Évaluation et calibration des comportements des agents pour les simulations immersives

Kévin Darty [1], **Julien Saunier** [2], **Nicolas Sabouret** [3]

1. *LIMSI-CNRS, UPR 3251, Univ. Paris-Sud, Orsay, France*
   Kevin.Darty@limsi.fr
2. *LITIS, INSA de Rouen, France*
   Julien.Saunier@insa-rouen.fr
3. *LIMSI-CNRS, UPR 3251, Univ. Paris-Sud, Orsay, France*
   Nicolas.Sabouret@limsi.fr

RÉSUMÉ. *L'un des principaux problèmes en simulation multi-agent est l'évaluation des comportements produits par les modèles d'agents, la définition de leurs paramètres et leur calibration. Ce problème est accentué lorsqu'on considère des environnements virtuels immersifs, dans lesquels les agents intelligents doivent reproduire des comportements humains et apparaître « réalistes » aux yeux des utilisateurs. Nous proposons d'enregistrer et d'analyser les comportements des agents pour évaluer leur similarité avec ceux des humains dans un environnement virtuel immersif. Nous proposons une méthode de classification des comportements humains et d'agrégation des comportements agents sur les classes d'humains pour construire une abstraction des comportements individuels. Les classes obtenues sont étudiées de façon à déterminer les manques, capacités et erreurs dans le modèle agent. Cette méthode nous permet 1) d'écarter les jeux de paramètres invalides, 2) de calibrer des simulations valides et 3) d'expliquer les manques du modèle agent pour l'améliorer.*

ABSTRACT. *In the context of agent-based simulation, a major issue is to define relevant parameters of the agent model and calibrate them. This issue is yet harder in immersive virtual environments, where intelligent agents reproduce human behaviour and interact with users. We propose to log and analyse agents behaviour to evaluate their similarity to humans behaviour in an immersive virtual environment. The behaviour archetypes obtained by clustering are studied in order to identify agent lacks, capacities and errors. This study enables to 1) dismiss invalid parameter sets, 2) calibrate valid simulations and 3) explain lacks in the agent models.*

MOTS-CLÉS : *simulation multi-agent, évaluation, calibration de paramètres, classification.*

KEYWORDS: *multi-agent simulation, evaluation, parameters calibration, clustering.*

DOI:10.3166/RIA.30.237-260

1. Introduction

Dans le contexte des simulations à base d'agents (voir par exemple (Mathieu, Brandouy, 2010 ; Bosse *et al*., 2011 ; Doniec *et al*., 2008)), les agents doivent souvent reproduire des comportements similaires à ceux qu'adopteraient des humains dans la même situation. Une difficulté de ces modèles de simulation est d'identifier les jeux de paramètres qui permettent d'obtenir des comportements « valides ». D'une part, le comportement des agents doit être crédible, *i.e.* les valeurs des paramètres doivent conduire à des comportements qu'un humain pourrait adopter. De l'autre, l'ensemble des comportements produits par les agents doit être représentatif de la population simulée, *i.e.* les jeux de paramètres doivent permettre d'obtenir une certaine variabilité dans les comportements des agents.

La majorité des travaux de recherche dans ce domaine s'intéresse surtout à la validation du comportement au niveau individuel. Cela conduit à définir des jeux de paramètres qui correspondent à des comportements moyens ou normatifs. Si ce choix est pertinent pour produire des simulations « crédibles » au niveau macroscopique, les *SMA* s'intéressent parfois aussi à l'étude de phénomènes microscopiques, pour lesquels les comportements normatifs ne sont pas adaptés (Lacroix *et al*., 2012 ; Bosse *et al*., 2011) car ils ne permettent pas de produire la variabilité de comportement espérée (Huraux, 2015). Dans ce contexte, plusieurs méthodes ont été proposées pour l'analyse semi-automatique des comportements d'ensembles d'agents en fonction des paramètres du modèle. Par exemple, Taillandier *et al.* (Taillandier, Drogoul, 2010) proposent une méthode pour évaluer les ensembles caractéristiques d'une méthode d'apprentissage supervisé qui contrôle un *SMA* géographique. Caillou *et al.* (Caillou, Gil-Quijano, 2012) proposent un modèle de classification des agents suivant leur comportement pour étudier l'impact des paramètres sur la dynamique du *SMA*.

Nous avons proposé dans (Darty *et al*., 2014a) et (Darty *et al*., 2014b) une méthode semi-automatique d'analyse des comportements des agents dans une simulation immersive en comparant des classifications de comportements produits par des agents et par des humains mis dans la même situation. La crédibilité du comportement des agents est alors étudiée en termes de capacités à reproduire des comportements humains, de manques (*i.e.* de comportements humains non reproduits par des agents) et d'erreurs (*i.e.* de comportements produits par des agents mais par aucun humain). Toutefois, aucun de ces modèles ne propose de méthode pour calibrer les paramètres de chaque agent de la simulation à partir de l'analyse des comportements.

Dans cet article, nous proposons d'étendre notre approche proposée dans (Darty *et al*., 2014b) pour l'évaluation et la calibration de simulations multi-agents. Pour cela, nous analysons les comportements produits par les agents indépendamment du modèle sous-jacent. Nous considérons donc les agents comme des boîtes noires et collectons des données sur leur comportement en fonction des paramètres du modèle. Nous comparons alors les traces de simulation à l'aide de mesures calibrées sur les comportements produits par les humains dans la même situation et recueillis à l'aide de simulations participatives (Guyot, Drogoul, 2005). L'agrégation des traces de compor-

tement à l'aide de méthodes de classification permet d'obtenir des archétypes de comportement (Darty *et al.*, 2014a). La composition de chaque classe (agents, humains ou mixte) permet d'identifier et d'expliciter les comportements agents en fonction des paramètres. Enfin, les classes nous permettent de définir la population d'agents pour la simulation.

La prochaine section présente les travaux connexes dans le domaine de l'évaluation des comportements. La section 3 présente notre méthode d'évaluation qui est une extension de nos travaux précédents (Darty *et al.*, 2014b) et décrit les algorithmes utilisés pour la classification et la comparaison des agents et des humains. La section 4 présente notre méthode de calibration d'une simulation multi-agent au sein d'un cycle d'amélioration. La section 5 montre la mise en œuvre de cette méthode dans le contexte de la simulation de conduite.

2. Travaux connexes

La notion de comportement couvre différents aspects, des actions bas niveau sélectionnées par un agent lors d'un cycle d'exécution, à des éléments plus complexes comme les mouvements de foule (Bosse *et al.*, 2011) ou les décisions macroéconomiques (Mathieu, Brandouy, 2010). Cependant, tous ces domaines partagent un même objectif : produire des résultats valides à l'aide de simulations multi-agents pour l'analyse et la prédiction du comportement humain.

La validité est vue dans la majorité des modèles à un niveau macroscopique : les méthodes d'analyse statistique de données sont alors bien adaptées pour déterminer la validité de la simulation (Drogoul *et al.*, 1995 ; Bosse *et al.*, 2011). Il s'agit de vérifier, à travers des données quantitatives, que les agents se comportent de manière similaire à ce qui est observé dans une situation « réelle ». Cependant, le fait que le comportement global soit valide ne garantit pas que les comportements individuels soient réalistes. C'est pourquoi, dans nos travaux, nous nous intéressons au réalisme du comportement au niveau microscopique : chaque agent devrait adopter un comportement ressemblant à celui d'un humain, tout en maintenant la cohérence au niveau macroscopique.

La comparaison de traces au niveau microscopique se heurte à un problème important (Caillou, Gil-Quijano, 2012) : les données recueillies ne peuvent être analysées directement puisque celles-ci sont souvent bruitées et de nature temporelle, alors que les comportements recherchés sont de plus haut niveau. Ceci implique un recours à des traitements de données de façon à donner un sens aux traces de bas niveau.

Dans le domaine des agents virtuels, plusieurs travaux se sont intéressés à la crédibilité des agents et à leur ressemblance aux humains du point de vue de leur réaction affective (Gratch, Marsella, 2005 ; Campano *et al.*, 2013), du comportement non verbal (Pelachaud, 2009), ou de la décision (Bosse *et al.*, 2011). Ces méthodes s'appuient sur l'évaluation de la crédibilité du comportement des agents par un observateur externe (Lester *et al.*, 1997). Cette démarche est coûteuse en temps. C'est pourquoi, si elle est bien adaptée pour étudier des agents virtuels, elle ne peut pas être utilisée

pour traiter des grands nombres d'agents, comme ceux que nous rencontrons dans les simulations multi-agents. En pratique, il est impossible de traiter et tester tous les paramètres sur des centaines d'agents via des méthodes qui nécessitent que le comportement de chaque agent soit observé et analysé par plusieurs humains.

Il existe peu de travaux qui s'intéressent à l'analyse objective (par opposition aux approches subjectives à base de jugement humain) et automatique pour valider des comportements au niveau microscopique. La plupart d'entre eux (Delaherche *et al.*, 2012 ; Gonçalves, Rossetti, 2013) proposent des approches à base d'apprentissage automatique selon des variables de bas niveau. Une approche originale, proposée par Caillou *et al.* (Caillou, Gil-Quijano, 2012), consiste à définir, avec l'aide d'experts du domaine, des variables de haut niveau qui sont ensuite utilisées pour décrire les comportements analysés via un algorithme de classification automatique. Dans le domaine applicatif que nous avons considéré (la simulation de conduite, voir section 5), les variables de bas niveau sont alors par exemple la vitesse, l'angle des roues, *etc* alors que les variables de haut niveau sont le temps avant collision avec le véhicule précédent, le nombre de changements de voie...

La principale limite de ces approches basées sur des données objectives est que, si elles permettent d'identifier les différences entre des catégories de comportements extraites à partir des traces de simulation (nous parlerons de *classes de traces*), elles ne fournissent aucune information au-delà des variables de bas niveau qui ont été utilisées. En particulier, elles ne permettent pas de donner un sens aux classes obtenues : les comportements détectés restent implicites. Au contraire, les approches subjectives, parce qu'elles s'appuient sur des analyses de haut niveau faites par des humains à travers des questionnaires validés par des experts, permettent d'obtenir des classes de comportement. Le modèle présenté dans (Darty *et al.*, 2014b) propose de combiner une méthode basée sur des traces recueillies en simulation (données « objectives ») et une méthode basée sur des annotations par des humains (donc des données « subjectives ») [1]. Nous présentons son fonctionnement dans la prochaine section.

3. Méthode d'évaluation des comportements d'agents

Nous présentons dans cette section l'approche mixte initiée dans (Darty *et al.*, 2014a) pour évaluer les comportements des agents dans le contexte d'environnements virtuels (*EV*) immersifs. Elle consiste à combiner l'analyse des traces d'interactions (approche objective) et les annotations d'observateurs (approche subjective), à utiliser des méthodes de classification pour identifier des typologies de comportement, et à définir des mesures sur les résultats de ces classifications.

La principale contribution de (Darty *et al.*, 2014b) portait sur la définition d'une méthodologie d'évaluation du comportement des agents en combinant l'aproche ob-

1. Dans la suite de cet article, nous parlerons abusivement de « méthode objective » et « méthode subjective » pour désigner ces deux méthodes, basées sur des évaluations à partir de traces ou à partir de questionnaires.

jective et l'approche subjective. Nous avons amélioré notre méthode initiale à l'aide d'un mécanisme d'agrégation dans (Darty *et al.*, 2014a) et nous avons complété le traitement effectué par l'algorithme de classification dans (Darty *et al.*, 2014c). Nous présentons dans cet article la méthode complète, qui intègre aussi un mécanisme de recalibration des paramètres.

3.1. Approche globale

Notre méthode générale d'évaluation est décrite dans la figure 1. Elle ne nécessite pas la connaissance préalable du modèle d'agent, ni de la plateforme de simulation, ni du simulateur utilisé. Nous considérerons donc le modèle d'agent comme une boîte noire.

La méthode est composée de 5 étapes (indiquées par les flèches en haut dans la figure 1) : *1)* le recueil des données de simulation, *2)* l'annotation des rejeux vidéo, *3)* la classification et l'agrégation automatique des données, *4)* la comparaison des classifications, et *5)* l'analyse et l'explicitation des classes par leur composition.

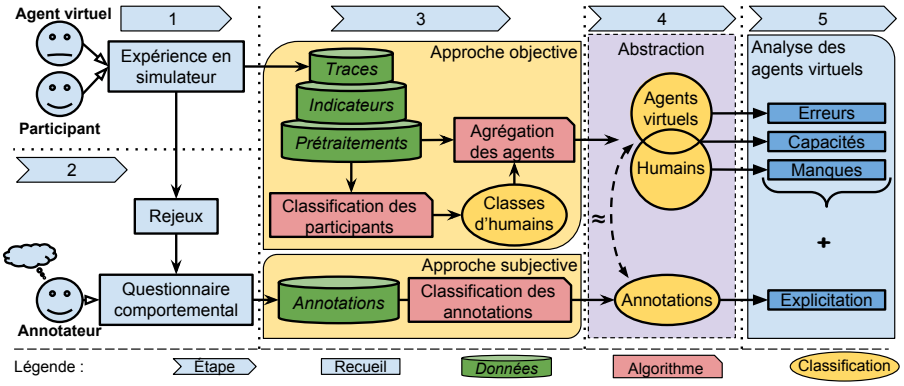

Figure 1. Méthode d'analyse et d'évaluation des comportements d'agents

Les cylindres indiquent les données recueillies puis traitées, les deux rectangles correspondent aux applications d'algorithmes de classifications et les cercles aux classifications obtenues sur les participants et les agents.

La première étape de notre méthode est le recueil de données quantitatives sur les comportements des participants (par exemple, dans le cas de la simulation de conduite, il s'agit des données sur la vitesse, l'écart inter-véhiculaire, *etc*). Nous recueillons des données brutes issues d'une simulation immersive en environnement virtuel. Nous générons aussi des données sur les agents. Pour cela, nous produisons de nouvelles simulations dans lesquelles le participant est remplacé par un agent. Différents types d'agents sont générés en explorant l'espace des paramètres du ou des modèles tels que la normativité, l'expérience, les paramètres décisionnels, *etc*. Nous appelons *acteurs principaux* l'ensemble de ces agents et des participants. Les données brutes des acteurs principaux sont appelées *traces* dans la figure 1.

La seconde étape est l'évaluation subjective des simulations effectuées sur l'ensemble des acteurs principaux. Pour cela, des vidéos sont enregistrées sur l'ensemble des simulations des acteurs principaux. Nous appelons *rejeu* la vidéo faite en vision subjective (c'est-à-dire en vue à la première personne) de la simulation effectuée par un acteur principal. L'utilisation de l'approche subjective consiste à annoter ces rejeux par une population différente de participants (appelés *annotateurs*) via un questionnaire de comportement. Cette étape produit un ensemble d'*annotations*.

L'objectif de la troisième étape est double : premièrement, analyser les comportements des agents en les comparant à ceux des humains, et deuxièmement expliciter les comportements des agents ainsi que les comportements manquants. L'analyse des comportements des agents et de leur capacité à simuler des comportements humains est faite par comparaison des traces de comportements entre humains et agents. Cela ne peut pas se faire sur des données brutes. Ces traces, particulièrement dans le cas des participants, sont bruitées : deux traces différentes peuvent représenter le même type de comportement tactique ou stratégique. C'est pourquoi, dans le but de généraliser l'analyse des traces à un plus haut niveau de comportement, nous proposons d'utiliser des catégories de comportement (appelées abstraction dans la figure 1). Ces catégories servent d'abstraction aux traces en rassemblant dans une même classe différentes traces représentatives d'un même comportement haut niveau. Cela est fait en utilisant une méthode de classification automatique.

Il serait possible d'utiliser un algorithme de classification automatique supervisée en faisant étiqueter par un expert un nombre important de traces. Cependant, cette solution est coûteuse en temps et en ressource. De plus, elle limite fortement le nombre d'agents analysable et contraint l'efficacité d'un cycle de conception itératif en requérant un nouvel étiquetage pour chaque ensemble de paramètres et à chaque modification du modèle. Pour ces raisons, ces algorithmes seront donc non-supervisés. Aussi, n'ayant pas d'information *a priori* sur le nombre de comportements, le nombre de classes doit être libre. C'est ce qui nous a conduits à utiliser les algorithmes de classification et notre méthode d'agrégation présentés dans les sections 3.2 et 3.3 ci-après.

L'explicitation des comportements des agents ainsi que des comportements manquants se fait via l'annotation. Ces annotations permettent une analyse de la typologie des acteurs principaux, et donc une étude des comportements adoptés selon un niveau différent via une approche permettant un autre angle (par exemple, comme nous le verrons dans la section 5, nous pourrons distinguer les conducteurs prudents, les conducteurs aggressifs, *etc*). Dans le même but et de la même manière que pour les traces de simulation, nous utilisons une méthode de classification identique sur le questionnaire d'annotations afin de construire des abstractions, et de les comparer aux abstractions issues des traces. Ce mécanisme est détaillé dans la section 3.4.

Dans le cas où nous analysons pour les deux approches des comportements sous le même angle et de même niveau, nous voulons vérifier que la classification automatique des annotations de comportements observés est liée aux traces de comportements haut niveau dans la tâche donnée. Pour cela, nous évaluons la similarité entre la typologie des participants (par l'annotation) et les classes de comportements observés en

simulation (par les traces). S'il y a une forte similarité entre la composition des classes de comportement et la composition des classes d'annotations, cela veut dire que les classes de comportement ont un sens en termes de typologie du participant.

Dans cette étape, nous analysons la composition de ces classes en termes d'agent et de participant. Cette comparaison conjointe des deux classifications nous permet d'évaluer tout en explicitant les capacités, les manques et les erreurs des agents dans la simulation et d'en extraire des scores et des taux de confiance afin d'évaluer les comportements des agents dans la simulation multi-agent.

3.2. Étape 3a : classification de traces de participant

Le premier objectif de la troisième étape est de comparer les comportements des participants avec ceux des agents de façon à évaluer la capacité des agents à reproduire le comportement des humains. Notre but est d'élaborer des catégories de comportement servant d'abstractions aux traces : chaque classe doit regrouper différentes traces représentatives du même type de comportement haut niveau (voir figure 1).

Pour commencer, les experts du domaines sont consultés afin d'identifier les indicateurs importants du domaine. Les valeurs de ces indicateurs sont calculées à partir des traces de simulation. Ces indicateurs sont alors transformés en scalaires par une série de prétraitements comme décrit dans la partie gauche de la figure 2.

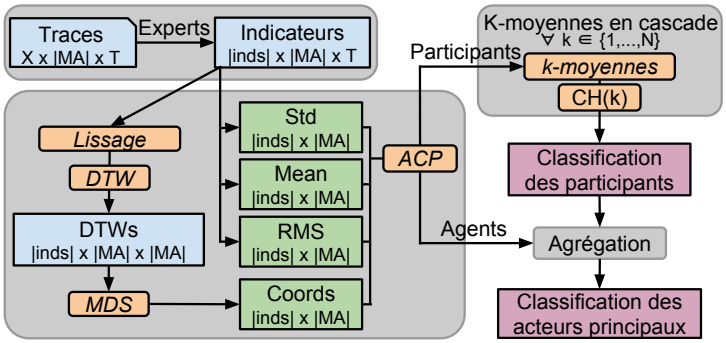

Figure 2. Prétraitements, classification et agrégation des traces selon le temps (T), le nombre de variables (X), d'indicateurs ($inds$), et d'acteurs principaux (MA)

La raison du prétraitement de ces indicateurs est qu'ils sont principalement temporels et qu'un même comportement produit par plusieurs acteurs principaux peut être adopté avec un décalage temporel. Dans le but de prendre cela en compte, nous utilisons l'algorithme Dynamic Time Warping (Salvador, Chan, 2007) (*DTW*) qui calcule les distances mutuelles. L'algorithme des K-moyennes nécessite d'avoir des données dans un espace dimensionnel dont les axes sont perpendiculaires. Par conséquent, afin d'inclure les similarités de *DTW* comme nouvelles variables décrivant les acteurs principaux, nous utilisons un algorithme de positionnement multi-dimensionnel (*MDS*) afin de replacer les acteurs principaux dans un espace dimensionnel. Nous procédons

alors à une Analyse en Composantes Principales (*ACP*) pour projeter les données dans un hyperplan dont les axes sont deux-à-deux décorrélés. Le résultat de ce traitement est un ensemble d'indicateurs non-temporels.

Dans le but d'esquisser les catégories de comportements, nous utilisons un algorithme de classification automatique sur ces indicateurs. Les comportements (spécifiques au domaine et à la situation) étant en nombre indéfini, et dans le but d'éviter l'étiquetage par un expert d'un nombre important de traces, l'algorithme de classification doit être non-supervisé (comme le K-moyennes). De plus, le nombre de comportements étant inconnu, nous utilisons le *Variance Ratio Criterion* (Caliński, Harabasz, 1974) pour déterminer le nombre approprié de classes.

Cette première partie de l'étape 3 a déjà été publiée et plus d'informations peuvent être trouvées dans (Darty *et al.*, 2014c). Dans ce papier, nous ajoutons un composant essentiel à cette méthode : l'agrégation des agents aux classes de participants.

3.3. Étape 3b : agrégation d'agents

Pendant le processus de classification, l'ajout d'acteurs principaux modifie les centroïdes des classes et peut par conséquent changer d'autres affectations. Cependant, les agents et les participants ne doivent pas être considérés de la même manière dans ce processus car les comportements humains représentent la référence à laquelle nous voulons comparer les comportements des agents. Pour cette raison, nous ne souhaitons pas que les indicateurs des agents modifient la classification des comportements humains. Dans le but de garder intactes les classes de participants, l'algorithme des k-moyennes est uniquement appliqué aux participants. Si les agents sont assez proches, ils sont par la suite agrégés à ces classes d'humains fixées. Sinon, ils sont classifiés dans de nouvelles classes d'agents.

Notre méthode fonctionne comme suit. Nous définissons pour chaque classe de participants C_i un seuil t_i au delà duquel l'agent a est considéré comme étant trop éloigné du centroïde m_i pour être agrégé. Ce seuil t_i est défini sur chaque dimension (*i.e.* pour chaque indicateur ind) comme la distance entre le centroïde m_i et le participant de cette classe le plus éloigné p :

$$t_i^{ind} = \forall\ p \in C_i,\ max(|m_i^{ind} - p^{ind}|) \tag{1}$$

Dans le but de permettre l'agrégation des voisins proches, nous élargissons les seuils t_i par un pourcentage de la moyenne des seuils, selon un ratio de tolérance ϵ : cela permet aux classes singletons (pour lesquelles $\forall\ ind,\ t_i^{ind} = 0$) de pouvoir rassembler des agents.

Chaque agent a est agrégé à la classe de participants C_i dont le centroïde est le plus proche parmi celles sous le seuil t_i pour chaque dimension. Si un ou des agents ne se sont pas agrégés aux classes de participants, le premier agent « restant » crée sa propre classe C_{k+1} qui est ajoutée à l'ensemble des classes afin que d'autres agents restants puissent s'y agréger. De même, chaque agent restant essaye de s'agréger à

une classe d'agents restants selon la même règle de seuil que celle utilisée pour les classes de participants ou crée sa propre classe sinon. Ainsi, comme schématisé dans la partie droite de la figure 2, nous obtenons une classification composée de tous les acteurs principaux : participants humains et agents virtuels.

3.4. Étape 3c : classification d'annotations

Le second objectif est d'analyser les comportements via les annotations selon l'approche subjective. Nous utilisons la même méthodologie que celle utilisée pour la classification des traces : l'identification des indicateurs clés, la classification non-supervisée des participants humains selon ces indicateurs et l'agrégation des agents virtuels aux classes de participants.

L'approche subjective demande une annotation manuelle des rejeux des vidéos : quand le nombre d'agents virtuels augmente, il devient irréaliste de les annoter tous. Cependant, sous l'hypothèse que les agents agrégés à une classe de traces seraient annotés de la même façon que les participants de cette classe, il est possible de rendre explicites les comportements des agents par l'annotation des participants. Dans ce cas, la combinaison de l'approche objective et subjective nous permet de comparer n'importe quel nombre d'agents ainsi que n'importe quel nombre de modèle d'agents. Néanmoins, les classes composées uniquement d'agents ne peuvent dès lors plus être explicitées. C'est pourquoi dans l'expérimentation présentée section 5, et considérant que nous n'avons qu'un nombre limité d'agents, nous avons utilisé une annotation manuelle pour l'ensemble des acteurs principaux (participants et agents).

La seconde difficulté pour la classification d'annotations est de choisir le bon ensemble d'indicateurs. Nous voulons que ces indicateurs soit spécifiques à la fois au domaine et à la situation. En général, les questionnaires de comportement permettent de caractériser le comportement général du participant. Cependant, le participant peut adopter un comportement spécifique dans une situation locale qui diffère de son habitude. C'est pourquoi, comme montré dans la section 5, nous avons adapté un questionnaire de comportement spécifique au domaine afin de définir des indicateurs pour l'annotation de la situation. Les scores d'échelles du questionnaire sont calculés en additionnant les questions concernées et normalisés entre 0 et 1.

Nous classifions ensuite les scores des participants en utilisant les mêmes algorithmes des K-moyennes et d'agrégation des agents, ce qui nous permet d'obtenir des classes d'annotations.

3.5. Étape 4 : comparaison de classes

La quatrième étape de notre méthode est la comparaison de ces deux classifications (celle de traces et celle d'annotations). Étant donné qu'elles évaluent toutes deux les comportements, avoir une forte similarité entre elles en termes de composition est une vérification partielle que la classification des traces à un sens en termes de catégories d'utilisateurs spécifiques à la situation, et donc correspond bien à des comportements

haut niveau spécifiques à la tâche. Nous évaluons la similarité entre ces deux classification (voir la flèche pointillée dans la figure 1) par l'Indice de Rand (*RI*) (Rand, 1971).

Dans le cas où le questionnaire d'annotations s'intéresse à des comportements sous un angle différent que celui adopté par les indicateurs issus des traces, les deux classifications apportent des informations complémentaires mais elles ne sont pas nécessairement similaires, leur comparaison n'a donc pas lieu d'être : la valeur de RI obtenue sera faible. C'est le cas, par exemple, lorsqu'un même type de participant humain (défini par les annotations) peut être amené à adopter des comportements différents pour une même situation, conduisant ainsi à des traces dissimilaires. C'est aussi le cas lorsqu'un même comportement (issu des traces de simulation) est adopté par différents types de participants humains (en termes d'annotations). C'est pourquoi il est nécessaire de pouvoir analyser plus en détail la composition des classes, ce que nous faisons dans l'étape 5.

3.6. Étape 5 : analyse des classes de comportement

La cinquième et dernière étape consiste en l'analyse des comportements produits par les agents. Il est possible de distinguer trois types de classe en termes de composition participant-agent :

1. Les classes mixtes (notées C_M) contenant à la fois des acteurs principaux humains et agents correspondent à des comportements haut niveau correctement reproduits par les agents et donc à des capacités du modèle d'agent. Nous notons $\mathbb{C}_M$ l'ensemble de ces classes mixtes.

2. Les classes composées uniquement d'agents (notées C_A) correspondent à des comportements produits uniquement par des agents. Dans la plupart des cas cela reflète des erreurs de simulation, au sens où aucun humain n'a adopté ces comportements dans cette situation. Cela peut aussi être dû à un échantillon trop faible de participants si l'hypothèse de représentativité de la population n'est pas respectée. L'ensemble de ces classes agents est noté $\mathbb{C}_A$.

3. Les classes composées uniquement de participants humains (notées C_H) correspondent à des comportements qui n'ont pas été reproduits par les agents. Elles correspondent à des comportements humains n'ayant pas été reproduits par les agents, et sont donc dus soit à des manques dans le modèle d'agent, soit à un échantillon trop faible d'agents dans l'espace des paramètres. L'ensemble de ces classes humains est noté $\mathbb{C}_H$.

Nous combinons ensuite cette comparaison humain-agent avec l'analyse de l'annotation afin de donner de l'information explicite (c'est-à-dire des caractéristiques haut niveau issues du questionnaire de comportement) sur les comportements des agents ainsi que sur les comportements manquants le cas échéant.

3.6.1. Taux de confiance

Les algorithmes de classifications peuvent produire des regroupements erronés parmi les classes de traces de participants, c'est-à-dire des erreurs d'affectation. En conséquence, une classe singleton peut représenter soit un comportement singulier, soit une erreur de classification. De même, l'algorithme d'agrégation des agents peut - par effet de seuil - agréger un agent à une classe d'humain dissimilaire (*i.e.* qui ne représente pas le même comportement). À l'inverse, une instance peut être exclue d'une classe qui correspond à son comportement. Il est donc nécessaire de fournir des métriques sur la confiance qui peut être attribuée au type d'une classe (erreur, manque, capacité).

Les taux de confiance dépendent du nombre d'acteurs principaux par classe et dans l'ensemble de la classification. Nous définissons donc pour une classe le nombre d'agents, le nombre d'humains $H(C)$, et le nombre total d'acteurs principaux $|C| = A(C) + H(C)$. De la même manière, nous définissons $A(\mathbb{C})$ et $H(\mathbb{C})$ pour une classification $\mathbb{C}$. Aussi, soient $|\mathbb{C}_H|$ le nombre de classes d'humains, $|\mathbb{C}_A|$ le nombre de classes d'agents, $|\mathbb{C}_M|$ le nombre de classes mixtes, et $|\mathbb{C}| = |\mathbb{C}_H| + |\mathbb{C}_A| + |\mathbb{C}_M|$ le nombre total de classes.

Le taux de confiance en une classe dépend de son type. Pour une classe mixte C_M (*i.e.* une capacité), le taux de confiance dépend du ratio d'humains $t_A(C)$ et du ratio d'agents $t_H(C)$ entre la classe étudiée et l'ensemble de la classification :

$$t_A(C_M) = \frac{A(C_M)}{A(\mathbb{C})} \in [0,1] \qquad t_H(C_M) = \frac{H(C_M)}{H(\mathbb{C})} \in [0,1] \tag{2}$$

Pour ces classes mixtes, la confiance est alors faible quand le taux d'humain est « éloigné » du taux d'agents. À l'inverse, nous considérons que la confiance est suffisamment forte quand le taux est « éloigné » de 0 :

$$t_{conf}(C_M) = 1 - |t_A(C) - t_H(C)| \tag{3}$$

Par exemple, une classe contenant 9 participants sur 12 et seulement 2 agents sur 20 a un taux de confiance de $1 - |\frac{9}{12} - \frac{2}{20}| = 0,35$. Conséquemment pour cette classe mixte, la confiance en la capacité de l'agent à reproduire ce comportement humain est faible et cette classe mixte représente potentiellement des erreurs de classification. À l'inverse, une classe contenant 2 participants sur 12 et 4 agents sur 20 a un taux de confiance de $1 - |\frac{2}{12} - \frac{4}{20}| = 0,97$. Il y a donc une forte confiance que cette classe mixte soit réellement une capacité du modèle de l'agent à reproduire le comportement humain adopté, la possibilité d'erreurs de classification est alors écartée.

Pour les classes contenant uniquement des agents, la confiance dépend uniquement du nombre d'agents par rapport au nombre moyen d'agents par classe dans l'ensemble de la classification E_A :

$$E_A(\mathbb{C}) = \frac{A(\mathbb{C})}{|\mathbb{C}_M| + |\mathbb{C}_A|} \qquad t_{conf}(C_A) = \frac{A(C)}{E_A(\mathbb{C})} \tag{4}$$

De même, la confiance en les classes contenant uniquement des humains dépend du nombre moyen d'humains par classe E_H :

$$E_H(\mathbb{C}) = \frac{H(\mathbb{C})}{|\mathbb{C}_M| + |\mathbb{C}_H|} \qquad t_{conf}(C_H) = \frac{H(C)}{E_H(\mathbb{C})} \tag{5}$$

Un taux de confiance élevé en une *classe d'humains* (respectivement une *classe d'agents*) signifie que cette classe peut être considérée comme un manque (respectivement une erreur) dans le modèle d'agent avec confiance.

3.6.2. Scores de type de classe

L'analyse de la composition des différentes classes nous permet de différencier erreurs, capacités, et manques. Il est alors possible de les quantifier. Nous proposons de calculer un score pour chacun de ces trois types (voir les équations 6) : le score de capacités S_c, le score de manques S_m, et le score d'erreurs S_e.

$$S_c = \frac{|\mathbb{C}_M|}{|\mathbb{C}_M| + |\mathbb{C}_H|} \qquad S_m = \frac{|\mathbb{C}_H|}{|\mathbb{C}_M| + |\mathbb{C}_H|} \qquad S_e = \frac{|\mathbb{C}_A|}{|\mathbb{C}_M| + |\mathbb{C}_A|} \tag{6}$$

Le score de capacités permet d'analyser quantitativement la capacité du modèle d'agent dans l'ensemble du *SMA* à reproduire des comportements humains en prenant en compte les manques (*i.e.* les classes d'humains). Par exemple pour une classification composée de 2 classes contenant uniquement des humains et 3 classes contenant à la fois des humains et des agents, le score de capacités sera de $\frac{3}{2+3}$. À l'inverse, le score de manques informe sur les manques du modèle en prenant en compte ces capacités. Le score d'erreurs quant à lui quantifie les comportements erronés (*i.e.* le nombre de classes d'agents) relativement aux classes contenant des agents (c'est-à-dire les classes mixtes et les classes d'agents). Plus les comportements adoptés par les agents sont dissimilaires des comportements humains, plus le score d'erreurs est élevé.

4. Cycle d'amélioration

Dans cette section, nous présentons - au sein d'un cycle d'amélioration - l'utilisation de cette analyse des compositions de classes afin d'évaluer la sur-représentation et la sous-représentation de comportements en se basant sur les scores exposés dans les sections précédentes. Ce cycle d'amélioration est schématisé dans la figure 3.

4.1. Calibration

Dans la simulation multi-agent, la calibration d'un modèle signifie l'ajustement des paramètres de façon à ce que certains buts globaux ou comportements désirés soient atteints (Fehler *et al.*, 2005 ; 2006), voir *e.g.* (Veremme *et al.*, 2012). La calibration des paramètres d'un modèle pour des modèles détaillés à base d'agents est

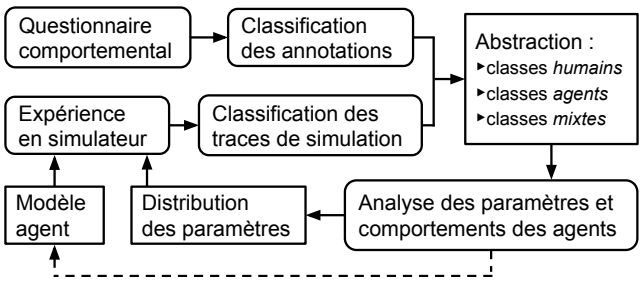

Figure 3. Cycle d'amélioration et détail de la calibration

un problème pour les techniques standards de calibration dû au large espace des paramètres, au long temps d'exécution de la simulation, et des incertitudes dans la conception du modèle. En simulations participatives, où les agents interagissent avec des participants, il faut de plus que les comportements individuels de ces agents soient « crédibles ».

Dans notre cas, nous nous appuyons sur les traces des comportements des participants collectées pendant l'expérimentation. Elles définissent alors un ensemble de comportements valides. En d'autres termes, notre objectif est de réduire la valeur du score d'erreurs S_e et celle du score de manques S_m, et donc d'augmenter la valeur du score de capacité S_c. Considérant le modèle d'agent comme une boîte noire pouvant produire différents comportements en fonction de ses paramètres, le processus de calibration doit s'assurer que :

– le comportement résultant de chaque agent a est crédible, c'est-à-dire que leur ensemble de paramètres $P_a = \{p_1, \ldots, p_i\}$ (avec i le nombre de paramètres du modèle) est individuellement valide ; et que

– la population d'agents reproduit bien les comportements humains d'un point de vue global et dans des proportions équivalentes, c'est-à-dire que la distribution des ensembles de paramètres $\mathcal{P} = \{P_1, \ldots, P_n\}$ avec n le nombre d'agents est valide globalement.

Concrètement, les agents appartenant aux classes d'erreurs ont montré des comportements non exhibés par les participants humains, et sont donc considérés comme non crédibles. En conséquence, leurs ensembles de paramètres doivent être retirés du groupe d'ensembles de paramètres valides. À l'inverse, les manques (*i.e.* les comportements exhibés par des humains mais par aucun agent) peuvent provenir de paramètres mal choisis, sous l'hypothèse que les agents peuvent produire les comportements identifiés.

4.1.1. Typologie plus fine des classes

Dans le but de proposer des proportions correctes de comportements valides, nous prenons en compte la composition des classes à la fois pour les données des agents et pour celles des humains. Nous pouvons alors raffiner les catégories précédemment

présentées dans la section 3.6 :

– les comportements erronés lorsque aucun humain n'appartient à la classe, correspondant aux classes d'agents ($H(C) = 0$),

– les capacités humaines basées sur les classes mixtes $\mathbb{C}_M$, comprenant :

- l'ensemble des comportements sur-représentés ($\mathbb{C}_{sur}$) quand la proportion d'agents est supérieure à la proportion d'humains ($A(C) >> H(C)$),

- l'ensemble des comportements correctement représentés ($\mathbb{C}_{valide}$) quand la proportion d'humains est « proche » de celle des agents ($A(C) \approx H(C)$), et

- l'ensemble des comportements sous-représentés ($\mathbb{C}_{sous}$) quand la proportion d'agents est inférieure à la proportion d'humains ($A(C) << H(C)$).

– les comportements manquants quand aucun agent n'appartient à la classe, correspondant aux classes d'humains ($A(C) = 0$).

Après une première calibration de l'ensemble des paramètres $\mathcal{P}$, les scores ne dépendent plus du nombre d'agents mais de la proportion d'agent dans la population totale. Généralement, la population d'agents est plus grande que la population humaine pour des raisons expérimentales (principalement le temps de recueil des données). De plus, l'opérateur $\approx$ séparant la sous-représentation et la sur-représentation de la représentation valide d'un comportement conduit à définir un seuil. Nous proposons de prendre en compte la taille de chaque classe (*e.g.* $\delta(C) = \frac{5}{100}|C|$).

4.1.2. Scores pour les classes de comportement

Afin d'améliorer la calibration, nous établissons - en plus des deux scores concernant les manques et les erreurs - trois scores pour les classes de comportement mixtes appelés scores de représentativité :

1. Sur-représentation (et son score $S_{sur} = \frac{|\mathbb{C}_{sur}|}{|\mathbb{C}_M|}$) :

$$\mathbb{C}_{sur} = \{C_M \in \mathbb{C}_M, A(C_M) > H(C_M) + \delta|C_M|\} \tag{7}$$

3. Sous-représentation (et son score $S_{sous} = \frac{|\mathbb{C}_{sous}|}{|\mathbb{C}_M|}$) :

$$\mathbb{C}_{sous} = \{C_M \in \mathbb{C}_M, A(C_M) < H(C_M) - \delta|C_M|\} \tag{8}$$

4. Représentation valide (et son score $S_{valide} = \frac{|\mathbb{C}_{valide}|}{|\mathbb{C}_M|}$) :

$$\mathbb{C}_{valide} = \{C_M \in \mathbb{C}_M, H(C_M) - \delta|C_M| < A(C_M) < H(C_M) + \delta|C_M|\} \tag{9}$$

Ces scores nous permettent d'évaluer la qualité de la calibration existante et de possiblement la valider. Dans le cas contraire, ils sont utilisés pour modifier les proportions des agents pour chaque ensemble de paramètres $\mathcal{P}_i$, explorer de nouveaux ensembles de paramètres, et retirer ceux qui sont invalides.

4.2. Ensembles de paramètres

En fonction du modèle d'agent et du nombre de paramètres, il est possible de générer le spectre complet des comportements d'agents possibles. Dans ce cas, un unique cycle via cette méthode est suffisant pour déterminer les paramètres valides et leur proportion dans la population d'agents.

Soit $\mathcal{P}_v$ le groupe des ensembles de paramètres valides correspondant aux comportements valides $\mathcal{B}_v$, avec $simul(P_i) = b \in \mathcal{B}$ l'ensemble des comportements possibles, et $p(b)$ la proportion de participants humains exhibant ce comportement.

Étant donné que plusieurs ensembles de paramètres peuvent produire le même comportement, la production d'un ensemble de paramètres $P(a_i)$ avec $i \in \{1, \ldots, n\}$ pour n agents implique de devoir choisir parmi plusieurs ensembles de paramètres. Nous proposons de sélectionner les ensembles de paramètres de la manière suivante : $P(a_i) = P_i \in \mathcal{P}_v$ avec une probabilité $p(P_i)$ dépendant de la proportion de comportements observés b et du nombre d'ensembles de paramètres P_j amenant au comportement b, d'où :

$$p(P_i) = n\frac{p(b)}{|P_j|} \text{ avec } P_j \in \mathcal{P}_v | simul(P_j) = b \tag{10}$$

De cette manière, les comportements qui étaient sous-représentés ont une probabilité plus forte d'être produits. En effet, la probabilité de sélectionner un ensemble de paramètre compatible avec eux augmente, tandis que l'inverse est vrai pour les comportements sur-représentés.

Il est aussi possible de choisir de manière arbitraire un des ensembles de paramètres $P_j \in \mathcal{P}_v | simul(P_j) = b$ et générer $n \times p(b)$ agents avec cet ensemble de paramètres. Selon la simulation, garder une hétérogénéité contrôlée des agents permet de produire des simulations plus « réalistes ». Cette hétérogénéité est contrôlée puisque tous les ensembles de paramètres (menant aux comportements appartenant à la même classe) produisent des comportements similaires d'après la classification, sinon identiques.

Il est à noter qu'en utilisant uniquement $\mathcal{P}_v$ et non $\mathcal{P}$, tous les ensembles de paramètres invalides détectés par la classification sont retirés.

4.3. Exploration de l'espace des paramètres

Afin de couvrir tous les ensembles de paramètres, le nombre d'agents devant être générés peut être important. De par l'utilisation de la connaissance d'expert ou des valeurs par défaut inclues dans les modèles, il est aussi possible de ne pas générer tous les comportements lors du premier cycle d'application de la méthode d'évaluation de la simulation multi-agent. Dans le cas où des manques dans le modèle d'agent sont trouvés, nous proposons d'inclure une fonction d'exploration décrite dans l'équation

11 qui choisit les ensembles de paramètres dans les zones non-explorées de l'espace des paramètres :

$$P(a_i) = \begin{cases} P_i \in \mathcal{P}_v & \textit{si } p > \gamma \\ P_k \notin \mathcal{P} & \textit{sinon} \end{cases} \quad (11)$$

Le paramètre d'exploration γ permet de chercher de nouveaux comportements de manière itérative, avec p une variable aléatoire uniforme. Afin de ne pas tester plusieurs fois les mêmes paramètres invalides, P_k ne doit jamais être choisi parmi les étapes précédentes $\mathcal{P}$. Dans le cas où P_k mène à un comportement valide, il est alors ajouté à $\mathcal{P}_v$; dans le cas contraire, il est écarté.

Notons que selon le modèle d'agent considéré, il peut exister une méthode spécifique afin d'explorer de manière efficace l'espace des paramètres. À l'opposé, fixer la valeur de γ à 1 revient à faire uniquement de l'exploration. Dans ce cas $\mathcal{P}_v$ se construit itérativement sur des ensembles différents.

4.4. Cycle

Comme vu dans les sections précédentes, si tous les comportements cibles - *i.e.* déterminés par les classes de traces des participants - sont reproduits, une seule étape de calibration est nécessaire. Quand plusieurs comportements sont manquants, explorer l'espace des paramètres peut permettre de découvrir de nouvelles capacités aux agents.

Les manques et les erreurs peuvent aussi être résolus par une intervention du concepteur du modèle d'agent. Dans ce cas, les informations issues de l'étape de l'annotation permettent d'identifier sémantiquement les comportements manquants et les erreurs, *i.e.* expliciter ces comportements.

Rappelons que l'annotation et l'expérimentation sur les participants humains n'est nécessaire qu'une unique fois, et ce indépendamment du nombre de cycles puisque le traitement des données des agents est basé sur l'agrégation sur les traces des participants humains. Ainsi, les données des agents ne modifient pas les comportements de référence formés des classes de traces des participants. Une autre application de cette méthode est donc de comparer plusieurs modèles et plusieurs calibrations de modèles par rapport à cette référence.

5. Évaluation

Cette section illustre notre méthode avec une application au cas de la simulation de conduite. Elle présente ensuite l'analyse de données ainsi que la discussion des résultats.

5.1. Cas d'étude : la simulation de trafic routier

Afin de tester notre méthode, nous proposons d'évaluer le réalisme des comportements des agents du simulateur de trafic routier de l'*IFSTTAR* (dont le dispositif est visible sur la figure 4), au sein du simulateur de conduite *ARCHISIM* dont nous ne détaillerons pas le fonctionnement des agents (les détails du modèle peuvent être trouvés dans (Champion *et al.*, 2001 ; Espié, Auberlet, 2007 ; Six, 2014)). Dans ce simulateur de conduite, le participant pilote un véhicule sur un parcours routier. Lors de la conduite sur ce circuit, le participant est en interaction avec des véhicules simulés générant le trafic. Nous sommes donc dans le cas de la simulation multi-agent immersive pour laquelle notre méthode s'applique.

Figure 4. Dispositif du simulateur de conduite (avec 3 *écrans, volant, boîte de vitesse et pédalier) utilisant le simulateur de trafic routier* ARCHISIM

Dans cette application de notre méthode d'évaluation sur le simulateur de conduite *ARCHISIM*, les données de simulation correspondent aux traces de simulation. Ces traces prennent en compte les informations sur l'environnement de conduite, sur l'acteur principal (c'est-à-dire le conducteur humain ou le conducteur simulé par un agent), et sur les véhicules simulés en interaction.

Dans le cadre de la conduite, les données issues de l'annotation correspondent aux réponses à un questionnaire de comportement de conduite.

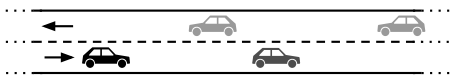

Figure 5. Scénario de l'expérimentation de conduite

Notre méthode est illustrée par l'expérimentation suivante : les participants conduisent une voiture sur une route contenant des véhicules simulés. Le circuit (montré en figure 5) fournit une situation de conduite sur une route à double sens. Cela correspond à environ 1 minute de conduite. L'acteur principal (en noir) rencontre un véhicule à vitesse réduite (en gris clair) sur la voie de droite et plusieurs véhicules en sens inverse sur la voie de gauche dont les distances les séparant vont en augmentant

de telle façon qu'il est presque impossible de doubler prudemment avant le premier véhicule arrivant en face.

5.2. Analyse de données

Les 22 participants de notre expérimentation de simulation de conduite sont des conducteurs réguliers âgés de 24 à 59 ans.

Premièrement, un test est effectué sans trafic simulé afin que le participant s'habitue au fonctionnement du simulateur et au circuit. Ensuite, le participant effectue le scénario, cette fois en interaction avec les véhicules simulés. Les données sont alors sauvegardées pour la phase de prétraitement et une vidéo est enregistrée pour le rejeu. Enfin, une autre population de 6 participants remplit le questionnaire d'annotation après visionnage des rejeux (22 participants et 14 agents). L'annotation consiste en un questionnaire, basé sur le Driver Behavior Questionnaire (DBQ) de (Reason *et al.*, 1990). Il comprend 5 questions correspondant aux cinq sous-échelle du DBQ (contrôle et actions, mémoire et attention, jugement et planification, violation accidentelle, violation délibérée) évaluées sur une échelle de Likert (Likert, 1932) à 7 points, plus une question sur le risque d'accidents notée sur 4 valeurs (aucun risque, risque possible, risque certain, sans opinion). Nous avons aussi ajouté une question relative au contrôle perçu (dans quelle mesure le conducteur observé contrôle son véhicule) notée sur la même échelle à 7 points, et une dernière question portant sur la nature de l'acteur principal (humain ou agent simulé). Nous n'avons pas utilisé ces deux dernières questions dans notre classification.

Les paramètres du modèle *ARCHISIM* sont : $v \in \mathbb{N}$ la vitesse désirée (en km/h), $reglmnt \in [0, 100]$ la volonté de conduire selon le code de la route, $infra \in [0, 100]$ la capacité de contrôle du véhicule selon l'infrastructure, $trafic \in [0, 100]$ la flexibilité du temps inter-véhiculaire selon le trafic, $soupl \in [0, 100]$ l'agressivité du conducteur (ne tenant pas compte des courtes variations), $exp \in \{0, 1\}$ l'expérience du conducteur.

Dans cette itération, les paramètres utilisés sont les valeurs moyennes pour chaque paramètre, tandis qu'un paramètre est modifié pour chaque agent. Le paramètre v est choisi parmi $\{100, 110, 120, 130, 140\}$; les quatre paramètres suivant ($reglemnt$, $infra$, $trafic$, $soupl$) sont par défaut à 50, et à 25 ou 75 sinon ; enfin exp est à 0 ou à 1.

5.3. Résultats de classification

Les classifications de traces et d'annotations sont comparées comme illustré dans la figure 6. Il y a 2 classes de comportement issues des traces : *1)* La classe de traces 1 est composée des acteurs principaux qui n'ont pas essayé de dépasser le véhicule à vitesse réduite. Étant une classe mixte, c'est une capacité du modèle agent à reproduire un comportement humain qui est celui de choisir de ne pas dépasser. *2)* La classe de traces 2 contient des acteurs principaux qui ont dépassé le véhicule à vitesse réduite.

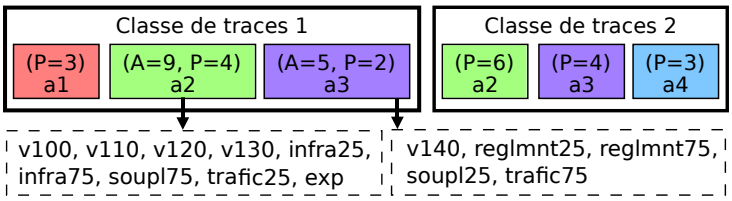

Figure 6. Comparaison des acteurs principaux entre la classification de traces (avec les participants P et les agents A), et la classification d'annotations regroupés par classe avec leur numéro ($a\#$), en fonction de leurs paramètres

Comme cette classe est uniquement composée de participants, c'est donc un manque dans le modèle agent : les agents ne peuvent pas choisir de ne pas dépasser comme les humains le font.

Il y a 4 classes de comportements issues des annotations : *1)* la classe d'annotations 1 a été considérée par les annotateurs comme ayant le style de conduite le plus dangereux, aucun agent n'a été considéré dangereux. *2)* la classe d'annotations 2 a été annotée comme correspondant à des conducteurs prudents. Étant une classe mixte, le comportement normatif humain peut donc être considéré comme partiellement reproduit. *3)* la classe d'annotations 3, mixte également, est une petite classe d'annotations dont les acteurs principaux ont été jugés comme étant des conducteurs ordinaires. *4)* la classe d'annotations 4 contient des acteurs principaux considérés comme légèrement dangereux, comportement non reproduit par les agents.

Dans la classe de traces 1, les indicateurs n'ont pas permis de distinguer la classe d'annotations 1 du reste des acteurs principaux, tandis que ces acteurs principaux ont essayé de dépasser en vain. Pareillement, les participants de la classe d'annotations 4 n'ont pas été séparés de la classe de traces 2 en une nouvelle classe.

5.4. Scores

Les scores correspondant à la classification des annotations dans cette expérimentation sont les suivants :

- le score d'erreur est $S_e = 0$ (car $|\mathbb{C}_A| = 0$),
- le score de manque est : $S_m = \frac{|\mathbb{C}_H|}{|\mathbb{C}_M|+|\mathbb{C}_H|} = \frac{1}{1+1} = \frac{1}{2}$.
- le score de capacité est : $S_c = \frac{|\mathbb{C}_M|}{|\mathbb{C}_M|+|\mathbb{C}_H|} = \frac{1}{1+1} = \frac{1}{2}$.

Le taux de confiance en la classe 1 comme étant une capacité du modèle est $t_{conf}(C_1) = 1 - \left|\frac{14}{14} - \frac{9}{22}\right| \approx 0,41$. Ce taux n'étant pas proche de 0, cette classe peut donc être considérée avec confiance comme étant une capacité du modèle.

La classe de traces 2 est une classe composée uniquement d'humains, le taux de confiance est alors calculé de la manière suivante : $E_H(\mathbb{C}) = \frac{22}{1+1} = 11$, $t_{conf}(C_2) = \frac{13}{11} \approx 1,18$. $t_{conf}(C_2)$ est proche de 1 signifiant que cette classe compo-

sée uniquement d'humains peut être traitée avec confiance comme étant un comportement manquant réel dans le modèle agent.

Ces scores permettent de trouver les nombres de comportements adéquats et inadéquats, et donc de produire l'ensemble des jeux de paramètres qui sont valides : $P \in \mathcal{P}_v = \{v100, v110, v120, v130, infra25, infra75, soupl75, trafic25, exp\} \cup \{v140, reglmnt25, reglmnt75, soupl25, trafic75\}$.

Il n'y a qu'une classe mixte. Cette dernière est sur-représentée, en effet pour un $\delta = \frac{5}{100}$ (*i.e.* peu de tolérance aux variations de proportions entre les comportements humains et agents) : $A(C_1) > H(C_1) + \delta|C_1| \Leftrightarrow 14 > 9 + \frac{5}{100} \times 23$. Les scores de représentativité correspondants sont : $S_{sur} = \frac{|\mathbb{C}_{sur}|}{|\mathbb{C}_M|} = 1$, $S_{sous} = \frac{|\mathbb{C}_{sous}|}{|\mathbb{C}_M|} = 0$, et $S_{valide} = \frac{|\mathbb{C}_{valide}|}{|\mathbb{C}_M|} = 0$.

Les scores de type de classe ont permis de quantifier les erreurs (0), capacités $(0,5)$, et manques $(0,5)$. Les taux de confiance ont permis de s'assurer que la classe mixte est bien une capacité du modèle d'agent ($t_{conf}(C_1) \approx 0,41$) et que la classe d'humains est bien un comportement humain manquant ($t_{conf}(C_2) \approx 1,18$). Les scores de représentativité montrent que la calibration originale sur-représente le comportement humain (correspondant à la classe mixte) dans la population d'agents.

Une fois ces scores calculés, il est possible de calibrer une nouvelle population d'agents.

5.5. Calibration

Nous avons trouvé que seule une partie des comportements humains étaient reproduits. Dans ce cas, il existe deux possibilités pour l'utilisateur de la simulation : soit entrer dans un cycle d'exploration de l'espace des paramètres, soit calibrer le système en utilisant la première analyse. Dans cette partie, nous nous concentrons sur la seconde possibilité.

Nous avons vu que tous les jeux de paramètres étaient valides mais sur-représentés, et qu'il y avait plusieurs comportements manquants. Par conséquent, afin d'obtenir des agents reproduisant le comportement humain, nous calibrons les nouveaux ensembles de paramètres des agents en les choisissant parmi ceux valides dans la première expérimentation et en raffinant les proportions grâce aux classes d'annotations.

La première classe contient 9 traces d'agents et 4 traces de participants, tandis que la seconde classe contient 5 traces d'agents et 2 traces de participants. Nous obtenons donc, en calculant leur probabilité d'apparition dans la prochaine calibration : $p(v100) = p(v110) = p(v120) = p(v130) = p(infra25) = p(infra75) = p(soupl75) = p(trafic25) = p(exp) = \frac{4/6}{9}$ et $p(v140) = p(reglmnt25) = p(reglmnt75) = p(soupl25) = p(trafic75) = \frac{2/6}{5}$.

De cette manière, les agents représentent stochastiquement la densité de comportements humains dans la simulation. Il est à noter que cette nouvelle calibration ne change pas les scores de capacités et de manques, qui sont basés sur la proportion de comportements humains correctement reproduits. Cependant, il réduit le score d'erreur à 0 en n'utilisant que des ensembles de paramètres valides, et améliore le score de représentativité en choisissant des ensembles de paramètres pour les agents selon les comportements observés chez les humains.

5.6. Discussion

Notre méthode introduit des métriques pour mesurer les scores résultant et pour corriger les paramètres des agents selon une telle étude. Les scores d'erreurs, de manques, et de capacités permettent au concepteur de la simulation multi-agent de trouver combien d'archétypes de comportement humain ont été correctement reproduits, combien de comportements agents ne devraient pas apparaître, et combien de comportements humains sont manquants. Le taux de confiance donne des indications sur la fiabilité des classes en fonction de leurs effectifs. Ensuite, en étudiant uniquement les archétypes des comportements correctement reproduits, les scores de calibration donnent de l'information sur les proportions de chaque comportement et leurs relations à la calibration des agents.

Une des originalités de ce processus de calibration est le contexte des simulations participatives. La fonction cible du processus de calibration n'est pas comme habituellement (Fehler *et al.*, 2005) au niveau macroscopique mais à un niveau individuel. La réalité virtuelle requièrt que chaque agent adopte des comportements crédibles, *i.e.* des comportements pouvant être produits par des humains. Dans ce contexte, nous retirons premièrement les ensembles de paramètres qui ne produisent pas de comportements valides. Nous calibrons ensuite les proportions d'agents avec les ensembles de paramètres restants selon les données des participants humains. Un unique cycle de notre méthode assure que les comportements valides sont détectés et qu'ils sont produits en des proportions correctes, nonobstant les comportements manquants.

Dans le cas d'une boîte noire où le modèle agent est inconnu et ne peut être modifié, si des comportements sont manquants alors une solution est d'explorer l'espace des paramètres afin de trouver de nouveaux comportements d'agents. Ces nouvelles étapes ne requièrent pas une autre expérimentation avec des participants humains, puisque les données de référence sont déjà disponibles. Chaque nouveau cycle permet de trouver potentiellement de nouveaux ensembles de paramètres, soit dans des classes déjà *mixtes*, soit dans les classes de *manques* précédentes. Cela permet aussi de trouver quelles zones de l'espace des paramètres produisent des comportements invalides.

Dans le cas d'une boîte blanche où le modèle agent est connu et peut être potentiellement modifié, les données d'annotations expliquent les comportements manquants et les comportements erronés, permettant ainsi d'améliorer le modèle agent (Fehler *et*

al., 2006). De plus, l'exploration des ensembles de paramètres peut être guidé par la connaissance du modèle (Fehler *et al*., 2005).

6. Conclusion et perspectives

Cet article présente une méthode semi-automatique de calibration de simulations multi-agents participative basée sur la combinaison de classifications non supervisées de traces de simulation et d'annotations par des participants.

L'analyse objective utilise premièrement un algorithme de classification non supervisée appliqué aux traces de simulation dans le but de classifier les comportements des participants, et deuxièmement une méthode d'agrégation pour comparer les comportements des agents à ceux des humains. Cette comparaison nous permet d'évaluer la crédibilité des comportements des agents en termes de capacités, de manques et d'erreurs. Cette méthode est générique pour les systèmes multi-agents (sous la contrainte de disposer d'un dispositif permettant le contrôle d'un agent par un participant humain) où les agents ont pour but de reproduire les comportements humains. Son application à un nouveau domaine demande l'adaptation de certains outils tels que le choix du questionnaire de comportement qui est spécifique au domaine.

L'expérimentation permet de définir un ensemble initial de traces valides qui servent de points de référence pour la calibration du modèle multi-agent. La calibration des paramètres du modèle suit une approche itérative. À chaque itération, nous obtenons les manques et les erreurs du modèle afin de raffiner l'espace des paramètres. Nous générons de nouveaux agents qui sont agrégés aux classes précédentes et nous calculons de nouveaux scores pour l'itération suivante. Notre méthode de validation a été appliquée à la simulation de trafic routier et a montré que des paramètres peuvent être correctement associés à des catégories de comportement.

L'originalité de cette approche est double. Premièrement, elle combine une analyse automatique des comportements des agents via les traces de simulation avec une analyse subjective basée sur l'évaluation humaine des comportements des agents, dans le but de définir un contexte spécifique à la situation. Cette combinaison permet une analyse de quel espace des paramètres des agents virtuels produit quel comportement perçu. Secondement, elle itère l'analyse de la classification afin de raffiner les capacités des agents à reproduire des comportements humains, tout en réduisant les manques et en supprimant les erreurs.

Plusieurs extensions doivent être examinées. La méthode d'agrégation dépend d'un taux de tolérance dont la valeur peut impacter la qualité des résultats : cet impact devra être vérifié. La convergence du modèle n'a pas encore été étudiée. Notre algorithme de classification donne des scores qui pourraient être utilisés afin de stopper le processus, mais une preuve de convergence est nécessaire lorsque le cycle n'est pas utilisé pour explorer de nouveaux paramètres.

Bibliographie

Bosse T., Hoogendoorn M., Klein M. C., Treur J., Van Der Wal C. N. (2011). Agent-based analysis of patterns in crowd behaviour involving contagion of mental states. In *Modern approaches in applied intelligence*, p. 566–577. Springer.

Caillou P., Gil-Quijano J. (2012). Simanalyzer: Automated description of groups dynamics in agent-based simulations. , p. 1353–1354.

Caliński T., Harabasz J. (1974). A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, vol. 3, n° 1, p. 1–27.

Campano S., Sabouret N., Sevin E. de, Corruble V. (2013). An evaluation of the cor-e computational model for affective behaviors. In *Proceedings of the 2013 international conference on autonomous agents and multi-agent systems*, p. 745–752.

Champion A., Éspié S., Auberlet J. M. (2001). Behavioral road traffic simulation with archisim. In *Summer computer simulation conference*, p. 359–364.

Darty K., Saunier J., Sabouret N. (2014a). Agents behavior semi-automatic analysis through their comparison to human behavior clustering. In *Intelligent virtual agents*, p. 154–163.

Darty K., Saunier J., Sabouret N. (2014b). Analyse des comportements agents par agrégation aux comportements humains. In *22èmes journées francophones sur les systèmes multi-agents (jfsma 2014)*. Cépaduès.

Darty K., Saunier J., Sabouret N. (2014c). A method for semi-automatic explicitation of agent's behavior: application to the study of an immersive driving simulator. In *The 6th international conference on agents and artificial intelligence (icaart 2014)*, p. 81–91. SciTePress.

Delaherche E., Chetouani M., Mahdhaoui A., Saint-Georges C., Viaux S., Cohen D. (2012). Interpersonal synchrony: A survey of evaluation methods across disciplines. *Affective Computing, IEEE Transactions on*, vol. 3, n° 3, p. 349–365.

Doniec A., Mandiau R., Piechowiak S., Espié S. (2008). A behavioral multi-agent model for road traffic simulation. *Engineering Applications of Artificial Intelligence*, vol. 21, n° 8, p. 1443–1454.

Drogoul A., Corbara B., Fresneau D. (1995). Manta: New experimental results on the emergence of (artificial) ant societies. *Artificial Societies: the computer simulation of social life*, p. 190–211.

Espié S., Auberlet J. M. (2007). Archisim: A behavioral multi-actors traffic simulation model for the study of a traffic system including its aspects. *International Journal of ITS Research*, vol. 1.

Fehler M., Klügl F., Puppe F. (2005). Techniques for analysis and calibration of multi-agent simulations. , p. 305–321.

Fehler M., Klügl F., Puppe F. (2006). Approaches for resolving the dilemma between model structure refinement and parameter calibration in agent-based simulations. , p. 120–122.

Gonçalves J., Rossetti R. J. F. (2013). Extending sumo to support tailored driving styles. *1st SUMO User Conference, DLR, Berlin - Adlershof, Germany*, vol. 21, p. 205–211.

Gratch J., Marsella S. (2005). Evaluating a computational model of emotion. *Autonomous Agents and Multi-Agent Systems*, vol. 11, n° 1, p. 23–43.

Guyot P., Drogoul A. (2005). Multi-agent based participatory simulations on various scales. In *Massively multi-agent systems i*, p. 149–160. Springer.

Huraux T. (2015). Simulation multi-agent d'un système complexe : combiner des domaines d'expertise par une approche multi-niveau. le cas de la consommation électrique résidentielle [PhD thesis].

Lacroix B., Mathieu P., Kemeny A. (2012). Formalizing the construction of populations in multi-agent simulations. *Engineering Applications of Artificial Intelligence*.

Lester J. C., Converse S. A. *et al.* (1997). The persona effect: affective impact of animated pedagogical agents. In *Proceedings of the sigchi conference on human factors in computing systems*, p. 359–366.

Likert R. (1932). A technique for the measurement of attitudes. *Archives of psychology*.

Mathieu P., Brandouy O. (2010). A generic architecture for realistic simulations of complex financial dynamics. , p. 185–197.

Pelachaud C. (2009). Modelling multimodal expression of emotion in a virtual agent. *Phil. Trans. R. Soc. B: Biological Sciences*, vol. 364, n° 1535, p. 3539–3548.

Rand W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, vol. 66, n° 336, p. 846–850.

Reason J., Manstead A., Stradling S., Baxter J., Campbell K. (1990). Errors and violations on the roads: a real distinction? *Ergonomics*, vol. 33, n° 10-11, p. 1315–1332.

Salvador S., Chan P. (2007). Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis*, vol. 11, n° 5, p. 561–580.

Six L. (2014). *Vers un modèle de comportements de véhicules lourds en utilisant une méthode incrémentale basée sur la vérification et l'hystérésis: le modèle archipl*. Thèse de doctorat non publiée, UniversitÃ© Pierre et Marie Curie.

Taillandier P., Drogoul A. (2010). Supervised feature evaluation by consistency analysis: application to measure sets used to characterise geographic objects. In *Knowledge and systems engineering (kse), 2010 second international conference on*, p. 63–68.

Veremme A., Lefevre É., Morvan G., Dupont D., Jolly D. (2012). Evidential calibration process of multi-agent based system: An application to forensic entomology. *Expert Systems with Applications*, vol. 39, n° 3, p. 2361–2374.