle cognitif, tout en s'assurant que les comportements produits restent sensibles aux modèles et intelligibles. Toutefois, dans cet article, nous évaluons uniquement la sensibilité, soit la capacité du modèle à impacter les comportements à partir des modèles cognitifs activés. Par exemple, en situation de crise, les médecins secouristes peuvent être amenés à utiliser un chiffon pour gagner de précieuses secondes au lieu de mettre des gants pour faire un point de compression. Nous voulons évaluer que l'ajout d'un modèle cognitif impactant le respect de la réglementation permet effectivement l'apparition d'un tel comportement.

Le plan de l'article est le suivant. Dans la section 2, nous décrivons l'état de l'art sur les approches et modèles cognitifs utilisés en simulation. Dans la section 3, nous présentons le méta-modèle. Dans la section 4, nous appliquons notre approche sur un projet de formation médicale. Dans la section 5, nous concluons sur les avantages et inconvénients de notre approche, ainsi que les perspectives envisageables.

2 État de l'art

Nous sommes intéressés par l'opérationnalisation de modèles cognitifs au sein d'un modèle d'agent afin d'impacter sensiblement le compor-

tement de manière intelligible. Nous présentons d'abord ce qui caractérise ces modèles cognitifs et nous montrons en quoi leur opérationnalisation est difficile. Nous détaillons ensuite quelques modèles cognitifs qui seront utilisés dans la suite de l'article. Enfin, nous présentons quelques modèles d'agent classiques en simulation multi-agent et la manière dont ils ont intégré des modèles cognitifs.

2.1 Modèles cognitifs

Selon le dictionnaire de psychologie de l'Association Américaine de Psychologie ¹ (APA), un modèle cognitif est une représentation simplifiée et théorique d'un ensemble de processus cognitifs/mentaux amenant à des phénomènes observables et qui fournit des explications sur la cause. Selon la même source, un processus cognitif est n'importe quelle fonction impliquée dans des activités telles que l'attention, la perception, l'apprentissage et la résolution de problèmes.

La littérature en SHS propose un grand nombre de modèles cognitifs selon les processus à modéliser. Dans le domaine des processus socio-affectifs, par exemple, on peut citer la théorie de l'évaluation cognitive des émotions proposée par Smith et Lazarus [22], le modèle OCEAN [5] pour la personnalité ou en encore le modèle OCC [20] très souvent utilisé en informatique pour opérationnaliser l'évaluation cognitive [20].

L'étude des travaux en informatique, qui implémentent de tels modèles, met en évidence la difficulté de généraliser l'opérationnalisation d'un modèle cognitif donné. Ainsi, le même modèle OCC peut être implémenté de différentes manières selon les modèles d'agent utilisés (par exemple, un modèle BDI [3] ou un modèle à base de règles [19]). En effet, un modèle peut impacter différentes opérations au sein du modèle d'agent, que ce soit pour la perception, pour la délibération, pour la mise à jour des données, pour la sélection d'action ou pour la communication. Il est donc difficile de proposer une opérationnalisation sur une seule de ces dimensions, car la manière dont on implémente le modèle cognitif pour une opération de perception peut impacter la manière dont il faut altérer le processus de décision. Cela est d'autant plus vrai que l'on souhaite être modulaire par rapport aux modèles cognitifs, tout en satisfaisant la sensibilité de chacun et l'intelligibilité du comportement.

1. <https://dictionary.apa.org>

2.2 Modèles cognitifs

Nous présentons dans cette section quelques modèles cognitifs pour illustrer notre approche.

Le premier modèle est celui proposé par Driskell et Salas [7], qui ont réalisé des études conséquentes sur les comportements d'équipes en situation extrêmes. Ils proposent ainsi « un nombre limité de mécanismes cognitifs, émotionnels et sociaux à travers duquel le stress impacte les performances » des membres d'une équipe. Un des phénomènes décrit est que le stress augmente la distraction et décroît l'attention, ce qui pourrait entraîner des erreurs de manipulation ou des oublis de la part d'un agent. Ici, on observe que le modèle cognitif impacte directement la perception de l'agent, les conditions d'exécution et indirectement la sélection d'action de par ses connaissances partielles erronées. Un autre mécanisme proposé dans le modèle de Driskell et Salas pourrait impacter d'autres opérations de l'agent : « sous stress, les personnes ont tendance à moins s'entraider. » [8]. L'opérationnalisation de ce modèle au sein d'un modèle d'agent présente donc un double défi, d'une part en raison de la variété des comportements décrits et d'autre part parce qu'il intervient sur les différentes opérations du modèle d'agent.

Ensuite, afin de combiner ce modèle avec d'autres, nous considérons des modèles déjà présents dans les travaux précédents de notre équipe. D'une part, nous avons les comportements de « *suiveur* » décrits par Demary [6] et résumés dans le tableau 1. En fonction de son profil, l'agent réagit différemment lorsqu'il reçoit un ordre de son supérieur (voir tableau 1).

Enfin, nous avons les activités limites tolérées par l'usage [10], amenant à des comportements non-réglementaires. Par exemple, une infirmière expérimentée prescrit un anti-douleur, sans l'accord du leader, pour faire face à l'urgence de la situation.

2.3 Modèles d'agent en simulation

Un modèle d'agent consiste à concevoir une boucle perception-raisonnement-action qui produit un comportement. L'opérationnalisation de modèles cognitifs au sein de ce modèle permet de générer des comportements plus humains et intelligibles.

Par exemple, REPLICANTS [15] est un modèle d'agent raisonnant sur le formalisme

	Communicant	Non-Communicant
Proactif	Je raisonne, j'informe le leader, je conteste si besoin, j'obéis au final (Exemplaire / Courageux)	Je prend des décisions, je conteste (Aliéné / Individualiste)
Passif	Je demande quoi faire et je fais ce qu'on me dit (Mouton / Faiseur)	Si on ne me dit rien je suis la procédure. Sinon, je fais ce qu'on me demande (Yes people / Ressource)

Tableau 1 – Tiré de [6]. Décrit le comportement en fonction du style *Suiveur* et *Communicant*

ACTIVITY-DL, inspiré des HTNs [9]. Ce formalisme modélise l'activité humaine de manière hiérarchique, proche de la représentation mentale des experts. Le parcours de l'arbre d'activité retourne un ensemble de tâches envisageables, ainsi que leur priorité, évaluée selon les préconditions. L'intégration de modèles cognitifs dans REPLICANTS se fait en modifiant ce calcul de priorité par l'ajout de bonus/malus en fonction du profil de l'agent et des propriétés de la tâche considérée. Par exemple, l'incorporation du modèle Demary [6] ajoute un bonus lorsque la tâche correspond à un ordre et que l'agent a un profil passif. Le modèle de Fadier [10] applique un malus selon les conditions réglementaires et le profil de l'agent. Intégrer de nouveaux modèles cognitifs nécessite donc de réviser ce calcul de priorité. La difficulté est alors de s'assurer que les modèles restent toujours impactants et que les comportements restent intelligibles.

Un autre modèle souvent utilisé dans les systèmes multi-agents est le modèle BDI [4]. Ainsi, dans la communauté JFSMA, nous avons cité en introduction les travaux de Bourgaïs *et al.* [3]. Le choix de conception qui est fait consiste à relier les modèles cognitifs aux paramètres du modèle OCEAN, ce qui limite les paramètres à définir. Une force de cette architecture est d'offrir une certaine modularité : le modélisateur peut choisir de n'utiliser que la prise de décision sans tenir compte de l'évaluation d'émotions par exemple. Toutefois, l'ensemble des opérations dépend de l'approche BDI : il n'est pas possible d'envisager une autre architecture de prise de décision.

2.4 Synthèse

Bien qu'il existe de nombreux modèles d'agent proposant une implémentation et une combinaison fine de différents modèles cognitifs, les choix de conception qui sont faits contraignent le modélisateur à s'adapter au modèle. À mesure que les besoins évoluent, les utilisateurs peuvent vouloir mettre en œuvre d'autres modèles d'agents pour la prise de décision ou d'autres modèles cognitifs pour produire de nou-

veaux comportements. Ces modifications sont difficiles, voire impossibles, dans ces modèles.

C'est pourquoi nous proposons dans cet article un méta-modèle d'agent, c'est-à-dire un modèle pour combiner plusieurs modèles cognitifs au sein d'un modèle d'agent. L'ensemble d'opérations du modèle d'agent est donc modifié et complété selon la combinaison de modèles cognitifs considérés. Notre méta-modèle doit permettre de programmer n'importe quel modèle d'agent.

Pour illustrer notre travail dans un contexte de formation, nous utilisons le modèle d'agent **REPLICANTS** [15] et les modèles cognitifs de Demary [6], de Fadier [10] et Driskell [8]. Nous montrerons comment notre méta-modèle permet de les intégrer simplement, d'une manière qui peut se généraliser à n'importe quel autre modèle d'agent et n'importe quels autres modèles cognitifs.

3 Méta-modèle OPACK

Le méta-modèle OPACK est un modèle de modèle d'agent pour la génération de comportements induite par des modèles cognitifs, dans un contexte de simulation interactive. Nous cherchons à faciliter l'intégration de modèles cognitifs, tout en obtenant des comportements intelligibles. Comme nous le verrons, ce modèle fait le moins d'hypothèses possibles sur le modèle d'agent et sur les modèles cognitifs utilisés.

3.1 Cas d'étude

Pour présenter le fonctionnement de notre méta-modèle, nous nous appuierons sur le cas d'étude suivant : nos agents sont des médecins secouristes. Ils doivent réaliser une procédure médicale sur un patient. Les observations terrain et les retours d'experts indiquent que cette procédure varie selon les scénarios et l'état des agents.

Notre objectif est rendre compte de cette variabilité grâce à des modèles cognitifs. Nous nous fondons sur un modèle d'agent **REPLICANTS**, déjà utilisé pour ce cas d'étude.

3.2 État d'un agent

L'état d'un agent est composé des éléments suivants :

P : ensemble des percepts, soit ce qui est actuellement perceptible par l'agent (mis à jour à chaque cycle d'exécution);

A : ensemble des actions actuellement menées par l'agent;

C : ensemble des caractéristiques de l'agent, pouvant représenter des variables physiologiques ou psychologiques comme le stress, ou bien des paramètres stables dans le temps comme la personnalité ou les qualifications;

K : ensemble des connaissances de l'agent, qui comprend non seulement les croyances de l'agent mais aussi les paramètres des actions, les plans éventuels, *etc.*

Le modélisateur définit chacun de ces éléments en fonction du domaine, mais il n'a pas l'obligation de tous les utiliser. Par exemple, il est possible de programmer un agent réactif dans lequel **K** sera vide.

3.3 Opérations

La brique de base de notre méta-modèle est la notion d'*opération*. Une opération est une fonction avec des entrées, une seule sortie et un rôle. L'ensemble d'opérations d'un agent est noté **O**.

Chaque opération a accès à l'ensemble de l'état de l'agent. Les entrées et la sortie sont caractérisées par un type et une valeur — par exemple, l'action (type) « poser un garrot » (valeur) ou le percept (type) « le patient est sain » (valeur). Le cas échéant, il est possible de définir des sous-types — par exemple, l'état du patient est un sous-type de percept.

Le rôle de l'opération définit sa signature (soit ses entrées et sorties). Les noms des rôles proviennent des fonctions mentales décrites par l'APA² :

- **Manipulation** : aucune entrée et sortie (opère uniquement à partir de l'état de l'agent);
- **Acquisition** : aucune entrée mais une sortie;
- **Storage** : des entrées mais aucune sortie (agit sur l'état de l'agent);
- **Interpretation** : des entrées et une sortie;
- **Transformation** : cas particulier d'interprétation où une des entrées est de même type que la sortie : c'est notre donnée manipulée. Les autres entrées ainsi que l'état permettent de contrôler la modification apportée à la donnée manipulée par l'opération.

2. <https://dictionary.apa.org>

Exemple Pour implémenter le modèle d'agent REPLICANTS dans le méta-modèle OPACK, une opération importante est la sélection d'action (sortie) à partir d'actions candidates (entrée) qui est une opération d'*interprétation*. Une autre opération importante est l'acquisition d'actions envisageables, notamment grâce au parcours de l'arbre qui permet d'extraire des actions candidates en fonction de l'état de la simulation. Cette opération d'*acquisition* ne reçoit pas d'entrée, mais utilise l'état de l'agent (dont l'arbre ACTIVITY-DL qui est dans **K** et les caractéristiques de l'agent **C**) pour calculer un ensemble d'actions.

Ces deux opérations sont illustrées sur la figure 2. Comme on le voit sur cette figure, la sortie d'une opération est reliée aux entrées d'autres opérations : nous parlons de *flux d'opérations*.

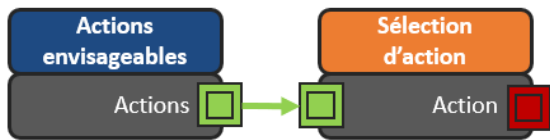


FIGURE 2 – Exemples de deux opérations pour implémenter le modèle d'agent REPLICANTS.

3.4 Flux d'opérations

Le modélisateur définit les opérations et les relie en un flux d'opérations, à la manière de la programmation par flux de Morrison [18].

Concrètement, un flux d'opérations est un graphe orienté acyclique dont les sommets sont des opérations et dont les arcs représentent les flux de données entre la sortie d'une opération et les entrées des suivantes. En complément, le modélisateur peut définir des arcs de précedence entre deux opérations, qui ne correspondent pas à un échange de données mais à un ordre d'exécution.

La figure 3 page suivante donne un exemple de flux. Les couleurs des étiquettes des sommets correspondent aux rôles des opérations, et les couleurs des arcs aux types des données échangées. Naturellement, une sortie ne peut être reliée qu'à des entrées de même type. Les arcs de précedence sont dessinés en gris pointillé.

3.5 Cycle procédural de l'agent

L'exécution d'un agent suit le cycle perception, délibération, action, comme illustré sur la figure 4 :

Perception. L'environnement communique à l'agent l'ensemble **P** de percepts. OPACK n'impose aucune contrainte sur la manière dont cet ensemble est construit et mis à jour par l'environnement. Pour gérer les modifications concurrentes, l'agent travaille sur une copie **P'** de **P**. L'agent ne modifie jamais **P** (seul l'environnement définit les percepts).

Délibération : exécution du flux (voir ci-après), qui modifie l'état de l'agent en agissant possiblement sur les ensembles **P'**, **A**, **C** et **K**.

Action : l'ensemble **A**, éventuellement modifié suite à l'exécution du flux, est utilisé par l'environnement qui est responsable de l'exécution des actions. OPACK n'impose aucune contrainte sur l'exécution des actions, sauf que chaque action terminée est marquée par l'environnement, pour permettre à l'agent d'en tenir compte dans son cycle procédural. Le succès ou l'échec d'une action est constaté par l'agent à travers les percepts (par exemple, j'essaye d'ouvrir la porte et je me rend compte qu'elle ne s'ouvre pas alors que l'action est terminée). L'agent peut choisir de supprimer de **A** les actions terminées mais l'environnement ne modifie jamais **A** de lui-même.

L'exécution d'un flux consiste à parcourir l'ensemble d'opérations dans l'ordre topologique du graphe orienté acyclique, défini par les deux types d'arcs, en commençant par les sommets sans prédécesseurs. Nous utilisons l'algorithme implémenté par Huang [13] dans la bibliothèque TASKFLOW³ qui utilise une approche multi-threadée pour effectuer en parallèle les opérations qui le peuvent. Nous avons ajouté à cette bibliothèque le transfert de données d'une opération à ses successeurs.

3.6 Intégration de modèles cognitifs

Un modèle cognitif peut impacter le modèle d'agent à deux niveaux :

1. sur l'état de l'agent (**P**, **A**, **C** et/ou **K**) en ajoutant des variables.
2. sur son ensemble d'opérations **O** en ajoutant des opérations ou en altérant le fonctionnement d'une opération.

Par exemple, si on ajoute un modèle cognitif du stress, il faut une opération pour gérer la nouvelle variable dans **C** et il faut altérer l'opération de sélection d'action pour tenir compte de l'état de stress lors du choix d'une action.

3. <https://taskflow.github.io/>

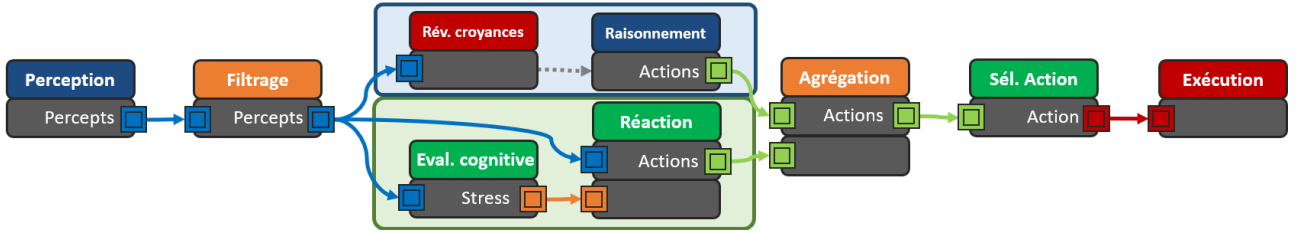


FIGURE 3 – Exemple d'un flux d'opérations arbitraire illustratif.

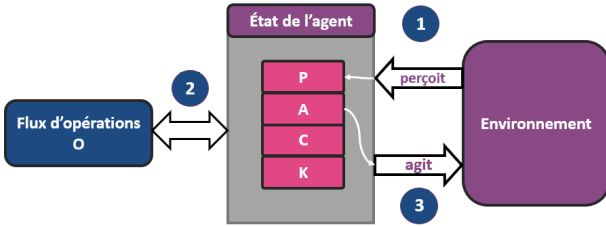


FIGURE 4 – Schématisation du cycle d'un agent.

Une fois que le modélisateur a ajouté les bonnes variables dans l'état de l'agent et, éventuellement, de nouvelles opérations dans **O**, il doit définir la manière dont le modèle cognitif altère les opérations existantes. Cela se fait au travers d'un ensemble de *fonction-impacts* dans notre modèle OPACK. Une *fonction-impact* est une fonction dont les entrées sont identiques à celles de l'opération qu'elle impacte. Elle peut aussi modifier les éléments de **C**, **K** ou **A** par effet de bord et, sauf dans le cas des opérations de rôle "Storage", elle calcule une sortie qui peut être de nature différente de celle de l'opération. Il faut ensuite combiner les sorties de ces *fonction-impacts* : pour cela, le modélisateur définit une *fonction-stratégie* qui combine les sorties des fonction-impacts pour calculer la sortie de l'opération.

Exemple Considérons le modèle cognitif de Driskell [8] qui indique, parmi d'autre, qu'une personne stressée est plus distraite, communique moins et réalise plutôt des actions maîtrisées. Intégrer ce modèle consiste à définir des fonction-impacts sur deux opérations : « *exécution* » (pour tenir compte d'un risque d'échec en cas de stress) et « *sélection d'action* ». Pour la sélection d'action, la *fonction-impact* pourrait, par exemple, renvoyer la liste des actions avec pour chacune d'elle un score calculé en fonction du stress défini dans l'état de l'agent. Ces scores peuvent être utilisés pour sélectionner l'action de l'agent.

Formalisation Soit une opération o et mc un modèle cognitif. On note I_o^{mc} l'ensemble des fonction-impact de mc sur o (éventuellement

vide). On définit $I_o = \cup_{mc} I_o^{mc}$ l'ensemble des fonction-impacts pour l'opération o . Toutes les fonction-impacts $f \in I_o$ ont les mêmes entrées que l'opération o et le même type de sortie, noté $\tau(I_o)$, qui peut être différent de type de sortie de l'opération o elle-même.

On définit une et une seule *fonction-stratégie* s_o par opération comme une fonction qui prend en entrée un ensemble, éventuellement vide, de valeurs de type $\tau(I_o)$ et qui calcule la sortie de o .

La contrainte que nous imposons dans notre méta-modèle est donc de définir une fonction-stratégie par opération et que celle-ci puisse travailler sur un ensemble quelconque, éventuellement vide, d'éléments de type $\tau(I_o)$. **C'est cette fonction-stratégie qui calcule le résultat de l'opération**, à partir des entrées de l'opération et des résultats des fonction-impacts. C'est le cœur de notre méta-modèle.

Notons qu'il est possible de définir plusieurs fonction-impact pour un mc donné et pour une même opération o . Par exemple, pour intégrer le modèle cognitif de Demary [6], le modélisateur peut définir une fonction-impact pour chacun des quatre quadrants du tableau 1.

Impact conditionnel Une fonction-impact définit *comment* un modèle cognitif altère une opération (et le comportement de l'agent). Mais il faut aussi définir *quand* cet impact a lieu en fonction du profil de l'agent et de son état (qui capture la situation). C'est ce que Faur [11] souligne dans son modèle PERSEED. Par exemple, un agent introverti se comporte de manière extravertie en présence d'amis, mais de manière introvertie en présence de collègues de travail. Il s'agit donc de déclencher des fonction-impacts différentes en fonction de la situation.

Nous définissons une fonction d'activation *act* qui prend en entrée l'état de l'agent et une fonction-impact f et qui retourne *true* ou *false*. Pour chaque opération o , lorsqu'elle est évaluée

dans le flux, nous calculons l'ensemble d'impacts $F_o = \{f \in I_o \text{ tq } act(f, PACK) = true\}$ des fonction-impacts activées en fonction de l'état de l'agent et nous en déduisons l'ensemble des résultats de ces fonction-impacts qui sera passé à la fonction-stratégie. Cet ensemble peut évidemment être vide.

3.7 Exemple d'opérationnalisation

Considérons le flux de la figure 2 composé de deux opérations (actions envisageables et sélection d'action) qui correspond à une partie du modèle d'agent REPLICANTS. Nous allons montrer comment intégrer dans ce modèle les deux modèles cognitifs de Driskell et de Demary.

Nous définissons tout d'abord la fonction-stratégie pour ces deux opérations, ce qui nécessite de définir 1) le type $\tau(I_o)$ des fonction-impacts de l'opération et 2) le résultat obtenu par défaut lorsque l'ensemble d'impacts F_o est vide.

- Pour « actions envisageables », nous choisissons que $\tau(I_o)$ est, comme pour o , un ensemble d'actions. Par défaut, la stratégie renvoie le résultat du parcours de l'arbre défini dans le modèle REPLICANTS, stocké dans **K**, c'est-à-dire un ensemble d'actions comprenant a minima l'action « ne rien faire ».
- Pour « sélection d'action », nous choisissons que $\tau(I_o)$ est un ensemble de couples (*action*, *score*). Par défaut, lorsque cet ensemble est vide, le système choisit une action aléatoirement dans la liste des actions fournies en entrée de l'opération.⁴

Nous souhaitons désormais impacter le comportement de l'agent par le biais de modèles cognitifs.

Ajout d'un impact : modèle cognitif de followership Commençons par un comportement simplifié de Demary [6] : un agent *passif* suit les ordres de son supérieur.

Nous supposons qu'à partir de **C**, nous pouvons déterminer si un agent est *passif* ou non. Nous supposons qu'à partir de **K**, nous pouvons déterminer si une action est *ordonnée* ou non par un supérieur.

Nous ajoutons une fonction-impact dans l'opération « sélection d'action ». Elle est active uniquement si l'agent est *passif*. Elle donne, pour

4. Il s'agit ici d'une vision simplifiée par rapport au modèle REPLICANTS. Celui-ci calcule en réalité un ordre total sur les actions en fonction des préconditions des actions.

chaque action, un score de 1 aux actions ordonnées par un supérieur et de -1 si elle a été interdite par un supérieur et de 0 sinon.

De même, nous pouvons ajouter des fonction-impacts pour chacun des 4 quadrants du modèle de Demary.

Ajout d'une opération Supposons désormais que nous souhaitons opérationnaliser un modèle de stress, comme celui de Lazarus [17], utilisé dans le modèle cognitif de Driskell. Ce modèle est basé sur un processus d'évaluation cognitive (*appraisal* en anglais) qui travaille sur un ensemble de variables d'évaluation dans l'environnement pour calculer une état émotionnel.

Nous ajoutons donc une variable de stress dans l'ensemble **C** de l'état de l'agent et une opération « Manipulation » qui opère sur les percepts de l'agent (qui correspondent aux variables d'évaluation cognitives) pour calculer la variable de stress. Il faut aussi ajouter un lien de précedence entre cette nouvelle opération et la sélection d'action.

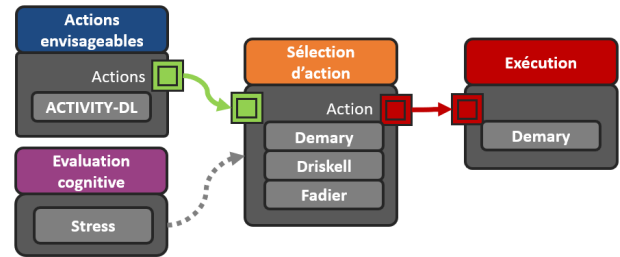


FIGURE 5 – Nouveau flux suite à l'intégration des modèles cognitifs

Ajout de plusieurs impacts : modèle cognitif de Driskell Pour refléter les impacts du stress sur les différentes opérations en nous appuyant sur les travaux de Driskell [8], nous définissons deux fonction-impacts, dépendantes du stress, au sein de l'opération « sélection d'action ». Elles attribuent chacune un score aux actions, telle que : « les actions coopératives ou communicatives ont un score -1 » et « les actions maîtrisées ont un score 1. » La fonction d'activation retourne vrai pour ces deux fonction-impacts uniquement lorsque l'agent est stressé.

Cet exemple simple montre comment il est possible d'intégrer plusieurs modèles cognitifs au sein d'un modèle d'agent dans le méta-modèle OPACK. Nous allons maintenant évaluer l'effet de ces modèles cognitifs sur la génération de comportement et leur intelligibilité.

4 Première évaluation

Pour évaluer la sensibilité des modèles cognitifs sur le comportement de l'agent, nous nous intéressons à un ensemble de mesures définies sur le comportement. Nous allons vérifier que chaque modèle cognitif modifie bien ces mesures comme il le devrait. Par exemple, si nous ajoutons un modèle cognitif qui réduit la communication d'une personne stressée, alors les mesures en lien avec les actes de communication doivent évoluer dans ce sens. Au contraire, les autres mesures ne devraient pas être impactées. Évidemment, il y aura des impacts relatifs puisqu'il s'agit de choisir des actions, donc de faire évoluer le comportement de l'agent.

Nous nous limitons dans cette première étude à l'opération de sélection d'action et nous allons évaluer comment les modèles cognitifs impactent cette sélection. Nous utilisons trois modèles cognitifs : Driskell, Demary et Fadier.

4.1 Protocole

Dans notre scénario, les agents représentant des secouristes doivent choisir parmi 8 actions qui ont les attributs suivants : *ordre* (est-ce que l'action est un ordre transmis par un supérieur), *com* (est-ce que l'action est une action de communication), *coop* (est-ce que l'action se fait à plusieurs ou aide les objectifs de l'équipe), *alu* (est-ce que l'action est tolérée par l'usage, voir section 2.2) et le niveau de qualification requis pour effectuer l'action (SC1, SC2 ou SC3). Nous avons 12,5% d'actions *ordre*, 37,5% d'actions *com*, 50% d'actions *coop*, 12,5% d'actions *ALU*. Nous avons 1 action requérant la qualification la plus haute (SC3) et 3 actions SC2 (les 4 restantes sont SC1).

Chaque agent va donc faire $2^8 - 1 = 255$ fois le cycle pour tirer l'ensemble des combinaisons possibles d'actions. En fonction de son état et des modèles cognitifs actifs ou non, il choisit une action (opération « sélection d'action »). Nous notons à chaque fois l'action choisie et les actions candidates.

Nous avons 24 états d'agent possibles (communiquant ou non, pro-actif ou passif, stressé ou non, qualification de l'agent). Pour chaque modèle cognitif, nous pouvons distinguer les états dans lesquels la fonction d'activation renverrait *vrai* pour les fonction-impacts du modèle cognitif considéré. Par exemple, la fonction d'activation renvoie *vrai* pour les fonction-impacts du modèle

de Driskell lorsque la variable *stress* est vraie. Nous notons De , Dr et ALU l'ensemble de ces états pour les modèles de Demary, Driskell et Fadier respectivement [6, 8, 10]. Réciproquement, nous notons $\overline{De}$, $\overline{Dr}$ et $\overline{ALU}$ l'ensemble des états qui n'activent pas les fonction-impacts des modèles de Demary, Driskell et Fadier respectivement. Concrètement, le groupe d'agents De est le groupe des agents pour lesquels le comportement devrait être impacté quand le modèle de Demary est présent.

4.2 Mesures

Nous considérons les mesures suivantes et nous indiquons entre parenthèse le modèle cognitif qui devrait impacter la mesure :

ORD = pourcentage d'actions « ordre », (De),
COM = pourcentage d'actions « com » (Dr),
ALU = pourcentage d'actions « alu » (ALU),
COOP = pourcentage d'actions « coop » (Dr),
QUAL = pourcentage d'actions choisies dont le niveau de qualification est inférieur ou égal à celui de l'agent (Dr).

Concrètement, pour une combinaison de modèle cognitifs donnée, $ORD(Dr) = 60\%$ signifie que dans l'ensemble des exécutions où une action « ordre » était proposée dans la liste des actions candidates et où la fonction d'activation aurait renvoyé *vrai* suivant les conditions du modèle de Driskell, l'agent a choisi une action « ordre » dans 60% des cas.

4.3 Résultats

Lorsqu'aucun modèle cognitif n'est présent, quelque soit la mesure, il n'y a pas de différences entre les groupes (puisque aucune fonction-impact n'est activée), comme la figure 6 le reporte. C'est le comportement de référence.

Lorsque Demary est le seul modèle cognitif activé et qu'une action ordonnée est proposée, le groupe De choisit toujours cette action ($ORD(De) = 1$) alors que le groupe $\overline{De}$ la choisira une fois sur 4 ($ORD(\overline{De}) = 0.2553$), comme dans le comportement de référence. Puisque, l'unique action ordonnée dans notre jeu de test est une action non-communicative, respectant la réglementation (non- ALU) et coopérative, nous observons une diminution de la mesure COM et ALU et une augmentation de la

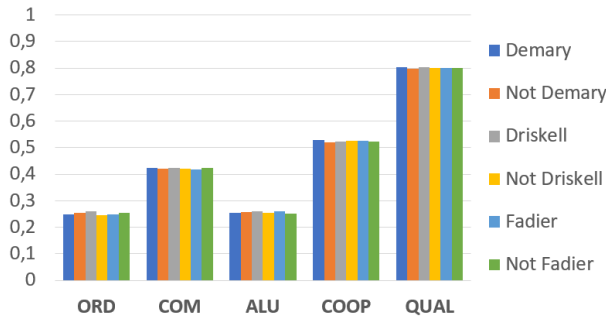


FIGURE 6 – Taux de sélection d'action lorsque aucun modèle n'est activé.

mesure *COOP* pour le groupe actif. Les autres groupes ne sont pas impactés et restent sur le comportement de référence. Ces résultats sont présentés sur la figure 7).

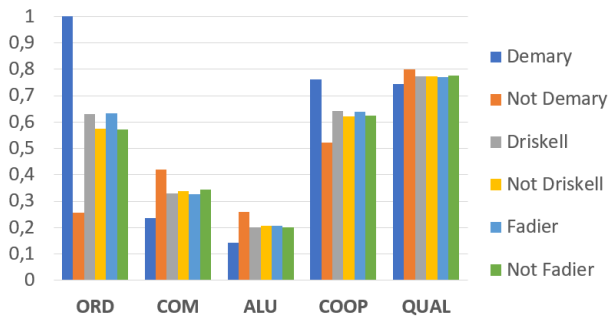


FIGURE 7 – Taux de sélection d'action lorsque seul le modèle de Demary est activé.

Nous obtenons le même résultat pour les autres modèles cognitifs lorsqu'ils sont activés seuls. Lorsque nous activons les trois modèles cognitifs, la tendance de chaque modèle cognitif se conserve (cf. figure 8). Par exemple, $ORD(De) = 0.51 > ORD(\overline{De}) = 0,11$, c'est-à-dire que le groupe actif de Demary continue à privilégier les ordres, contrairement au groupe non-actif. C'est le cas pour tous les modèles cognitifs. Par contre, nous observons des impacts sur les groupes a priori non-concernés par un modèle, en raison du fait que les actions peuvent correspondre à plusieurs critères.

5 Conclusions & Perspectives

Nous avons présenté les fondations d'un méta-modèle pour intégrer dans un modèle d'agent un ensemble de modèles cognitifs. Notre méta-modèle fait un minimum d'hypothèses sur le modèle d'agent et sur les modèles cognitifs considérés. Nous avons détaillé comment, grâce au flux d'opérations, le modélisateur peut définir

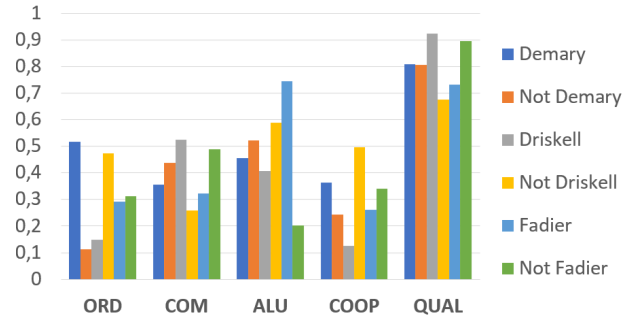


FIGURE 8 – Taux de sélection d'action lorsque tous les modèles sont activés.

et relier les opérations entre-elles pour modéliser le fonctionnement du modèle d'agent. Ensuite, nous avons décrit l'intégration de modèles cognitifs pour impacter cet ensemble d'opérations, grâce à la notion de fonction-impacts et de fonction-stratégies, qui permet de construire des combinaisons de modèles cognitifs non connus au préalable. Enfin, nous avons réalisé une première évaluation de cette approche pour vérifier que les comportements obtenus sont correctement impactés par les modèles cognitifs.

Nous avons montré que les modèles cognitifs ont un impact net sur la sélection d'action, qui est un élément important de la génération de comportements. Pour étendre notre évaluation et qualifier l'intelligibilité du modèle d'agent construit sur notre méta-modèle OPACK, il faudrait simuler le modèle d'agent complet et analyser les séquences d'actions produites. Nous pourrions alors demander à des experts du domaine de qualifier les comportements cognitifs observés (est-ce que l'agent était stressé, est-ce qu'il était communiquant), pour répondre à la question : « le changement de comportement produit par l'intégration ou non d'un modèle cognitif est intelligible » ?

Remerciements

Ces travaux sont financés par le projet DGA RAPID ORCHESTRAA.

Références

- [1] P. G. Balaji and D. Srinivasan. An Introduction to Multi-Agent Systems. In Dipti Srinivasan and Lakhmi C. Jain, editors, *Innovations in Multi-Agent Systems and Applications - 1*, Studies in Computational Intelligence, pages 1–27. Springer, Berlin, Heidelberg, 2010.

- [2] Camille Barot. *Scénarisation d'environnements virtuels : vers un équilibre entre contrôle, cohérence et adaptabilité*. These de doctorat, Compiègne, February 2014.
- [3] Mathieu Bourgeois. *Vers des agents cognitifs, affectifs et sociaux dans la simulation*. phdthesis, Normandie Université, November 2018.
- [4] Michael Bratman. *Intention, Plans, and Practical Reason*. Center for the Study of Language and Information, 1987.
- [5] Paul T. Costa Jr. and Robert R. McCrae. The Revised NEO Personality Inventory (NEO-PI-R). In *The SAGE handbook of personality theory and assessment, Vol 2 : Personality measurement and testing*, pages 179–198. Sage Publications, Inc, Thousand Oaks, CA, US, 2008.
- [6] Guillaume Demary. Évaluation cognitive du leader dans une dyade hiérarchique : des comportements non verbaux du suiveur aux comportements de leadership. *Bulletin de psychologie*, Numéro 569(5) :271–275, November 2020.
- [7] Tripp Driskell, James E. Driskell, and Eduardo Salas. Mitigating Stress Effects on Team Cohesion. In *Team Cohesion : Advances in Psychological Theory, Methods and Practice*, volume 17 of *Research on Managing Groups and Teams*, pages 247–270. Emerald Group Publishing Limited, January 2015.
- [8] Tripp Driskell, Eduardo Salas, and James E. Driskell. Teams in extreme environments : Alterations in team development and teamwork. *Human Resource Management Review*, 28(4) :434–449, December 2018.
- [9] Kutluhan Erol, James Hendler, and Dana Nau. HTN Planning : Complexity and Expressivity. *Proceedings of the National Conference on Artificial Intelligence*, 2, May 1994.
- [10] Elie Fadier, Cecilia De La Garza, and Armelle Didelot. Safe design and human activity : construction of a theoretical framework from an analysis of a printing sector. *Safety Science*, 41(9) :759–789, November 2003.
- [11] Caroline Faur. *Approche computationnelle du regulatory focus pour des agents interactifs : un pas vers une personnalité artificielle*. These de doctorat, Université Paris-Saclay (ComUE), October 2016.
- [12] Jonathan Gratch and Stacy Marsella. A domain-independent framework for modeling emotion. *Cognitive Systems Research*, 5(4) :269–306, December 2004.
- [13] Tsung-Wei Huang, Dian-Lun Lin, Chun-Xun Lin, and Yibo Lin. Taskflow : A Lightweight Parallel and Heterogeneous Task Graph Computing System. *IEEE Transactions on Parallel and Distributed Systems*, 33(6) :1303–1320, June 2022.
- [14] Lauriane Huguet, Domitile Lourdeaux, and Nicolas Sabouret. Présentation du projet VICTEAMS. In *Workshop Affect Companion Artificiel Interaction (WACAI 2016)*, Brest, France, June 2016.
- [15] Lauriane Huguet, Domitile Lourdeaux, and Nicolas Sabouret. Moteur de sélection de tâches pour des personnages virtuels autonomes non omniscients. In *Workshop affects, compagnons artificiels et interaction (WACAI 2018)*, Porquerolles, France, June 2018.
- [16] John E Laird, Allen Newell, and Paul S Rosenbloom. SOAR : An architecture for general intelligence. Technical report, Stanford, 1986.
- [17] Richard S. Lazarus and Susan Folkman. *Stress, Appraisal, and Coping*. Springer Publishing Company, March 1984.
- [18] J. Paul Morrison. Flow-based programming. In *Proc. 1st International Workshop on Software Engineering for Parallel and Distributed Systems*, pages 25–29, 1994.
- [19] M. Ochs, N. Sabouret, and V. Corruble. Simulation of the Dynamics of Nonplayer Characters' Emotions and Social Relations in Games. *IEEE Transactions on Computational Intelligence and AI in Games*, 1(4) :281–297, December 2009.
- [20] Andrew Ortony, Gerald L. Clore, and Allan Collins. *The Cognitive Structure of Emotions*. Cambridge University Press, Cambridge, 1988.
- [21] Barry G. Silverman, David Pietrocola, Ben Nye, Nathan Weyer, Oleg Osin, Dan Johnson, and Ransom Weaver. Rich socio-cognitive agents for immersive training environments : case of NonKin Village. *Autonomous Agents and Multi-Agent Systems*, 24(2) :312–343, March 2012.
- [22] Craig A. Smith and Richard S. Lazarus. Emotion and adaptation. In *Handbook of personality : Theory and research*, pages 609–637. The Guilford Press, New York, NY, US, 1990.