

---

# Activité collective et personnages autonomes : moteur décisionnel reposant sur des relations de confiance

Lucile Callebert, Domitile Lourdeaux, Jean-Paul Barthès

Sorbonne universités, Université de technologie de Compiègne, CNRS, Heudiasyc  
UMR 7253, CS 60 319, 60 203 Compiègne, France

{lucile.callebert/domitile.lourdeaux/jean-paul.barthes}@hds.utc.fr

---

*RÉSUMÉ.* Lorsqu'ils travaillent en équipe, les humains font des erreurs. Pour entraîner un apprenant en environnement virtuel collaboratif à s'adapter à des coéquipiers ayant des comportements non optimaux, nous proposons (1) une augmentation du langage de description de l'activité ACTIVITY-DL ainsi que des mécanismes de propagation de contraintes qui faciliteront le raisonnement des agents ; et (2) un modèle d'agent dans lequel chaque agent est décrit par trois dimensions (intégrité, bienveillance, compétences) correspondant au modèle de confiance MDS. De plus chaque agent a des buts collectifs et personnels et des croyances sur l'intégrité, la bienveillance et les compétences de chaque autre. Ce modèle d'agent est associé à un moteur décisionnel permettant de générer des comportements proches de ceux des humains. En particulier, les agents prennent les autres en compte et sont capables de raisonner sur leurs croyances sur les autres, à la fois lorsqu'ils choisissent quel but privilégier (collectif ou individuel) et lorsqu'ils sélectionnent une tâche. Nous avons conduit une évaluation préliminaire dans laquelle les participants ont évalué les comportements générés avec notre système.

*ABSTRACT.* When working in teams, people make mistakes. To train someone in a collaborative virtual environment to adapt to teammates that behave non optimally, we propose (1) an augmentation of the ACTIVITY-Description Language as well as mechanisms of propagation of constraints that will facilitate agents' reasoning; and (2) an agent model in which each agent is described through three dimensions (integrity, benevolence, abilities) corresponding to the MDS trust model. Besides each agent has different personal and collective goals and has beliefs about others' integrity, benevolence and abilities. This agent model is associated to a decision-making system that allows agents to adopt human-like behaviors. In particular, agents take others into account and are able to reason on their beliefs about others both when choosing which goal (collective or individual) to focus on and when selecting a task. We conducted a preliminary evaluation in which participants evaluated the behaviors produced with our system.

*MOTS-CLÉS :* systèmes multi-agents, prise de décision, confiance, activité collective.

*KEYWORDS :* multi-agents systems, decision-making, trust, collective activity.

---

DOI:10.3166/RIA.31.153-181 © 2017 Lavoisier

## 1. Introduction

Les environnements virtuels sont très utilisés pour la formation, puisqu'ils apportent de nombreux avantages : ils permettent par exemple de réduire les coûts de formation et de garantir la sécurité de l'apprenant. Nous nous intéressons aux environnements virtuels pour la formation à des activités collaboratives. Dans ces environnements, l'apprenant peut par exemple être entouré de plusieurs personnages virtuels avec lesquels il devra collaborer. Les personnages doivent alors être capables de s'organiser, de manière à ce que l'équipe ainsi formée puisse accomplir la tâche collective et soit efficace. Pour générer de tels comportements, des personnages autonomes sont souvent utilisés (Hill Jr *et al.*, 2003 ; Pizzi, Cavazza, 2007). En effet, l'utilisation de personnages autonomes permet de générer un très grand nombre de scénarios différents, puisque le comportement des personnages n'a pas besoin d'être spécifié pour chacun des scénarios, ce qui demanderait un effort de programmation considérable.

L'objectif visé est alors souvent d'obtenir des comportements d'équipe optimaux : chaque membre de l'équipe prête attention aux autres et s'investit dans l'activité collective. L'équipe doit être organisée et efficace : l'accent est mis sur la performance. Cet objectif de performance d'une équipe d'agents reproduisant des comportements d'équipe humains est également présent dans le domaine des systèmes multi-agents (SMA) (Tambe, 1997 ; Hübner *et al.*, 2002). Cette problématique est abordée au travers de deux grands axes : (i) de nombreux travaux s'appuient sur des spécifications structurelle (description de la structure de l'équipe ; rôles, relations entre les rôles, etc.), fonctionnelle (description de l'activité à réaliser par l'équipe) et déontique (droits et obligations associés à chaque rôle) (Hubner *et al.*, 2007). La collaboration est alors assurée puisqu'elle est prédéfinie par ces spécifications. (ii) D'autres travaux mettent l'accent sur les communications et sur le partage d'information et d'objectifs entre les agents. Des formalisations telles que la théorie des intentions conjointes (Cohen, Levesque, 1990) ou des plans partagés (Grosz, Kraus, 1996) visent en effet à définir les éléments nécessaires, en termes de croyances mutuelles des agents, à l'organisation de l'équipe : établir ces croyances mutuelles nécessite un partage d'information entre les agents. De nombreux SMA visant à reproduire des comportements d'équipe humains s'appuient sur ces formalisations : on peut citer par exemple GRATE\* (Jennings, 1995) ou RETSINA-MAS (Giampapa, Sycara, 2002). Que ce soit dans le domaine des SMA ou des environnements virtuels, certains travaux combinent les deux approches : par exemple STEAM (Tambe, 1997) s'appuie sur la théorie des intentions conjointes et sur une description de l'activité à réaliser sous forme d'une hiérarchie d'opérateurs pour générer les comportements d'agents.

Si ces travaux partagent le même objectif de performance des comportements produits, nous adoptons une problématique différente : nous souhaitons reproduire des comportements semblables à ceux des humains, c'est-à-dire des comportements qui peuvent (i) être optimaux et (ii) ne pas être optimaux. Dans tous les cas, ces comportements doivent être explicables, de manière à ce que l'apprenant qui y sera confronté puisse les comprendre. De plus, nous nous intéressons aux équipes dans lesquelles les rôles ne sont pas prédéfinis et qui sont auto-organisées : les membres de l'équipe

s'adaptent à la situation et aux comportements des autres, la prise de décision est décentralisée entre les membres (Anderson, Franks, 2003). Enfin, nous souhaitons en particulier reproduire les processus d'organisation implicite des équipes. En effet, si les communications favorisent la collaboration, elles ne sont pas indispensables : l'équipe peut interagir sans communication directe mais au travers de l'observation de l'environnement et des comportements des autres (Cao *et al.*, 1997). Cette observation du comportement des autres doit alors permettre aux agents de comprendre les autres, d'anticiper leurs comportements et d'en tenir compte lors de leur prise de décision. Pour générer de tels comportements, nous proposons de nous appuyer sur les relations de confiance. La confiance est en effet un facteur qui influence toutes nos décisions (Gambetta, 2000). En particulier dans les situations de collaboration, il est judicieux de se poser des questions telles que *Ai-je confiance en les compétences de mon coéquipier?* avant de décider d'une tâche à faire.

Dans cet article, nous présentons un système permettant la génération de comportements de personnages autonomes dans des activités collectives et tenant compte des relations de confiance. Les sections 2 et 3 sont consacrées à la présentation de travaux portant respectivement sur la confiance dans le domaine des sciences humaines et sociales (SHS) et sur l'utilisation de la confiance pour la génération de comportements d'agents en informatique. Puisque nous nous intéressons aux activités collectives, nous présentons section 4 nos contributions portant sur un formalisme de description de ces activités permettant de supporter le raisonnement des agents, avant d'exposer, section 5, le fonctionnement de notre moteur décisionnel permettant de générer les comportements d'agents. Finalement, nous décrivons une évaluation préliminaire de nos travaux section 6 avant de conclure section 7.

## 2. Fondements théoriques

Si la confiance est parfois considérée comme influençant toutes nos décisions ou comme constituant la "glue de la société" (Castelfranchi, Falcone, 2010), elle est primordiale dans les situations de travail collectif où plusieurs personnes peuvent être amenées à collaborer. En SHS, l'influence de la confiance sur les comportements a notamment été étudiée au travers de jeux type dilemme du prisonnier, et plusieurs auteurs ont montré que les individus coopèrent plus lorsqu'ils font confiance (Chaudhuri *et al.*, 2002; Wong *et al.*, 2005). Au niveau des groupes, une influence positive de la confiance sur la performance du groupe a été observée (Palanski *et al.*, 2011). Cependant, avoir *trop* confiance peut être dommageable, car cela empêche les membres d'un groupe de remettre en cause les compétences ou les idées des autres, ce qui peut conduire à une performance du groupe médiocre voire catastrophique (Erdem, 2003). La confiance doit donc être mesurée pour que les comportements individuels soient optimaux et la performance du groupe maximale : (Marsh, Briggs, 2009) introduisent le concept de *confiance mal placée* : notre confiance est mal placée par exemple lorsque l'on a trop ou trop peu confiance en la compétence de quelqu'un étant donné son niveau de compétence réel.

Nous nous intéressons particulièrement dans nos travaux à la *confiance interpersonnelle*, c'est-à-dire à la relation de confiance qui lie deux individus (Castaldo, 2002). En effet, s'appuyer sur une telle relation de confiance pour prendre une décision permettra à un agent de tenir compte de chacun des autres et de faire un choix informé sur la tâche à faire. Il n'y a pas de consensus dans la littérature SHS sur le caractère contextualisé (i.e. *i* fait confiance à *j* pour faire *X*) ou décontextualisé (i.e. *i* fait confiance à *j* de manière générale) de cette confiance interpersonnelle. Néanmoins dans le cadre de nos travaux, les tâches que pourront réaliser les agents nécessitant des compétences diverses, nous souhaitons pouvoir distinguer la confiance de *i* en *j* pour faire la tâche  $\tau_1$  de la confiance de *i* et *j* pour faire la tâche  $\tau_2$  : nous adoptons une définition de la confiance comme étant contextualisée. Enfin, bien qu'il n'y ait pas de consensus sur une définition de la confiance, il semble que la notion de croyances sur le futur soit commune à nombre de définitions. En effet, lorsqu'il fait confiance à *j*, l'agent *i* est en attente d'un comportement particulier de la part de *j* (Mayer *et al.*, 1995 ; Castelfranchi, Falcone, 2010 ; Rousseau *et al.*, 1998). Cette prédiction de *i* sur le comportement de *j* peut s'appuyer notamment sur une évaluation par *i* de certaines caractéristiques de *j*, comme sa compétence. La confiance mal (*resp.* bien) placée de *i* en *j* consiste donc en une prédiction de *i* sur le comportement de *j* qui aura de grandes chances de se révéler fausse (*resp.* juste) car *i* sous-estime ou surestime (*resp.* estime correctement) certaines caractéristiques de *j*.

De nombreux auteurs se sont alors attachés à déterminer quelles sont les caractéristiques ou dimensions de *j* qui sont évaluées par *i* lorsque *i* décide de faire confiance à *j*, et de nombreux modèles dimensionnels de la confiance ont alors été proposés. Dans (Mishra, 1996) par exemple, les dimensions de *compétence*, d'*ouverture d'esprit*, de *soucis* (i.e. *j* est perçu par *i* comme étant généralement attentif aux autres et concerné par leurs intérêts) et de *fiabilité* ont été proposées ; cependant dans ce modèle la confiance n'est pas contextualisée. Nous avons choisi dans nos travaux de nous appuyer sur le modèle de confiance proposé par (Mayer *et al.*, 1995), que nous désignons par les termes "modèle MDS" (du nom des auteurs Mayer, Davis et Schoorman) dans la suite de ce papier. En effet, ce modèle fait partie des travaux majeurs du domaine, et il a notamment été validé expérimentalement (Jarvenpaa *et al.*, 1998 ; Gill *et al.*, 2005), ce qui n'est pas le cas pour la plupart des autres modèles (Mishra, 1996 ; McKnight, Chervany, 2001). Les dimensions du modèle MDS sont les suivantes : (i) *i* a confiance en l'intégrité de *j*, i.e. *i* pense que *j* est honnête et respecte ses engagements ; (ii) *i* a confiance en la bienveillance de *j* à son égard, i.e. *i* pense que *j* veut son bien ; (iii) *i* a confiance en la compétence de *j* sur une tâche particulière.

Nous nous intéressons dans la section suivante aux travaux sur la génération de comportements d'agents qui utilisent la notion de confiance.

### 3. Travaux connexes

La confiance a beaucoup été étudiée dans le domaine des environnements virtuels tout comme dans celui dans SMA. Dans le domaine des environnements virtuels la

confiance a servi de support à la prise de décision pour un agent, notamment dans des situations où l'agent doit décider de croire ou non les informations transmises par un autre. Elle peut alors être modélisée comme une simple valeur numérique (Marsella *et al.*, 2004) ou comme un vecteur à plusieurs dimensions : (Traum *et al.*, 2005) utilisent les dimensions de *solidarité* (i.e. similarité entre mes buts et ceux de l'autre), de *crédibilité* (i.e. cohérence des dires de l'autre par rapport à mes croyances) et de *familiarité* (i.e. respect par l'autre des normes sociales appropriées). La définition de ces différentes dimensions ne s'appuie cependant pas sur un modèle de confiance issu de théories SHS et est donc difficilement justifiables, ce qui peut nuire à l'explicitabilité des comportements produits.

Les systèmes de confiance sont communs dans le domaine des SMA. Une revue détaillée et une classification des principaux systèmes de confiance sont proposées dans (Pinyol, Sabater-Mir, 2013). Les auteurs différencient les systèmes de confiance numériques des systèmes cognitifs. Les premiers utilisent des approches issues de la théorie de jeux et ne permettent pas une représentation explicite des éléments cognitifs qui composent la confiance. À l'inverse, les systèmes cognitifs accordent autant d'importance au processus de prise de décision qu'à la décision de faire confiance elle-même, et les éléments sur lesquels s'appuie cette décision sont formalisés en termes d'éléments cognitifs tels que les croyances et les désirs, ce qui permet d'expliquer la décision prise. De plus, les définitions de la confiance utilisées dans ces systèmes s'appuient sur des théories de SHS, ce qui permet de justifier les comportements produits. Nous nous intéressons donc aux systèmes de confiance cognitifs, et en particulier aux trois systèmes suivants : (1) Le système BDI+Repage (Pinyol *et al.*, 2012) s'appuie sur les concepts d'image (dont la définition est similaire à celle de la confiance :  $i$  a une croyance sur  $j$ ) et de réputation ( $i$  pense que le groupe  $I$  a une croyance sur  $j$ ). Les croyances d'un agent  $i$  sur l'image et la réputation d'un agent  $j$  ayant le rôle  $r$  sont représentées par des ensembles flous. (2) Un modèle computationnel de la confiance sociocognitive est proposé par (Castelfranchi, Falcone, 2010). Les auteurs proposent que la confiance d'un agent  $i$  en un agent  $j$  est nécessairement située par rapport à un but  $g$  (i.e.  $i$  a un but  $g$  et  $i$  a ou n'a pas confiance en  $j$  pour réaliser ce but  $g$ ) et est composée de plusieurs dimensions : la *capacité* et la *volonté* de  $j$  à réaliser  $g$  composent le noyau de la confiance, alors que l'*opportunité* de  $j$  et la *dépendance* de  $i$  à  $j$  pour réaliser  $g$  sont d'autres éléments à prendre en compte. (3) Enfin, une formalisation en logique multimodale de la confiance s'appuyant sur les travaux de (Castelfranchi, Falcone, 2010) est proposée dans ForTrust (Herzig *et al.*, 2010; Lorini, Demolombe, 2008), où la confiance occurrente (i.e.  $i$  a confiance en  $j$  pour réaliser  $g$  en faisant l'action  $\alpha$  dans l'immédiat) est distinguée de la confiance dispositionnelle (i.e.  $i$  n'a pas le but  $g$ , mais  $i$  a confiance en  $j$  pour réaliser  $g$  s'il devait avoir le but  $g$ ).

Dans tous ces travaux, les processus de prise de décision à partir des valeurs de confiance sont explicités. De plus, la confiance est contextualisée. ForTrust et le modèle de (Castelfranchi, Falcone, 2010) permettent également de distinguer plusieurs dimensions dans la confiance, ce qui n'est pas le cas dans BDI+Repage mais paraît essentiel : en effet, avoir confiance en la compétence de  $j$  pour faire une tâche n'est pas suffisant, encore faut-il penser que  $j$  ait l'intention de faire la tâche. Au regard

de nos objectifs, la principale limite de ces travaux est que ces systèmes sont centrés sur la notion de *délégation* : l'agent  $i$  peut-il déléguer une tâche à l'agent  $j$  ? Or nous souhaitons proposer un système dans lequel les agents ne seront pas seulement capables de déléguer, mais seront également capables d'anticiper des comportements des autres et de raisonner sur ces projections de manière à sélectionner l'action qu'eux-mêmes feront. Une telle anticipation doit notamment permettre aux agents d'adopter des comportements d'aide (e.g. l'agent  $j$  a un but  $g$  et  $i$  anticipe que  $j$  n'est pas capable de réaliser seul ce but  $g$  : il aide alors  $j$  à réaliser  $g$ ), qui sont des comportements naturellement adoptés dans des équipes de travail humaines.

Nous détaillons dans les sections suivantes notre proposition sur un formalisme de description de l'activité permettant de supporter le raisonnement des agents (section 4) et sur un moteur décisionnel permettant de générer de tels comportements (section 5).

#### 4. Description et traitement de l'activité collective

Nous souhaitons proposer un système qui permette à une équipe de personnages virtuels de s'auto-organiser sans que cette organisation ne repose sur des rôles prédéfinis. Les comportements générés devront néanmoins être représentatifs de comportements que l'on pourrait observer en réalité : nous avons donc besoin d'une description de l'activité à réaliser, dans un formalisme qui pourra être interprété par les agents. Pour cela, nous souhaitons que la description de l'activité puisse prendre la forme d'un arbre dont la racine correspondrait à la tâche de plus haut niveau, c'est-à-dire au but que les agents doivent atteindre. Chaque tâche est ensuite décomposée en sous-tâches, et les tâches feuilles de l'arbre correspondent à des actions. Une telle description de l'activité par une structure arborescente est classiquement utilisée dans de nombreux travaux (Tambe, 1997 ; Porteous *et al.*, 2013) et facilite le raisonnement des agents et la répartition des tâches (i.e. il est plus simple de raisonner sur une seule tâche de haut niveau que sur plusieurs dizaines d'actions). En outre, le contexte de nos travaux est celui de la formation en environnement virtuel : le langage de description de l'activité doit donc être expressif, de façon à pouvoir décrire précisément les comportements attendus.

##### 4.1. Choix d'un langage de description de l'activité

Il existe de très nombreux formalismes de description de l'activité (pour un état de l'art plus complet voir (Barot, 2014)). (i) Des règles type *si ... alors ...* peuvent être utilisées, cependant il peut être difficile de gérer un très grand nombre de règles. (ii) On peut utiliser des diagrammes d'activité : l'activité est alors découpée sous forme de rôles, et chaque rôle est décrit par une séquence d'actions à effectuer. Ce type de représentation ne convient cependant pas à notre problématique puisque nous nous focalisons sur des activités où les rôles ne sont pas prédéfinis. (iii) On peut utiliser des automates ou des réseaux de Pétri, comme dans LORA – *Language for Object Relation Application* (Gerbaud, 2008). Un tel formalisme de description de l'activité peut cependant être difficile à aborder pour les ergonomes ou experts du domaine qui

seront amenés à décrire les activités. (iv) Les opérateurs de planification permettent d'organiser la description de l'activité en une structure hiérarchique : l'opérateur de plus haut niveau correspond à l'objectif à atteindre alors que les opérateurs feuilles sont des actions à réaliser. Une telle structure hiérarchique est idéale pour supporter le raisonnement des agents mais ces langages sont difficilement accessibles à des non-informaticiens et leur expressivité est limitée. (v) ACTIVITY-DL est un langage inspiré de travaux en ergonomie et est accessible aux non informaticiens. Il permet une description de l'activité sous forme d'arbre de tâches et supporte le raisonnement des agents. Par ailleurs, ACTIVITY-DL est un langage très expressif qui permet par exemple de spécifier plusieurs types de précondition différents selon que ces dernières doivent être vérifiées (e.g. pour *ouvrir la porte*, la porte doit être fermée) ou qu'elles rendent l'exécution de la tâche favorable (e.g. *se laver les mains* n'est pertinent que si les mains sont sales). En environnement virtuel de formation, la description de l'activité pouvant également servir au suivi de l'apprenant, l'expressivité du langage est un atout non négligeable. C'est donc sur ACTIVITY-DL que s'est porté notre choix.

## 4.2. Description de l'activité avec ACTIVITY-DL

Nous expliquons brièvement en 4.2.1 les spécificités de ACTIVITY-DL sur lesquelles repose notre contribution, détaillée ensuite. Plus de détails sur le langage peuvent être trouvés dans (Barot, 2014).

### 4.2.1. ACTIVITY-DL : contraintes logiques et temporelles

ACTIVITY-DL permet de décrire l'activité sous forme d'un arbre de tâches. À chaque tâche composée est associée (1) une contrainte temporelle (e.g. les contraintes SEQ-ORD et PAR-SIM signifient que les sous-tâches doivent être exécutées respectivement séquentiellement dans l'ordre donné et simultanément alors que IND marque l'absence de contrainte temporelle sur la réalisation des sous-tâches) et (2) une contrainte logique (e.g. ET signifie que toutes les sous-tâches doivent être réalisées pour valider la tâche mère et OU signifie qu'une seule des sous-tâches doit être réalisée pour valider la tâche mère).

### 4.2.2. ACTIVITY-DL : aspects collaboratifs

Bien qu'il soit très expressif, le langage ACTIVITY-DL a été pensé pour la description de l'activité d'un individu, et nous avons proposé d'ajouter plusieurs éléments pour supporter la description des activités collectives (Callebert *et al.*, 2016b ; 2016a). Sur chaque tâche *feuille*  $\tau_k$  peuvent être renseignés : (i) un nombre d'intervenants minimum  $n_{min}(\tau_k)$  et maximum  $n_{max}(\tau_k)$  et (ii) une compétence nécessaire, notée  $\kappa$ . Cet élément nous permet en particulier de faire un lien entre la description de l'activité et le modèle de confiance MDS qui servira de support à notre moteur décisionnel.

#### 4.2.3. Exemple de description d'activité en ACTIVITY-DL

Un exemple d'arbre de tâches ACTIVITY-DL est présenté dans la figure 1. Dans cet arbre, les éléments (i) de nombre d'intervenants et (ii) de compétences sont spécifiés dans les bulles blanches associées à chaque tâche feuille (en bas à droite). Par ailleurs, les tâches sont numérotées (en haut à gauche) et pour désigner une tâche particulière on utilisera la notation  $\tau_{\text{numéro}}$ .

Dans cet exemple, la tâche  $\tau_1 = \text{monter le bureau}$  est racine de l'arbre et correspond donc au but à atteindre. Pour la réaliser, les tâches  $\tau_{11} = \text{préparer les outils}$ ,  $\tau_{12} = \text{transporter les éléments}$  et  $\tau_{13} = \text{assembler les éléments}$  doivent toutes être réalisées séquentiellement (contraintes ET/SEQ-ORD) :

- $\tau_{11}$  est également une tâche composée, dont une seule tâche fille  $\tau_{111} = \text{chercher les outils}$  ou  $\tau_{112} = \text{acheter les outils}$  a besoin d'être réalisée (contrainte OU/IND).  $\tau_{111}$  est une action pour laquelle un nombre quelconque de personnes peuvent intervenir :  $n_{\min}(\tau_{111}) = 1$  et  $n_{\max}(\tau_{111}) = +\infty$  (noté INF sur la figure). Aucune compétence n'est nécessaire pour faire cette action (NONE).  $\tau_{112}$  est une action pour laquelle une seule personne peut intervenir ( $n_{\min}(\tau_{112}) = n_{\max}(\tau_{112}) = 1$ ). Cette personne doit avoir le permis de conduire pour aller au magasin (compétence PERMIS).

- $\tau_{12}$  est une action qui se fait à deux et pour laquelle les intervenants doivent savoir bricoler (compétence BRICO) de manière à manipuler correctement les éléments.

- Enfin,  $\tau_{13}$  est une tâche composée dont les sous-tâches  $\tau_{131} = \text{vérifier l'emplacement}$  des éléments par rapport au plan,  $\tau_{132} = \text{maintenir les éléments}$  et  $\tau_{133} = \text{clouer les éléments}$  doivent être faites simultanément (contrainte ET/PAR-SIM).  $\tau_{132}$  se fait seul et ne nécessite pas de compétence.  $\tau_{131}$  tout comme  $\tau_{133}$  se font seul également mais nécessitent d'avoir la compétence BRICO.

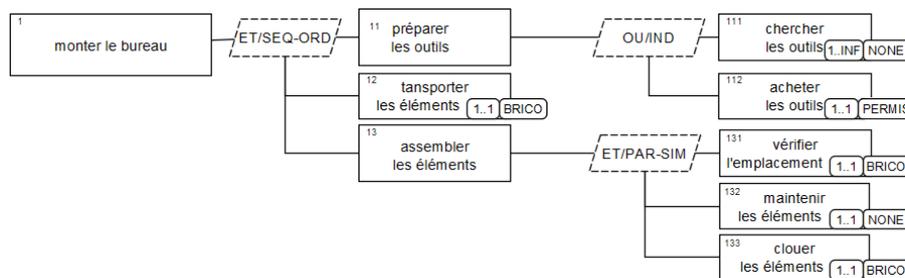


Figure 1. Exemple d'un arbre de tâches décrit en ACTIVITY-DL. Les éléments de nombre d'intervenants et de compétences renseignés lors de la description de l'activité figurent dans les bulles blanches en bas de chaque tâche

#### 4.2.4. Génération et propagation de contraintes et d'informations

Pour faciliter le processus de description de l'activité et assurer la cohérence de la description obtenue, ces éléments ne sont renseignés que sur les tâches feuilles.

Néanmoins, les agents raisonnant en priorité sur des tâches abstraites, nous avons proposé dans (Callebert *et al.*, 2016b ; 2016a) de propager ces éléments depuis les tâches feuilles jusqu'aux tâches racines sous forme de contraintes et d'informations :

- Les *contraintes* générées à partir des éléments spécifiés sur les tâches feuilles doivent permettre aux agents de déterminer si l'équipe a les ressources nécessaires en termes de (i) nombre d'intervenants (voir 4.3) et de (ii) compétences pour pouvoir réaliser l'activité collective (voir 4.4). Ces contraintes sont donc générées en début de simulation et sont statiques.

- Les *informations* générées à partir des éléments spécifiés sur les tâches feuilles doivent permettre à chaque agent de déterminer s'il peut personnellement prendre part à l'activité à un instant  $t$  ; elles sont dynamiques et actualisées chaque fois qu'une action est terminée dans l'environnement virtuel. Pour qu'un agent puisse prendre part à une tâche, en dehors du fait que les pré-conditions de la tâche doivent être vérifiées, (i) l'agent doit avoir les compétences nécessaires pour participer à la tâche (voir 4.6) et (ii) le nombre d'intervenants maximal ne doit pas être atteint (voir 4.5). Les informations propagées portent donc sur ces deux éléments.

#### 4.3. Propagation des contraintes en termes de nombre d'intervenants

Pour chaque tâche composée  $\tau_k$  dont les  $n$  sous-tâches sont celles de l'ensemble  $\{\tau_{k1}, \dots, \tau_{kn}\}$ , nous cherchons à exprimer  $n_{min}(\tau_k)$  le nombre minimal d'intervenants nécessaires pour réaliser  $\tau_k$  en fonction de  $n_{min}(\tau_{k1}), \dots, n_{min}(\tau_{kn})$ . Ce nombre minimal d'intervenants est dépendant des contraintes logique et temporelle associées à  $\tau_k$ , mais les règles de propagations données dans la formule 1 sont relativement simples et nous les abordons au travers de l'exemple développé en section 4.2.3 :

- Cas OU/SEQ-ORD et OU/IND. Un seul intervenant est nécessaire pour réaliser  $\tau_{11}$ , car il suffit de réaliser  $\tau_{111}$  ou  $\tau_{112}$ , qui nécessitent chacune un intervenant au minimum.

- Cas ET/PAR-SIM. Trois intervenants sont nécessaires pour réaliser  $\tau_{13}$  :  $\tau_{131}$ ,  $\tau_{132}$  et  $\tau_{133}$  nécessitent chacune un intervenant au minimum, mais doivent *toutes* être réalisées *simultanément*.

- Cas ET/SEQ-ORD et ET/IND. La tâche  $\tau_1$  nécessite au minimum trois intervenants, puisque  $\tau_{11}$ ,  $\tau_{12}$  et  $\tau_{13}$  doivent *toutes* être réalisées *séquentiellement* et nécessitent au minimum un, deux et trois intervenants respectivement.

$$\begin{cases} n_{min}(\tau_k) = \min_{i \in [1;n]}(n_{min}(\tau_{ki})) & \text{si OU/SEQ-ORD ou OU/IND} \\ n_{min}(\tau_k) = \sum_{i=1}^n n_{min}(\tau_{ki}) & \text{si ET/PAR-SIM} \\ n_{min}(\tau_k) = \max_{i \in [1;n]}(n_{min}(\tau_{ki})) & \text{si ET/SEQ-ORD ou ET/IND} \end{cases} \quad (1)$$

#### 4.4. Propagation des contraintes en termes de compétences

La propagation des contraintes en termes de compétences est plus complexe, car ces contraintes doivent permettre de répondre à une question plus difficile : *Quelles compétences faut-il pour faire la tâche  $\tau_k$  et combien d'intervenants doivent avoir chaque compétence pour faire la tâche  $\tau_k$  ?* Par exemple, on a besoin de savoir que pour réaliser  $\tau_k$ , il faut (*option 1*) un intervenant ayant la compétence X et deux intervenants ayant la compétence Y **OU** (*option 2*) un intervenant ayant la compétence X et trois intervenants ayant la compétence Z.

##### 4.4.1. Formalisation

Nous définissons  $c_{col}(\tau_k)$  la contrainte en termes de compétences attachée à  $\tau_k$ . Par ailleurs, pour exprimer le fait que  $n_i(\tau_k)$  intervenants devront avoir la compétence  $\kappa_i$  pour réaliser la tâche  $\tau_k$ , nous utilisons la notation  $n_i(\tau_k) \cdot \kappa_i$ . De plus, si une tâche feuille ne nécessite aucune compétence, nous exprimons cette information par la notation  $c_{col}(\tau_k) = \text{NONE}$ .

Pour les tâches composées, nous exprimerons toujours  $c_{col}(\tau_k)$  comme une forme normale disjonctive (FND) de manière à exprimer clairement les différentes options. Formellement,  $c_{col}(\tau_k)$  sera donc de la forme définie dans la formule 2, où  $n(k)$  est le nombre de conjonctions  $C_j(\tau_k)$  qui composent  $c_{col}(\tau_k)$ . Les conjonctions  $C_j(\tau_k)$  sont de la forme définie dans la formule 3 où  $m(j)$  représente le nombre de compétences de la conjonction, et  $n_x(C_j(\tau_k))$  représente le nombre de personnes devant avoir la compétence  $\kappa_x$  pour faire  $\tau_k$  selon la conjonction  $C_j(\tau_k)$ .

$$c_{col}(\tau_k) = C_1(\tau_k) \vee \dots \vee C_{n(k)}(\tau_k) = \bigvee_{j=1}^{n(k)} C_j(\tau_k) \quad (2)$$

$$C_j(\tau_k) = n_1(C_j(\tau_k)) \cdot \kappa_1 \wedge \dots \wedge n_{m(j)}(C_j(\tau_k)) \cdot \kappa_{m(j)} \quad (3)$$

##### 4.4.2. Propagation

Nous cherchons à exprimer, pour une tâche composée  $\tau_k$  dont les  $n$  sous-tâches sont celles de l'ensemble  $\{\tau_{k1}, \dots, \tau_{kn}\}$ , la contrainte en termes de nombre d'intervenants  $c_{col}(\tau_k)$  en fonction des contraintes  $c_{col}(\tau_{ki})$  attachées aux sous-tâches de  $\tau_k$ . De même que pour les contraintes en termes de nombre d'intervenants, les règles de propagation des contraintes en termes de compétences sont dépendantes des contraintes logiques et temporelles associées à chaque tâche.

**Contrainte logique OU.** Dans les cas OU/SEQ-ORD et OU/IND,  $c_{col}(\tau_k)$  est simplement la disjonction des contraintes en termes de compétences  $c_{col}(\tau_{ki})$  attachées à chacune des tâches filles  $\tau_{ki}$ , comme exprimé dans la formule 4. Dans l'exemple

présenté section 4.2.3, on a  $c_{col}(\tau_{111}) = \text{NONE}$  et  $c_{col}(\tau_{112}) = 1 \cdot \text{PERMIS}$ , ce qui donne  $c_{col}(\tau_{11}) = 1 \cdot \text{PERMIS} \vee \text{NONE}$ , ce qui revient à  $c_{col}(\tau_{11}) = \text{NONE}$ .

$$c_{col}(\tau_k) = \bigvee_{i=1}^n c_{col}(\tau_{ki}) = \bigvee_{i=1}^n \left( \bigvee_{j=1}^{n(ki)} C_j(\tau_{ki}) \right) = \bigvee_{j=1}^{\sum_{i=1}^n n(ki)} C_j(\tau_k) \quad (4)$$

**Contrainte logique ET.** Dans le cas où la contrainte logique est ET,  $c_{col}(\tau_k)$  est la conjonction des  $c_{col}(\tau_{ki})$ , comme exprimé dans la formule 5.

$$c_{col}(\tau_k) = \bigwedge_{i=1}^n c_{col}(\tau_{ki}) = \bigwedge_{i=1}^n \left( \bigvee_{j=1}^{n(ki)} C_j(\tau_{ki}) \right) \quad (5)$$

Pour obtenir une FND telle que définie dans la formule 2, il est alors nécessaire de transformer la contrainte obtenue en appliquant la règle de distributivité de l'opérateur  $\wedge$ . Néanmoins, les conjonctions  $C_j(\tau_k)$  ainsi obtenues peuvent comporter plusieurs fois la même compétence. Dans notre exemple, on a  $c_{col}(\tau_{131}) = 1 \cdot \text{BRICO}$ ,  $c_{col}(\tau_{132}) = \text{NONE}$  et  $c_{col}(\tau_{133}) = 1 \cdot \text{BRICO}$ , ce qui donne  $c_{col}(\tau_{13}) = 1 \cdot \text{BRICO} \wedge \text{NONE} \wedge 1 \cdot \text{BRICO} = 1 \cdot \text{BRICO} \wedge 1 \cdot \text{BRICO}$ . Le calcul du nombre de participants nécessaires pour chaque compétence dépend alors de la contrainte temporelle associée à la tâche :

- Cas PAR-SIM. Si les sous-tâches de  $\tau_k$  doivent être faites **simultanément**, l'expression de  $c_{col}(\tau_{13})$  peut être simplifiée en  $c_{col}(\tau_{13}) = (\mathbf{1} + \mathbf{1}) \cdot \text{BRICO}$ .
- Cas SEQ-ORD et IND. De même que pour  $c_{col}(\tau_{13})$ ,  $c_{col}(\tau_1)$  fait apparaître plusieurs fois la même compétence :  $c_{col}(\tau_1) = \text{NONE} \wedge 2 \cdot \text{BRICO} \wedge 2 \cdot \text{BRICO}$ . Cela peut être simplifié en  $c_{col}(\tau_1) = \mathbf{max(2;2)} \cdot \text{BRICO}$ , car les tâches doivent être faites **séquentiellement**.

Formellement, lorsqu'une conjonction  $C_j(\tau_k)$  est de la forme exprimée dans la formule 6 (i.e. contient plusieurs fois la même compétence  $\kappa_\ell$ ), alors le nombre d'agents  $n_\ell(C_j(\tau_k))$  devant avoir cette compétence est donné par la formule 7.

$$C_j(\tau_k) = n_{1,1}(C_j(\tau_k)) \cdot \kappa_1 \wedge \dots \wedge n_{1,m_1}(C_j(\tau_k)) \cdot \kappa_1 \wedge n_{x,1}(C_j(\tau_k)) \cdot \kappa_x \wedge \dots \wedge n_{x,m_x}(C_j(\tau_k)) \cdot \kappa_x \quad (6)$$

$$\begin{cases} n_\ell(C_j(\tau_k)) = \max_{\alpha \in [1; m_\ell]} (n_{\ell, \alpha}(C_j(\tau_k))) & \text{si SEQ-ORD ou IND} \\ n_\ell(C_j(\tau_k)) = \sum_{\alpha=1}^{m_\ell} n_{\ell, \alpha}(C_j(\tau_k)) & \text{si PAR-SIM} \end{cases} \quad (7)$$

#### 4.5. Propagation des informations sur le nombre maximal d'intervenants

Nous définissons  $n_{max}(\tau_k, t)$  le nombre d'intervenants maximum qui peuvent participer à la réalisation de la tâche  $\tau_k$  à l'instant  $t$ . Ici, la propagation des informations n'est dépendante que de la contrainte temporelle associée à  $\tau_k$  :

– Cas IND et PAR-SIM. Dans notre exemple, à  $t_0$  aucune des tâches n'est réalisée, et  $\tau_{111}$  et  $\tau_{112}$  *peuvent* être réalisées *simultanément* par respectivement 1 et  $+\infty$  intervenants au maximum; on a donc  $n_{max}(\tau_{11}, t_0) = 1 + \infty = +\infty$ . De même,  $\tau_{131}$ ,  $\tau_{132}$  et  $\tau_{133}$  *doivent* être réalisées *simultanément* par au maximum une personne chacune; on a donc  $n_{max}(\tau_{13}, t_0) = 1 + 1 + 1 = 3$ .

– Cas SEQ-ORD. Les tâches  $\tau_{11}$ ,  $\tau_{12}$  et  $\tau_{13}$  *doivent* être réalisées *séquentiellement*, et c'est donc  $\tau_{11}$  qui devra être faite en premier. À  $t_0$ , on a donc  $n_{max}(\tau_1, t_0) = n_{max}(\tau_{11}, t_0) = +\infty$ .

Formellement, les règles de propagation des informations dynamiques en termes de nombre maximal d'intervenants sont données dans la formule 8, où  $\tau_{kk}$  est la première sous-tâche de  $\tau_k$  que l'on n'a pas encore réalisée (cas SEQ-ORD), et  $\{\tau_{k1}, \dots, \tau_{km}\}$  est l'ensemble des  $m$  sous-tâches de  $\tau_k$  qui ne sont pas encore réalisées (cas IND et PAR-SIM).

$$\begin{cases} n_{max}(\tau_k, t) = n_{max}(\tau_{kk}, t) & \text{si SEQ-ORD} \\ n_{max}(\tau_k, t) = \sum_{i=1}^m n_{max}(\tau_{ki}, t) & \text{si PAR-SIM ou IND} \end{cases} \quad (8)$$

#### 4.6. Propagation des informations sur la compétence

Les informations sur les compétences doivent permettre à un agent de déterminer s'il a personnellement les compétences requises pour participer à une tâche. Dans notre exemple, l'*équipe* doit posséder la compétence BRICO pour réaliser  $\tau_1$  (*contrainte* en termes de compétences), alors qu'à  $t_0$ , un *agent* ne possédant pas cette compétence peut participer  $\tau_1$  en faisant  $\tau_{111}$  ou  $\tau_{112}$  (*informations* sur les compétences attachées à  $\tau_1$ ).

Nous désignons par  $c(\tau_k, t)$  les informations dynamiques en termes de compétences attachées à  $\tau_k$  à l'instant  $t$ , et  $c(\tau_k, t)$  sera une disjonction de compétences :  $c(\tau_k, t) = \kappa_1 \vee \kappa_2 \vee \dots \vee \kappa_m$ . Tout comme pour la propagation des informations dynamiques sur le nombre d'intervenants, les informations dynamiques sur les compétences sont dépendantes des contraintes temporelles associées à chaque tâche composée :

– Cas IND ou PAR-SIM. Les deux sous-tâches de  $\tau_{11}$  pouvant être réalisées simultanément et  $\tau_{112}$  nécessitant la compétence PERMIS, un agent qui veut participer à  $\tau_{11}$  peut avoir la compétence PERMIS ou ne pas avoir de compétence :  $c(\tau_{11}, t_0) = \text{PERMIS} \vee \text{NONE}$ . De même,  $c(\tau_{13}, t_0) = \text{BRICO} \vee \text{NONE}$ . À noter qu'ici et contrairement à  $c_{col}(\tau_k)$ , les éléments autres que NONE ne doivent pas être supprimés car l'agent aura besoin de connaître l'*ensemble* des compétences qui pourraient être utiles (voir 5.3).

– Cas SEQ-ORD. À  $t_0$ , la sous-tâche de  $\tau_1$  à faire est  $\tau_{11}$ ; on a donc  $c(\tau_1, t_0) = c(\tau_{11}, t_0) = \text{PERMIS} \vee \text{NONE}$ , tandis qu'à  $t_1$  (i.e.  $\tau_{11}$  est réalisée), on aura  $c(\tau_1, t_1) = \text{BRICO}$ .

Les règles de propagation des informations dynamiques en termes de compétences sont données dans la formule 9 où  $\tau_{kk}$  est la première sous-tâche de  $\tau_k$  que l'on n'a pas encore réalisée (cas SEQ-ORD), et  $\{\tau_{k1}, \dots, \tau_{km}\}$  est l'ensemble des  $m$  sous-tâches de  $\tau_k$  qui ne sont pas encore réalisées (cas IND et PAR-SIM).

$$\begin{cases} c(\tau_k, t) = c(\tau_{kk}, t) & \text{si SEQ-ORD} \\ c(\tau_k, t) = c(\tau_{k1}, t) \vee \dots \vee c(\tau_{km}, t) & \text{si PAR-SIM ou IND} \end{cases} \quad (9)$$

#### 4.7. Conclusion sur la description et le traitement de l'activité

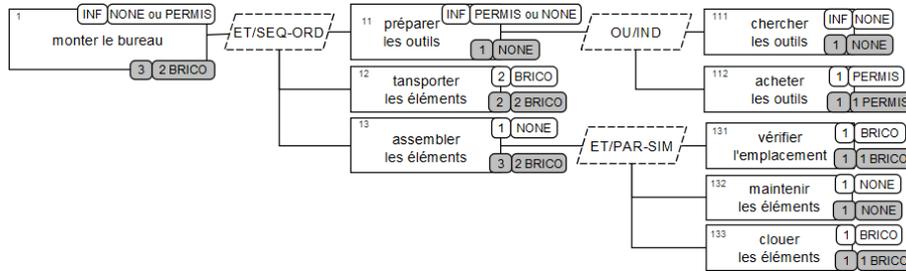


Figure 2. Même arbre de tâches que celui présenté en figure 1 sur lequel apparaissent à droite de chaque tâche les contraintes (bulles grisées) et les informations (bulles blanches) générées pas le système

Nous avons choisi de décrire l'activité que les agents devront effectuer en ACTIVITY-DL, un langage approprié au contexte et aux objectifs que nous avons présentés en début de section. Nous avons proposé d'ajouter à ce langage la possibilité de spécifier sur les tâches feuilles (i.e. correspondant à des actions) des éléments de nombre d'intervenants et de compétences. En effet, nous souhaitons voir la collaboration émerger et nous avons donc décidé de ne pas imposer aux agents une activité à faire en fonction d'un rôle qu'ils joueraient.

Par ailleurs, les agents raisonnant en priorité sur des tâches abstraites de haut niveau, nous avons proposé de propager, des tâches filles aux tâches mères, des contraintes et des informations dynamiques générées automatiquement à partir des éléments de nombre d'intervenants et de compétences spécifiés sur les tâches feuilles. Nous avons détaillé ici les mécanismes de propagation de ces contraintes et de ces informations. La figure 2 présente l'arbre de tâches décrit à titre d'exemple en 4.2.3 après propagation des contraintes et informations. Nous présentons dans la section suivante le fonctionnement du moteur décisionnel de nos agents. Ce moteur décisionnel leur permet notamment de raisonner sur des arbres de tâches décrits en ACTIVITY-DL et plus particulièrement sur les contraintes et informations générées par notre système.

## 5. Moteur décisionnel

Notre moteur décisionnel doit permettre à chaque agent de sélectionner une action à faire dans l'environnement virtuel en tenant compte des autres. Pour cela les agents raisonneront sur des arbres de tâches ACTIVITY-DL et sur les relations de confiance qu'ils entretiennent avec chaque autre agent de l'environnement, décrites selon le modèle de confiance MDS. Nous présentons dans un premier temps le modèle d'agent sur lequel s'appuie notre moteur décisionnel, puis nous abordons le fonctionnement général du moteur décisionnel, avant d'en détailler chaque étape.

### 5.1. Modèle d'agent

Nos agents sont décrits par des buts, des dimensions personnelles et des croyances :

- Chaque agent  $x$  a deux types de buts : des buts personnels et des buts collectifs dont les ensembles sont respectivement notés  $\Gamma_{x,self}$  et  $\Gamma_{x,team}$ . Si  $x$  a un but collectif, alors  $x$  pense que ce but est partagé par au moins un autre agent de l'environnement. Par exemple, un agent Anna (noté  $a$ ) peut avoir les buts collectifs  $\gamma_b = \text{monter le bureau}$  et  $\gamma_e = \text{installer l'équipement informatique}$  et le but personnel de  $\gamma_m = \text{formaliser le modèle}$  :  $\Gamma_{a,team} = \{\gamma_b, \gamma_e\}$ , et  $\in \Gamma_{a,self} = \{\gamma_m\}$ .

- Les dimensions des agents correspondent aux dimensions du modèle de confiance MDS : chaque agent  $x$  a une valeur d'intégrité  $i_x \in [0; 1]$ , des valeurs de bienveillance  $b_{x,y_i} \in [0; 1]$  envers chaque autre agent  $y_i$  de l'environnement, et des niveaux de compétence  $a_{x,\kappa} \in [0; 1]$  ( $a$  pour *ability*) pour chaque compétence  $\kappa$  pouvant être spécifiée dans la description en ACTIVITY-DL de l'activité. Par exemple, Anna peut être intègre ( $i_a = 0,75$ ), bienveillante envers Béa et Cara ( $b_{a,b} = b_{a,c} = 0,75$ ) et avoir un niveau de compétence de 0,75 pour l'informatique ( $a_{a,INFO} = 0,75$ ).

- Enfin, chaque agent  $x$  a des croyances sur les autres agents  $y_i$  de l'environnement. En particulier, chaque agent  $x$  a des croyances notées  $\Gamma_{y_i,self}^x$  et  $\Gamma_{y_i,team}^x$  sur les ensembles des buts personnels et collectifs de  $y_i$ . L'agent  $x$  a également des croyances sur l'intégrité de  $y_i$ , la bienveillance de  $y_i$  envers chaque autre agent  $y_j$  et les niveaux de compétence de  $y_i$ , notées respectivement  $i_{y_i}^x \in [0; 1]$ ,  $b_{y_i,y_j}^x \in [0; 1]$  et  $a_{y_i,\kappa}^x \in [0; 1]$ . Les croyances d'un agent  $x$  à propos d'un agent  $y_i$  représentent sa relation de confiance avec ce dernier. On parlera de confiance *bien placée* lorsque les croyances de  $x$  sur  $y_i$  correspondent à la réalité, et de confiance *mal placée* dans le cas contraire. Par exemple, Anna peut penser que Béa a les mêmes buts collectifs qu'elle ( $\Gamma_{b,team}^a = \{\gamma_b, \gamma_e\}$ ); a le but personnel  $\gamma_p$  d'écrire un papier ( $\Gamma_{b,self}^a = \{\gamma_p\}$ ); n'est pas très intègre ( $i_b^a = 0,25$ ); est bienveillante envers elle-même et envers Cara ( $b_{b,a}^a = b_{b,c}^a = 0,75$ ); a un niveau de compétence de 0,5 pour l'informatique ( $a_{b,INFO}^a = 0,5$ ).

Ces éléments seront utilisés tout au long de cette section pour illustrer le fonctionnement de notre moteur décisionnel. Nous donnons dans le tableau 1 les caracté-

ristiques de des agents tels qu'utilisés dans cet exemple. Par ailleurs et par soucis de simplicité, tous nos agents ont ici une confiance bien placée en les autres.

Tableau 1. Caractéristiques de agents

		Anna	Béa	Cara
Buts	Collectifs	$\gamma_b, \gamma_e$	$\gamma_b, \gamma_e$	$\gamma_b, \gamma_e$
	Personnels	$\gamma_m$	$\gamma_p$	$\emptyset$
Intégrité		$i_a = 0,75$	$i_b = 0,25$	$i_c = 0,5$
Bienveil.	Anna	—	$b_{b,a} = 0,75$	$b_{c,a} = 0,75$
	Béa	$b_{a,b} = 0,75$	—	$b_{c,b} = 0,75$
	Cara	$b_{a,c} = 0,75$	$b_{b,c} = 0,75$	—
Compétence Info.		$a_{a,INFO} = 0,75$	$a_{b,INFO} = 0,5$	$a_{c,INFO} = 0,5$

Nous considérons ici que tous les agents ont une connaissance exacte de l'activité à réaliser (i.e. tous les agents raisonnent sur les mêmes arbres de tâches ACTIVITY-DL et partagés entre tous les agents); et, les agents évoluant dans le même environnement, nous considérons que chaque agent a connaissance de ce que les autres ont fait et sont en train de faire. Dans ce cadre, nous proposons un moteur décisionnel pour ces agents, de manière à ce que ceux-ci puissent sélectionner une tâche à faire tout en tenant compte de leurs coéquipiers. Les étapes de raisonnement proposées dans notre moteur décisionnel sont les suivantes :

(1) Chaque agent ayant plusieurs buts, la première question à laquelle l'agent doit répondre est la suivante : *Quel but a le plus d'importance pour moi ?* Nous définissons pour répondre à cette question un mécanisme de calcul de l'importance des buts. La valeur d'importance ainsi calculée représente la motivation de l'agent à atteindre chacun des buts, et permettra ainsi d'exprimer un ordre de préférence de l'agent sur les buts à réaliser. Dans un environnement collaboratif, chaque agent est influencé par les autres, et c'est notamment par le calcul de l'importance des buts que nous modélisons cette influence. Cette étape est présentée en 5.2.

(2) Un ordre de préférence sur les buts ayant été établi, chaque agent doit répondre à la question *Quelle-tâche ai-je le plus d'intérêt à faire ?* Répondre à cette question est complexe car l'importance du but que la tâche permettrait de réaliser n'est pas le seul paramètre à prendre en compte : la compétence de l'agent est aussi déterminante. Par ailleurs, si l'agent  $x$  accorde beaucoup d'importance au but  $\gamma_1$  et un peu moins au but  $\gamma_2$ , mais pense que  $y$  accorde énormément d'importance à  $\gamma_1$ , n'a-t-il pas intérêt à laisser  $y$  réaliser  $\gamma_1$  pour se consacrer à la réalisation de  $\gamma_2$  ? Pour répondre à ces questions, nous proposons que chaque agent calcule l'utilité de chaque tâche pour lui-même et pour chaque autre agent, ce qui lui permettra d'imaginer des répartitions de tâches entre tous les agents, et ainsi sélectionner une tâche à faire en tenant compte de ce que pourraient faire les autres. Cette étape est détaillée en 5.3.

## 5.2. Importance des buts

Le calcul de l'importance des buts par un agent  $x$  se fait en plusieurs étapes car  $x$  est influencé par les autres agents : après avoir calculé une valeur d'importance initiale des buts,  $x$  imagine quelle est l'importance initiale des buts pour les autres, ce qui a un impact sur lui. L'agent  $x$  calcule alors une valeur d'importance finale des buts. Finalement, l'agent  $x$  fait de même pour les autres agents. Nous détaillons ici ces différentes étapes.

### 5.2.1. Importance initiale des buts

L'intégrité d'un agent étant définie dans le modèle de confiance de MDS comme l'honnêteté de l'agent en question et plus particulièrement comme le respect de ses engagements par cet agent, nous considérons, comme exprimé dans les formules 10 et 11, qu'un agent intègre accordera plus d'importance aux buts collectifs sur lesquels il s'est engagé qu'à ses buts individuels, ce qui ne sera pas le cas d'un agent non intègre. La formule 10 permet à  $x$  de calculer  $iImp_x(\gamma) \in [0; 1]$ , l'importance qu'il accorde initialement au but  $\gamma$ , alors que la formule 11 permet à  $x$  de calculer  $iImp_{y_i}^x(\gamma) \in [0; 1]$ , l'importance que  $y_i$  accorde initialement à chacun de ses buts  $\gamma$  selon  $x$  (i.e.  $x$  raisonne alors sur ses croyances sur les buts et l'intégrité de  $y_i$ ).

$$\begin{cases} iImp_x(\gamma) = i_x & \text{si } \gamma \in \Gamma_{x,team} \\ iImp_x(\gamma) = 1 - i_x & \text{si } \gamma \in \Gamma_{x,self} \end{cases} \quad (10)$$

$$\begin{cases} iImp_{y_i}^x(\gamma) = i_{y_i}^x & \text{si } \gamma \in \Gamma_{y_i,team}^x \\ iImp_{y_i}^x(\gamma) = 1 - i_{y_i}^x & \text{si } \gamma \in \Gamma_{y_i,self}^x \end{cases} \quad (11)$$

Par exemple, Anna étant intègre, elle calculera  $iImp_a(\gamma_b) = iImp_a(\gamma_e) = 0,75$  et  $iImp_a(\gamma_m) = 0,25$ . Anna pensant que Béa n'est pas intègre, elle calculera  $iImp_b^a(\gamma_b) = iImp_b^a(\gamma_e) = 0,25$  et  $iImp_b^a(\gamma_p) = 0,75$ .

### 5.2.2. Influence des autres et importance finale des buts

Les agents formant une équipe, chacun est influencé par les autres. L'influence de l'agent  $y_i$  sur l'importance accordée par  $x$  à ses buts est représentée par la valeur  $\delta_{x,y_i} \in [0; 1]$ , donnée par la formule 12 :

– Si  $x$  pense que  $y_i$  accorde moins d'importance que lui-même à l'un des buts (i.e.  $iImp_x(\gamma) > iImp_{y_i}^x(\gamma)$ ), alors l'importance que  $x$  accorde à ce but sera diminuée proportionnellement à la différence entre  $iImp_x(\gamma)$  et  $iImp_{y_i}^x(\gamma)$ . En effet, l'importance accordée à un but par un agent peut être vue comme la motivation de l'agent à réaliser ce but ; or (Kerr, 1983) a mis en évidence que la motivation d'un individu diminue lorsqu'il pense que ses coéquipiers sont moins motivés. Par exemple, Anna pense que Béa accorde moins d'importance qu'elle au but  $\gamma_b$  ( $iImp_a(\gamma_b) > iImp_b^a(\gamma_b)$ ). L'influence de Béa sur l'importance accordée par Anna à ce but sera donc négative :  $\delta_{a,b}(\gamma_b) = 0,5 \times (0,25 - 0,75) = -0,25$ .

– Si  $x$  pense que  $y_i$  accorde plus d'importance que lui-même à l'un des buts (i.e.  $iImp_x(\gamma) \leq iImp_{y_i}^x(\gamma)$ ), alors, (i) Si  $x$  est bienveillant envers  $y_i$  (i.e.  $b_{x,y_i} > 0,5$ ),

$x$  aura à cœur les intérêts de  $y_i$  selon le modèle de confiance MDS ; l'importance que  $x$  accorde à ce but augmentera proportionnellement à la bienveillance de  $x$  envers  $y_i$ . Si  $x$  n'accorderait aucune importance à ce but (i.e.  $\gamma \notin \Gamma_{x,team}$  et  $\gamma \notin \Gamma_{x,self}$ ), alors  $x$  pourra adopter de manière altruiste le but de  $y_i$  (Castelfranchi, Falcone, 2010). (ii) si  $x$  n'est pas bienveillant envers  $y_i$  (i.e.  $b_{x,y_i} \leq 0,5$ ), il ne souhaitera pas aider ce dernier et l'importance que  $x$  accorde aux buts de  $y_i$  ne sera pas modifiée. Dans notre exemple, Anna est bienveillante envers Béa ; on a donc  $\delta_{a,b}(\gamma_p) = \max(0 ; 0,75 - 0,5) \times (0,25 - 0) = 0,125$ . Anna va donc adopter de manière altruiste le but de Béa ; autrement dit Anna pourra aider Béa à réaliser son but personnel.

Notons que  $x$  n'est pas influencé par les autres pour l'évaluation de l'importance de ses buts personnels. En effet, pour  $\gamma \in \Gamma_{x,self}$ , aucune valeur  $\delta_{x,y_i}$  n'est calculée ; il serait incohérent qu'un agent diminue l'importance qu'il accorde à un but personnel justement parce que ce but n'est pas partagé par d'autres.

$$\begin{cases} \delta_{x,y_i}(\gamma) = 0,5(iImp_{y_i}^x(\gamma) - iImp_x(\gamma)) & \text{si } iImp_x(\gamma) > iImp_{y_i}^x(\gamma) \\ \delta_{x,y_i}(\gamma) = \max(0 ; b_{x,y_i} - 0,5) \\ \quad \times (iImp_{y_i}^x(\gamma) - iImp_x(\gamma)) & \text{si } iImp_x(\gamma) \leq iImp_{y_i}^x(\gamma) \end{cases} \quad (12)$$

$$\Delta_x(\gamma) = \frac{\sum_{i=1}^{n(\gamma)} \delta_{x,y_i}(\gamma)}{n(\gamma)} \quad (13)$$

$$\begin{cases} fImp_x(\gamma) = \max(0 ; iImp_x(\gamma) + \Delta_x(\gamma)) & \text{si } \Delta_x(\gamma) \leq 0 \\ fImp_x(\gamma) = \min(1 ; iImp_x(\gamma) + \Delta_x(\gamma)) & \text{si } \Delta_x(\gamma) > 0 \\ fImp_x(\gamma) = iImp_x(\gamma) & \text{sinon} \end{cases} \quad (14)$$

L'ensemble des agents ayant une influence sur  $x$ , ce dernier calcule pour chaque but  $\gamma$  une moyenne  $\Delta_x(\gamma) \in [0;1]$  des influences de chaque agent  $y_i$  sur lui comme exprimé dans la formule 13, où  $n(\gamma)$  représente le nombre d'agents  $y_i$  ayant  $\gamma$  pour but. Finalement, l'agent  $x$  calcule, à partir de  $iImp_x(\gamma)$  et de  $\Delta_x(\gamma)$ , une valeur  $fImp_x(\gamma) \in [0;1]$  correspondant à l'importance finale qu'il accorde au but  $\gamma$  (formule 14).

Par exemple ici Cara influence également l'importance accordée par Anna à  $\gamma_b$  : nous aurons  $\delta_{a,c}(\gamma_b) = -0,125$  car Anna pense que Cara est moyennement intègre ( $i_c^a = 0,5$ ). On aura alors  $\Delta_a(\gamma_b) = (-0,25 - 0,125)/2 \simeq -0,188$  ; et l'importance finale accordée par Anna à  $\gamma_b$  sera  $fImp_a(\gamma_b) \simeq 0,75 - 0,188 \simeq 0,562$ . Par ailleurs Anna adopte le but personnel de Béa  $\gamma_m$ , puisque  $\Delta_a(\gamma_m) = 0,125$ , et donc  $fImp_a(\gamma_m) = 0,125$ .

Enfin, l'agent  $x$  considère que, tout comme lui, chaque agent  $y_i$  est influencé par chaque autre agent  $y_j$  de l'environnement. En appliquant les mêmes processus que pour lui-même :

– L'agent  $x$  calcule l'influence individuelle de chaque agent  $y_j$  sur l'importance accordée par  $y_i$  au but  $\gamma$ , représentée par la valeur  $\delta_{y_i,y_j}^x(\gamma) \in [0;1]$  (formule 15). Par exemple, Anna pense qu'elle-même est Cara ont une influence sur l'importance

accordée par Béa à  $\gamma_b$  :  $\delta_{b,a}^a(\gamma_b) = 0,25 \times (0,75 - 0,25) = 0,125$  et  $\delta_{b,c}^a(\gamma_b) = 0,25 \times (0,5 - 0,25) \simeq 0,062$ .

– Puis  $x$  calcule l'influence combinée de tous les agents  $y_j$  ayant le but  $\gamma$ , représentée par la valeur  $\Delta_{y_i,y_j}^x(\gamma) \in [0; 1]$  (formule 16). Ici, Anna calcule  $\Delta_b^a(\gamma_b) \simeq (0,125 + 0,062)/2 \simeq 0,094$ .

– Enfin,  $x$  calcule l'importance finale  $fImp_{y_i}^x(\gamma)$  accordée par  $y_i$  au but  $\gamma$  (formule 17) selon  $x$ . Ici, Anna obtient  $fImp_b^a(\gamma_b) \simeq 0,25 + 0,094 \simeq 0,344$ .

$$\begin{cases} \delta_{y_i,y_j}^x(\gamma) = 0,5(iImp_{y_j}^x(\gamma) - iImp_{y_i}^x(\gamma)) & \text{si } iImp_{y_i}^x(\gamma) > iImp_{y_j}^x(\gamma) \\ \delta_{y_i,y_j}^x(\gamma) = \max(0; b_{y_i,y_j}^x - 0,5) & \text{si } iImp_{y_i}^x(\gamma) \leq iImp_{y_j}^x(\gamma) \\ \quad \times (iImp_{y_j}^x(\gamma) - iImp_{y_i}^x(\gamma)) & \end{cases} \quad (15)$$

$$\Delta_{y_i}^x(\gamma) = \frac{\sum_{i=1}^{n(\gamma)} \delta_{y_i,y_j}^x(\gamma)}{n(\gamma)} \quad (16)$$

$$\begin{cases} fImp_{y_i}^x(\gamma) = \max(0; iImp_{y_i}^x(\gamma) + \Delta_{y_i}^x(\gamma)) & \text{si } \Delta_{y_i}^x(\gamma) \leq 0 \\ fImp_{y_i}^x(\gamma) = \min(1; iImp_{y_i}^x(\gamma) + \Delta_{y_i}^x(\gamma)) & \text{si } \Delta_{y_i}^x(\gamma) > 0 \\ fImp_{y_i}^x(\gamma) = iImp_{y_i}^x(\gamma) & \text{sinon} \end{cases} \quad (17)$$

### 5.3. Utilité et répartitions des tâches

Une fois l'importance des buts calculée pour lui-même et pour les autres, l'agent  $x$  doit chercher comment réaliser les buts les plus importants pour lui. L'agent  $x$  va pour cela calculer l'utilité des tâches, puis réfléchir à différentes répartitions possibles des tâches entre les agents afin de faire un choix informé quant à la tâche qu'il sélectionnera. Nous développons ces mécanismes dans les sections suivantes.

#### 5.3.1. Utilité des tâches

L'agent  $x$  raisonne tout d'abord sur les contraintes en termes de nombre minimal d'intervenants et de compétences associées à chaque tâche (voir 4.3 et 4.4) : si l'une de ces contraintes n'est pas vérifiée, l'utilité de la tâche est nulle. Si ces contraintes sont vérifiées, alors l'agent  $x$  calcule l'utilité  $U_x(\tau_k) \in [0; 1]$  de la tâche  $\tau_k$  comme décrit dans la formule 18 : l'utilité d'une tâche est proportionnelle à la fois à l'importance du but  $\gamma$  auquel elle contribue (l'utilité de  $\tau_k$  est nulle si elle ne contribue à aucun but) et à la compétence de l'agent pour réaliser cette tâche. La compétence de l'agent  $x$  pour réaliser la tâche est notée  $a_{x,\kappa} \in [0; 1]$  et pour la calculer, l'agent  $x$  raisonne sur ses niveaux de compétence  $a_{x,\kappa}$  (voir 5.1 – on considère par défaut  $a_{x,\text{NONE}} = 0,5$ ) et sur les informations en termes de compétences  $c(\tau_k, t)$  attachées à chaque tâche (voir 4.6). Par exemple ici, Anna n'a pas la compétence PERMIS spécifiée dans les informations en termes de compétences attachées à  $\tau_1$ ;  $\tau_1$  contribuant au but  $\gamma_b = \text{monter le}$

*bureau*. On a donc  $a_a(\tau_1) = a_{a,\text{NONE}} = 0,5$ , ce qui donne  $U_a(\tau_1) \simeq 0,562 \times 0,5 \simeq 0,281$ .

$$U_x(\tau_k) = fImp_x(\gamma) \times a_x(\tau_k) \quad \text{avec} \quad a_x(\tau_k) = \max_{\forall \kappa \text{ de } c(\tau_k, t)} (a_{x, \kappa}) \quad (18)$$

$$U_{y_i}^x(\tau_k) = fImp_{y_i}^x(\gamma) \times a_{y_i}^x(\tau_k) \quad \text{avec} \quad a_{y_i}^x(\tau_k) = \max_{\forall \kappa \text{ de } c(\tau_k, t)} (a_{y_i, \kappa}^x) \quad (19)$$

Encore une fois, l'agent  $x$  calcule ensuite, pour toutes les tâches  $\tau_k$ , la valeur  $U_{y_i}^x(\tau_k) \in [0; 1]$  : ce qu'il pense être l'utilité de  $\tau_k$  pour chaque autre agent  $y_i$  en appliquant les mêmes processus que pour lui-même (formule 19 – par défaut,  $a_{y_i, \text{NONE}}^x = 0,5$ );  $x$  raisonne alors sur ses croyances sur l'importance finale que  $y_i$  accorde au but  $\gamma$  auquel  $\tau_k$  contribue et sur ses croyances sur les compétences de  $y_i$ . Dans notre exemple, Anna pense que Béa n'a pas non plus la compétence PERMIS. Elle calcule donc  $a_b^a(\tau_1) = 0,5$  et  $U_b^a(\tau_1) \simeq 0,344 \times 0,5 \simeq 0,172$ .

### 5.3.2. Répartition des tâches

Avoir calculé les utilités des tâches pour chaque agent permet à  $x$  de générer des répartitions possibles de tâches ; ainsi  $x$  pourra sélectionner une tâche en tenant compte non seulement de son intérêt propre (i.e. utilité des tâches pour lui), mais aussi en fonction de ce qu'il pense que les autres pourraient faire (i.e. utilité des tâches pour les autres selon  $x$ ).

L'agent  $x$  doit donc imaginer une répartition des tâches qui soit globalement la plus satisfaisante possible pour l'ensemble des agents (i.e. l'utilité globale de la répartition – somme des utilités de chaque tâche selon  $x$ , pour chaque agent assigné à cette tâche – doit être maximisée). Il s'agit d'un problème classique d'affectation, qui peut être résolu en temps polynomial par l'application de la méthode hongroise (Kuhn, 1955). Cette méthode permet, pour un ensemble de  $n$  tâches pouvant chacune être effectuée par *une* personne et pour un ensemble de  $m$  personnes, chaque personne ayant une utilité différente pour chaque tâche, de proposer une répartition des tâches entre les personnes maximisant l'utilité globale. Dans notre cas, certaines tâches  $\tau_k$  pouvant être effectuées à plusieurs, nous les "dédoublons" donc  $n_{max}(\tau_k, t)$  fois avant d'appliquer la méthode hongroise. Plusieurs agents pourront ainsi être assignés à une même tâche dans les répartitions générées. Notons qu'il est possible que, dans la répartition maximisant l'utilité globale,  $n(\tau_k)$  agents soient assignés à  $\tau_k$  avec  $n(\tau_k) < n_{min}(\tau_k)$ , auquel cas  $\tau_k$  ne pourra pas être réalisée. Néanmoins nous n'éliminons pas ce type de solution de manière à éviter les situations de blocage (e.g. trois agents doivent réaliser deux tâches  $\tau_1$  et  $\tau_2$  qui nécessitent chacune deux intervenants minimum : il ne faut pas empêcher les répartitions telles que deux agents sont assignés à  $\tau_1$  et un seul à  $\tau_2$  – ou inversement).

Dans le cas où l'application de la méthode hongroise retourne plusieurs répartitions, nous proposons que l'agent  $x$  choisisse la répartition pour laquelle l'utilité de la tâche à laquelle il est assigné est maximisée. En effet, ce "gain" d'utilité est assuré si  $x$  fait la tâche ; mais  $x$  n'ayant aucun contrôle sur le comportement des autres, il n'a pas

d'intérêt à sélectionner une répartition telle que l'utilité de  $y$  selon lui soit maximisée. Si plusieurs répartitions répondent à ce critère, l'agent  $x$  choisira une répartition au hasard. Finalement, l'agent  $x$  sélectionne la tâche à laquelle il est assigné dans la répartition choisie. Par ailleurs, l'agent  $x$  ne décide pas de la tâche que chaque autre agent réalisera : il anticipe leurs comportements pour faire un choix éclairé. Dans le cas où les croyances de l'agent  $x$  sur les autres agents correspondent à la réalité (confiance bien placée), alors l'anticipation par  $x$  des comportements des autres sera juste ; i.e. chaque autre agent  $y_i$  devrait sélectionner la tâche à laquelle il était assigné dans la répartition imaginée par  $x$ , ce qui ne sera pas le cas si les croyances de  $x$  sur les autres sont fausses (confiance mal placée).

Reprenons notre exemple : si l'on considère les utilités de chaque tâche pour chaque agent calculée par Anna telles que présentées dans le tableau 2, alors la méthode hongroise retournera deux répartitions possibles de tâches : (1) Anna et Cara installent l'équipement informatique et Béa écrit un papier ou (2) Anna installe l'équipement informatique, Béa écrit un papier et Cara monte le bureau. Anna étant assignée à la même tâche dans ces deux répartitions, l'une d'entre elles sera choisie au hasard ; supposons ce que soit la (1) : Anna sélectionnera la tâche  $\tau_2 = \text{installer l'équipement informatique}$ .

Tableau 2. Utilités calculées par Anna pour toutes les tâches et tous les agents

	$\tau_1$ monter bureau	$\tau_2$ installer info.	$\tau_3$ formaliser modèle	$\tau_4$ écrire papier
Anna	$U_a(\tau_1) \simeq 0,281$	$U_a(\tau_2) \simeq 0,422$	$U_a(\tau_3) = 0,125$	$U_a(\tau_4) \simeq 0,094$
Béa	$U_b^a(\tau_1) \simeq 0,172$	$U_b^a(\tau_2) \simeq 0,172$	$U_b^a(\tau_3) \simeq 0,031$	$U_b^a(\tau_4) = 0,375$
Cara	$U_c^a(\tau_1) \simeq 0,234$	$U_c^a(\tau_2) \simeq 0,234$	$U_c^a(\tau_3) \simeq 0,031$	$U_c^a(\tau_4) \simeq 0,094$

#### 5.4. Processus récursif

Les agents raisonnant en priorité sur des tâches abstraites, la tâche sélectionnée ne correspond pas nécessairement à une action. Dans ce cas, les agents appliquent récursivement le processus de sélection d'une tâche en raisonnant sur les sous-tâches de la tâche sélectionnée, et ce jusqu'à sélectionner une tâche correspondant à une action. Toutefois lors des appels récursifs successifs du processus de sélection de tâche, l'agent  $x$  raisonne uniquement sur le sous-ensemble des agents pour lesquels il anticipe qu'ils sélectionneront la même tâche que lui. En effet, imaginons que  $\tau_2$  soit une tâche composée avec une contrainte ET/IND de deux sous-tâches  $\tau_{21} = \text{installer ordinateur}$  et  $\tau_{22} = \text{installer imprimante}$ . Anna doit maintenant décider si elle fait  $\tau_{21}$  ou  $\tau_{22}$ . Pour prendre cette décision, Anna tiendra uniquement compte de Cara puisqu'elle pense que Béa fera  $\tau_3$ .

Ici, les croyances d'Anna correspondant à la réalité, Béa fera effectivement la tâche  $\tau_3$ , alors qu'Anna et Cara réaliseront  $\tau_{21}$  et  $\tau_{22}$ .

### 5.5. Confiance et performance de groupe

Le moteur décisionnel proposé dans cette section permet à chaque agent de prendre une décision à partir de ses prédictions sur les comportements des autres, ces prédictions étant générées à partir de ses croyances sur les autres. Des croyances fausses d'un agent sur un autre entraînent des mauvaises prédictions de la part du premier sur le comportement du second et nous permettent ainsi de générer des comportements d'équipe non optimaux (voir exemple section 6).

Bien que notre objectif principal ne soit pas la performance de l'équipe mais la vraisemblance des comportements générés, il est souhaitable que, dans des conditions *idéales*, la répartition émergente des tâches entre les agents soit un maximum efficace. En effet, lorsque tous les agents sont intègres, bienveillants, et que leurs croyances sur les autres sont justes (conditions idéales), la répartition des tâches entre les agents devrait être idéale : chaque agent devrait faire la tâche pour laquelle il est le plus compétent. Pour vérifier que cela est bien le cas, nous avons testé la performance d'une équipe de trois agents devant se répartir quatre tâches. Nous avons défini la mesure de performance d'équipe suivante : la performance de l'équipe est initialement nulle et, chaque fois qu'une tâche nécessitant la compétence  $\kappa$  est réalisée par un agent  $x$ , la performance de l'équipe est augmentée proportionnellement à  $a_{x,\kappa}$  (i.e. le niveau de compétence de  $x$  pour  $\kappa$ ). Nous avons étudié les quatre cas suivants : (1) une seule tâche requiert une compétence, (2) deux tâches requièrent deux compétences différentes, (3) trois tâches requièrent trois compétences différentes et (4) quatre tâches requièrent quatre compétences différentes. Le niveau de compétence des agents est tiré au hasard dans l'ensemble  $\{0 ; 0,25 ; 0,5 ; 0,75 ; 1\}$ . Nous avons alors comparé, sur 500 simulations pour chacun des quatre cas, la performance de l'équipe dans deux conditions : (i) les agents utilisent notre moteur décisionnel et (ii) les agents ne tiennent pas compte de leurs croyances sur les autres dans leur raisonnement mais sélectionnent simplement la tâche ayant la plus forte utilité pour eux (i.e. celle pour laquelle ils sont le plus compétent).

Tableau 3. Comparaison de la performance de l'équipe entre les conditions (i) et (ii)

	Meilleure	Egale	Moins bonne
Cas 1	49,4 %	50,6 %	0,0 %
Cas 2	54,8 %	45,0 %	0,2 %
Cas 3	58,2 %	40,6 %	1,2 %
Cas 4	49,0 %	49,6 %	1,4 %

Le tableau 3 présente les résultats obtenus en termes de comparaison de la performance de l'équipe entre les deux conditions et pour chaque cas. On observe que la performance des agents avec notre moteur décisionnel (condition (i)) est dans la grande majorité des simulations meilleure ou identique à la performance des agents ne tenant pas compte des autres (condition (ii)) : notre proposition est donc efficace en termes d'affectation émergente des tâches. Néanmoins, on observe que, dans un faible pourcentage des simulations, la performance des agents est moins bonne dans la condition (i) que dans la condition (ii) : cela s'explique par le fait que dans des

conditions idéales, notre moteur décisionnel permet aux agents de sélectionner une tâche de manière à ce qu'à chaque instant, la performance de l'équipe soit maximisée. Par exemple, si le niveau de compétence de l'agent  $x$  est plus élevé que celui de l'agent  $y$  pour  $\kappa_1$  et  $\kappa_2$ , nécessaires respectivement pour réaliser  $\tau_1$  et  $\tau_2$ , mais si l'agent  $y$  atteint son niveau maximal de compétence pour  $\kappa_1$ , alors les agents s'organiseront de manière à ce que  $y$  fasse  $\tau_1$  et  $x$  fasse  $\tau_2$ . Pourtant, il aurait pu être plus avantageux que  $y$  fasse  $\tau_3$  et laisse  $x$  faire  $\tau_1$  et  $\tau_2$ . Autrement dit, maximiser la performance de l'équipe à  $t_1$ , puis à  $t_2$ , puis à  $t_3$ , etc. n'est pas équivalent à maximiser la performance de l'équipe sur l'ensemble de la simulation. Maximiser la performance sur l'ensemble de la simulation nécessiterait d'intégrer des mécanismes de planification. Si cela pourrait être une perspective intéressante, il faut néanmoins souligner que l'intégration de tels mécanismes entraînerait ici une explosion combinatoire et rendrait impossible l'utilisation de notre modèle dans une simulation en temps réel.

### 5.6. Conclusion sur le moteur décisionnel

Afin de permettre à des agents de tenir compte de leurs relations de confiance avec les autres lors de la sélection d'une tâche, nous avons proposé un modèle d'agent s'appuyant le modèle de confiance MDS. Dans ce modèle, la dimension de la bienveillance est particulièrement intéressante puisqu'elle permettra à nos agents d'adopter des comportements d'aide. Nous avons également proposé un moteur décisionnel qui permet à nos agents de calculer l'importance qu'ils accordent à leurs buts puis l'utilité de chacune des tâches afin de sélectionner la tâche correspondant selon eux à la meilleure répartition possible des tâches. Ces mécanismes permettent notamment de reproduire les effets suivants :

(1) *Adoption du but d'un autre agent.* Dans notre exemple, Anna adopte le but personnel de Béa. C'est grâce à ce mécanisme que l'on pourra générer des comportements d'aide (e.g. après avoir terminé d'installer l'équipement informatique, Anna aide Béa à rédiger un papier).

(2) *Augmentation ou diminution de la motivation.* La motivation des agents pour réaliser les buts collectifs est influencée par ce qu'ils pensent de la motivation des autres.

(3) *Tenir compte des autres pour sélectionner une tâche.* De manière générale, les agents tiennent compte de chacun des autres pour sélectionner une tâche. Par exemple, si Anna pense que Cara sera plus compétente pour  $\tau_{21} = \text{installer ordinateur}$  que pour  $\tau_{22} = \text{installer imprimante}$ , alors elle choisira de faire  $\tau_{22}$  de manière à laisser Cara faire  $\tau_{21}$ .

(4) *Confiance mal placée.* Les cas de confiance mal placée (i.e. les croyances d'un agent sur un autre ne correspondent pas à la réalité) permettent de générer des comportements non idéaux. Un exemple de scénario produit à partir d'un cas de confiance mal placée sera présenté section 6.

(5) *Raisonnement sur des arbres de tâches* ACTIVITY-DL. Les agents utilisent en particulier (i) les contraintes et (ii) informations générées par notre système (voir 4) :

(i) les agents ne tiendront pas compte des tâches qui ne sont pas réalisables par l'équipe ; et (ii) les agents utilisent les informations en termes de compétences associées à chaque tâche pour en calculer l'utilité. Les informations sur le nombre maximal de personnes sont quant à elles utilisées pour la génération des répartitions de tâches.

Les éléments 1, 2, 3 et 5 permettent notamment l'émergence de comportements humains collaboratifs sans communication entre les agents, alors que l'élément 4 permet en particulier de générer des comportements non-optimaux au niveau de l'équipe mais cohérents au niveau des individus. Notre proposition répond donc à nos objectifs. Nous présentons dans la section suivante une évaluation perceptive préliminaire de notre système.

## 6. Évaluation préliminaire

Nous avons conduit deux évaluations perceptives préliminaires de notre système et plus particulièrement du moteur décisionnel proposé. Nous nous sommes concentrés sur deux éléments : l'intégrité des agents et la confiance en les compétences.

### 6.1. Intégrité des agents et efficacité perçue

La première évaluation que nous avons conduite porte sur l'influence de l'intégrité des agents sur leurs comportements. Nous avons fait les deux hypothèses suivantes : **(H1)** Une équipe dont les trois agents sont intègres sera perçue par les participants comme plus efficace qu'une équipe dont seulement deux agents sont intègres, qui sera elle-même perçue comme plus efficace qu'une équipe dont aucun agent n'est intègre. **(H2)** Un agent intègre sera jugé par les participants comme contribuant plus au bon fonctionnement de l'équipe qu'un agent non intègre.

Pour tester ces hypothèses nous avons demandé aux participants d'observer les comportements de nos agents sur un scénario de rangement de bureau : trois agents  $x$ ,  $y$  et  $z$  ont le but collectif de *ranger le bureau* (tâche racine), qui peut être réalisé en faisant les sous-tâches *accrocher le tableau*, *laver le sol*, *monter le bureau*, *cacher la machine à café* et *réparer l'ordinateur* (contrainte ET/IND). Les agents ont également chacun les buts personnels de *boire un café* et de *se détendre*. Aucune de ces tâches ne nécessite de compétence particulière, et toutes ces tâches se font seul, hormis *monter le bureau* qui se fait à deux. Les relations entre les agents sont neutres (i.e.  $\forall y_i, y_j, b_{y_i, y_j} = 0, 5$ ) et les croyances des agents sur les autres sont exactes. Nous avons généré trois variantes de ce scénario :

(v1) Les trois agents sont intègres (i.e.  $i_x = i_y = i_z = 1$ ) et participent tous à la réalisation du but collectif avant de boire un café et de se détendre.

(v2) L'agent  $x$  n'est pas intègre alors que les agents  $y$  et  $z$  le sont (i.e.  $i_x = 0$  et  $i_y = i_z = 1$ );  $y$  et  $z$  rangent le bureau pendant que  $x$  se détend et boit un café.

(v3) Aucun agent n'est intègre (i.e.  $i_x = i_y = i_z = 0$ ); les agents boivent un café et se détendent et le but collectif n'est pas réalisé.

Douze participants ont observé chacun trois vidéos de même durée, correspondant chacune à une variante de ce scénario. N'ayant pas pu coupler notre moteur décisionnel à un environnement virtuel, nous avons présenté les comportements des agents aux participants sous forme d'un graphe tracé dynamiquement, tel que dans la figure 3 : le temps est présenté sur l'axe des abscisses ; les lignes horizontales définissent des espaces qui correspondent chacun à une action. Les agents sont représentés par des points de couleur dont la position évolue en fonction du temps, et les traits de couleurs représentent l'historique des actions des agents. Par exemple, l'agent  $x$  (en rose) a d'abord caché la machine à café, puis a monté le bureau et bu un café, et est en train de se détendre. Après l'observation de chaque vidéo, les participants étaient invités à indiquer, sur une échelle de Likert à cinq niveaux (de 1 = *pas du tout d'accord* à 5 = *tout à fait d'accord*) à quel degré ils étaient d'accord avec deux affirmations portant respectivement sur l'efficacité de l'équipe ( $a1$  – *Les agents forment une équipe efficace*) et sur le comportement d'un agent particulier dans l'équipe ( $a2$  – *L'agent  $x$  contribue au bon fonctionnement de l'équipe*).

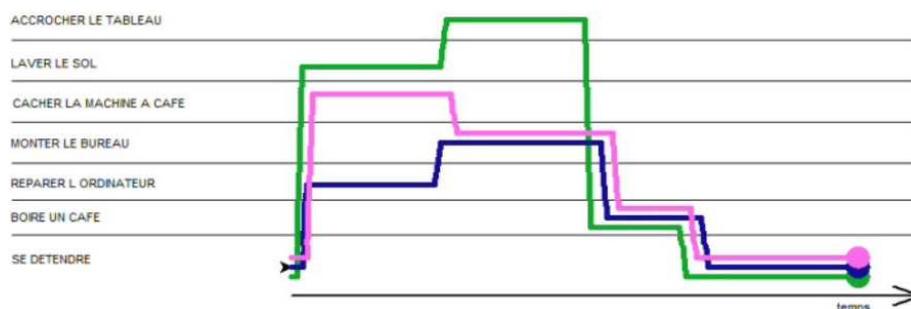


Figure 3. Dernière image de la vidéo présentant le comportement des agents dans la variante  $v1$  du scénario de rangement de bureau

Les résultats obtenus sont présentés dans le tableau 4 – à gauche. Concernant notre première hypothèse, l'équipe d'agents est perçue comme très efficace dans la variante  $v1$ , moyennement efficace dans la variante  $v2$  et pas du tout efficace dans la variante  $v3$ . Pour vérifier que les résultats obtenus sont significatifs, nous avons appliqué une ANOVA sur l'efficacité perçue de l'équipe ( $a1$ ), dont le résultat est significatif ( $F(2,33) = 62$ ,  $p < 0,05$ ). Pour vérifier que les participants distinguent bien une variation de l'efficacité de l'équipe dans chaque variante, nous avons appliqué trois comparaisons post-hoc paire par paire avec le test de Bonferroni, qui sont toutes significatives (voir tableau 4 – à droite). Notre hypothèse **H1** est donc supportée. En ce qui concerne notre seconde hypothèse, l'agent  $x$  est perçu comme contribuant fortement au bon fonctionnement de l'équipe dans la variante  $v1$ , ce qui n'est pas le cas dans les variantes  $v2$  et  $v3$ . Nous avons également effectué une ANOVA ( $F(2,33) = 349$ ,  $p < 0,05$ ) puis des tests de Bonferroni qui sont significatifs pour les paires de variantes  $v1$  et  $v2$ , et  $v1$  et  $v3$ , et non significatif pour la paire  $v2$  et  $v3$ . En effet, dans les variantes  $v2$  et  $v3$ , l'agent  $x$  a un comportement identique puisqu'il n'est pas intègre. L'hypothèse **H2** est donc également supportée.

Tableau 4. Scénario de rangement de bureau – à gauche : moyennes  $m$  et écarts types  $\sigma$  obtenus pour chacune des affirmations et chacune des variantes – à droite : valeurs  $t$  obtenues par les tests de Bonferroni pour chacune des affirmations et chacune des paires de variantes

	$a1$	$a2$		$a1$	$a2$
$v1$	$m = 4,92 ; \sigma = 0,29$	$m = 4,83 ; \sigma = 0,39$	$v1$ et $v2$	$t = 6,05$	$t = 23,36$
$v2$	$m = 3,17 ; \sigma = 1,19$	$m = 1,00 ; \sigma = 0,00$	$v1$ et $v3$	$t = 13,53$	$t = 22,34$
$v3$	$m = 1,00 ; \sigma = 0,00$	$m = 1,17 ; \sigma = 0,58$	$v2$ et $v3$	$t = 7,49$	$t = 1,02$

## 6.2. Confiance en les compétences et relation de confiance perçue

La seconde évaluation que nous avons conduite porte sur la perception des relations de confiance entre les agents. Nous avons fait l'hypothèse **H3** : les participants seront capables d'identifier le niveau de confiance d'un agent en un autre de par le comportement des agents. Pour tester cette hypothèse nous nous sommes appuyés sur un scénario de sauvetage de victime : trois agents doivent réaliser les tâches  $\tau_1 = \text{préparer l'ambulance}$ ,  $\tau_2 = \text{faire les premiers soins}$  et  $\tau_3 = \text{rassembler et transmettre les informations}$ . Chaque action peut se faire seul ou à deux, et chaque agent a des compétences différentes qui lui permettent de réaliser une ou plusieurs tâches : un médecin (agent  $m$ ) est compétent pour réaliser toutes les tâches ( $a_m(\tau_1) = a_m(\tau_2) = a_m(\tau_3) = 1$ ); un infirmier (agent  $i$ ) n'est compétent que pour la tâche  $\tau_2$  ( $a_i(\tau_1) = a_i(\tau_3) = 0$  et  $a_i(\tau_2) = 1$ ); enfin un ambulancier (agent  $a$ ) n'est compétent que pour la tâche  $\tau_1$  ( $a_a(\tau_2) = a_a(\tau_3) = 0$  et  $a_a(\tau_1) = 1$ ). Enfin, les relations entre les agents sont neutres. Nous avons généré deux variantes de ce scénario :

( $v1$ ) La confiance du médecin en les compétences de l'ambulancier et de l'infirmier est bien placée; i.e. ses croyances sur les compétences des autres sont exactes (en particulier,  $a_a^m(\tau_1) = a_i^m(\tau_2) = 1$  : le médecin considère l'ambulancier et l'infirmier comme compétents à juste titre). L'ambulancier prépare alors l'ambulance pendant que l'infirmier effectue les premiers soins et que le médecin rassemble et transmet les informations.

( $v2$ ) La confiance du médecin en les compétences de l'ambulancier et de l'infirmier est mal placée; i.e. ses croyances sur les compétences des autres sont fausses (en particulier,  $a_a^m(\tau_1) = a_i^m(\tau_2) = 0$  : le médecin considère à tort l'ambulancier et l'infirmier comme incompetents). Le médecin aide alors l'ambulancier à préparer l'ambulance puis aide l'infirmier à faire les premiers soins avant de rassembler et transmettre les informations.

Tout comme pour la première évaluation, douze participants ont observé deux vidéos présentant sous forme de graphe tracé dynamiquement le comportement des agents dans les variantes  $v1$  et  $v2$  respectivement. Les affirmations présentées aux sujets portaient sur la confiance du médecin en l'ambulancier ( $a3 - \text{Le médecin a confiance en l'ambulancier}$ ) et en l'infirmier ( $a4 - \text{Le médecin a confiance en l'infirmier}$ ). Les résultats obtenus sont présentés dans le tableau 5 – à gauche : le médecin est perçu comme ayant fortement confiance en les autres dans la variante  $v1$ , alors qu'il

est perçu comme ayant peu confiance en les autres dans la variante v2, ce qui correspond à notre hypothèse. Pour vérifier si ces résultats sont significatifs, nous avons appliqué des tests de Student sur la confiance perçue du médecin en l'ambulancier (a3) et sur la confiance perçue du médecin en l'infirmier (a4). Les résultats (voir tableau 5 – à droite) montrent que les différences d'appréciation entre les deux variantes sont significatives à  $p < 0,01$ . Notre hypothèse **H3** est donc vérifiée.

Tableau 5. Scénario de sauvetage de victime – à gauche : moyennes  $m$  et écarts types  $\sigma$  obtenus pour chacune des affirmations et chacune des variantes – à droite : valeurs  $p$  obtenues par les tests de Student pour la comparaison des deux variantes

	a3	a4		
v1	$m = 4,92 ; \sigma = 0,29$	$m = 4,92 ; \sigma = 0,29$	v1 et v2 – a3	$p < 0,0001$
v2	$m = 1,5 ; \sigma = 0,91$	$m = 3,08 ; \sigma = 1,73$	v1 et v2 – a4	$p = 0,004$

### 6.3. Discussion

Nous avons, par la première évaluation, pu vérifier que nous sommes capables de générer les comportements d'équipes de travail plus ou moins efficaces : cela répond à nos besoins, puisque nous pourrions ainsi confronter un apprenant à différents types d'équipe de manière à entraîner ses capacités d'adaptation. Néanmoins l'influence d'autres facteurs (e.g. bienveillance des agents, compétence des agents) sur l'efficacité de l'équipe mériterait d'être étudiée. Notre modèle s'appuyant sur des relations de confiance, nous avons souhaité, par la seconde évaluation, vérifier que celles-ci peuvent être perçues dans les comportements des agents. Toutefois nous ne nous sommes intéressés ici qu'à la confiance en les compétences, et il serait intéressant de conduire le même type d'évaluation pour vérifier la perception de la confiance d'un agent en l'intégrité et la bienveillance d'un autre. Enfin, nous avons pu vérifier par ces évaluations préliminaires que la méthode appliquée pour présenter aux sujets les comportements des agents (i.e. par un graphe tracé dynamiquement) fonctionne, et pourrait être utilisée pour une évaluation plus complète.

## 7. Conclusion et perspectives

Nous nous intéressons dans ces travaux à la génération des comportements de personnages virtuels autonomes pour des activités collectives. En particulier, nous cherchons à générer des comportements réalistes et représentatifs de ceux observés sur le terrain. Ainsi, tout comme dans les équipes de travail humaines, nous souhaitons avoir des agents qui font des erreurs, ou à l'inverse qui aident les autres même s'ils n'ont rien à y gagner. Nos travaux s'inscrivant dans le cadre plus large de la formation en environnement virtuel, les comportements générés doivent néanmoins être explicables et justifiables, de manière à ce que l'apprenant puisse les comprendre et les anticiper, mais aussi de manière que l'apprentissage en environnement virtuel soit transposable en environnement réel. A ce jour, la plupart des travaux s'intéressant à la génération de comportements d'équipe dans le domaine des EV et des SMA privilégient avant

tout la performance de l'équipe au détriment de la vraisemblance des comportements individuels.

Pour répondre à nos objectifs, nous avons proposé une augmentation du langage de description de l'activité ACTIVITY-DL de manière à supporter la description d'activités collectives, ainsi que des mécanismes de génération et de propagation de contraintes et d'informations qui facilitent le raisonnement des agents sur les arbres de tâches ACTIVITY-DL. Nous avons également proposé un modèle d'agent s'appuyant sur le modèle de confiance MDS ainsi qu'un moteur décisionnel qui permet aux agents de tenir compte des autres à la fois lors du calcul de l'importance qu'ils accordent à leurs buts et lors de la sélection d'une tâche. Enfin nous avons conduit des évaluations perceptives préliminaires de notre système dont les résultats sont encourageants.

Outre la perspective d'une évaluation plus complète de nos travaux, nous souhaitons par la suite nous concentrer sur l'enrichissement de notre proposition, et ce par l'ajout de deux éléments principaux. (1) La notion de degré de confiance manque à notre modèle : lorsqu'un agent pense qu'un autre n'est pas intègre, il est *sûr* de ce qu'il pense. Or dans la réalité nos opinions sont souvent plus nuancées et il serait pertinent d'intégrer cet élément à notre proposition. Nous pourrions ainsi associer à chaque croyance une valeur de certitude comme cela est fait par (Castelfranchi, Falcone, 2010) ; ou représenter les croyances des agents par des ensembles flous comme dans BDI+Repage. (2) Nous nous sommes jusqu'ici concentrés sur les processus qui permettent à un agent de prendre une décision quant à la tâche à réaliser ; cependant, pour compléter notre moteur décisionnel, nous devrions également proposer des mécanismes de mise à jour des croyances des agents. En effet, la non-mise à jour des croyances des agents pourrait entraîner des comportements incohérents voire irréalistes sur un scénario plus long que ceux présentés ici (e.g. un agent médecin qui continue indéfiniment à aider l'agent infirmier bien que celui-ci ait démontré sa compétence, et qui retarde ainsi l'exécution des tâches que lui seul peut effectuer, ralentissant donc l'équipe). Ces travaux futurs devraient permettre d'améliorer notre système et le réalisme des comportements générés.

#### Remerciements

*Ces travaux ont été menés dans le cadre d'une thèse financée par la Direction Générale de l'Armement (DGA) et le Labex MS2T, lui-même financé par le gouvernement français au travers du programme Investissements d'avenir géré par l'Agence Nationale de la Recherche (ANR – référence ANR-11-IDEX-0004-02).*

#### Bibliographie

- Anderson C., Franks N. R. (2003). Teamwork in animals, robots, and humans. *Advances in the Study of Behavior*, vol. 33, p. 1–48.
- Barot C. (2014). *Scénarisation d'environnements virtuels*. Thèse de doctorat – Université de technologie de Compiègne.

- Callebert L., Lourdeaux D., Barthès J.-P. (2016a). Moteur décisionnel reposant sur un modèle de confiance pour des agents autonomes. In *Rencontre des jeunes chercheurs en intelligence artificielle*.
- Callebert L., Lourdeaux D., Barthès J.-P. (2016b). A trust-based decision-making approach applied to agents in collaborative environments. In *International conference on agents and artificial intelligence*, p. 287–295.
- Cao Y. U., Fukunaga A. S., Kahng A. (1997). Cooperative mobile robotics: Antecedents and directions. *Autonomous robots*, vol. 4, n° 1, p. 7–27.
- Castaldo S. (2002). Meanings of trust: a meta-analysis of trust definitions. In *Proceedings of the european academy of management conference, stockholm, published online*.
- Castelfranchi C., Falcone R. (2010). *Trust theory: A socio-cognitive and computational model* (vol. 18). John Wiley & Sons.
- Chaudhuri A., Sopher B., Strand P. (2002). Cooperation in social dilemmas, trust and reciprocity. *Journal of Economic Psychology*, vol. 23, n° 2, p. 231–249.
- Cohen P. R., Levesque H. J. (1990). Intention is choice with commitment. *Artificial intelligence*, vol. 42, n° 2-3, p. 213–261.
- Erdem F. (2003). Optimal trust and teamwork: From groupthink to teamthink. *Work Study*, vol. 52, n° 5, p. 229–233.
- Gambetta D. (2000). Can we trust trust. *Trust: Making and breaking cooperative relations*, vol. 13, p. 213–237.
- Gerbaud S. (2008). *Contribution à la formation en réalité virtuelle*. Thèse de doctorat – INSA de Rennes.
- Giampapa J. A., Sycara K. (2002). *Team-oriented agent coordination in the RETSINA multi-agent system*. Rapport technique. Pittsburgh, PA, Robotics Institute.
- Gill H., Boies K., Finegan J. E., McNally J. (2005). Antecedents of trust: Establishing a boundary condition for the relation between propensity to trust and intention to trust. *Journal of business and psychology*, vol. 19, n° 3, p. 287–302.
- Grosz B. J., Kraus S. (1996). Collaborative plans for complex group action. *Artificial Intelligence*, vol. 86, n° 2, p. 269–357.
- Herzig A., Lorini E., Hübner J. F., Vercouter L. (2010). A logic of trust and reputation. *Logic Journal of IGPL*, vol. 18, n° 1, p. 214–244.
- Hill Jr R. W., Gratch J., Marsella S., Rickel J., Swartout W. R., Traum D. R. (2003). Virtual humans in the mission rehearsal exercise system. *KI*, vol. 17, n° 4, p. 5.
- Hübner J. F., Sichman J. S., Boissier O. (2002). Spécification structurelle, fonctionnelle et déontique d'organisations dans les sma. In *Jfsma*, p. 205–216.
- Hubner J. F., Sichman J. S., Boissier O. (2007). Developing organised multiagent systems using the MOISE+ model: Programming issues at the system and agent levels. *International Journal of Agent-Oriented Software Engineering*, vol. 1, n° 3-4, p. 370–395.
- Jarvenpaa S. L., Knoll K., Leidner D. E. (1998). Is anybody out there? Antecedents of trust in global virtual teams. *Journal of management information systems*, vol. 14, n° 4, p. 29–64.

- Jennings N. R. (1995). Controlling cooperative problem solving in industrial multi-agent systems using joint intentions. *Artificial intelligence*, vol. 75, n° 2, p. 195–240.
- Kerr N. L. (1983). Motivation losses in small groups: A social dilemma analysis. *Journal of Personality and Social Psychology*, vol. 45, n° 4, p. 819.
- Kuhn H. W. (1955). The Hungarian Method for the assignment problem. *Naval Research Logistics Quarterly*, vol. 2, p. 83–97.
- Lorini E., Demolombe R. (2008). From binary trust to graded trust in information sources: a logical perspective. In *Trust in agent societies*, p. 205–225. Springer.
- Marsella S. C., Pynadath D. V., Read S. J. (2004). PsychSim: Agent-based modeling of social interactions and influence. In *Proceedings of the international conference on cognitive modeling*, vol. 36, p. 243–248.
- Marsh S., Briggs P. (2009). Examining trust, forgiveness and regret as computational concepts. In *Computing with social trust*, p. 9–43. Springer.
- Mayer R. C., Davis J. H., Schoorman F. D. (1995). An integrative model of organizational trust. *Academy of management review*, vol. 20, n° 3, p. 709–734.
- McKnight D. H., Chervany N. L. (2001). Trust and distrust definitions: One bite at a time. In *Trust in cyber-societies*, p. 27–54. Springer.
- Mishra A. K. (1996). Organizational responses to crisis. *Trust in Organizations. Frontiers of theory and research*, p. 261–287.
- Palanski M. E., Kahai S. S., Yammarino F. J. (2011). Team virtues and performance: An examination of transparency, behavioral integrity, and trust. *Journal of Business Ethics*, vol. 99, n° 2, p. 201–216.
- Pinyol I., Sabater-Mir J. (2013). Computational trust and reputation models for open multi-agent systems: a review. *Artificial Intelligence Review*, vol. 40, n° 1, p. 1–25.
- Pinyol I., Sabater-Mir J., Dellunde P., Paolucci M. (2012). Reputation-based decisions for logic-based cognitive agents. *Autonomous Agents and Multi-Agent Systems*, vol. 24, n° 1, p. 175–216.
- Pizzi D., Cavazza M. (2007). Affective storytelling based on characters' feelings. In *Proceedings of the aaai fall symposium on intelligent narrative technologies*, p. 111–118.
- Porteous J., Charles F., Cavazza M. (2013). Networking: using character relationships for interactive narrative generation. In *Proceedings of the 2013 international conference on autonomous agents and multi-agent systems*, p. 595–602.
- Rousseau D. M., Sitkin S. B., Burt R. S., Camerer C. (1998). Not so different after all: A cross-discipline view of trust. *Academy of management review*, vol. 23, n° 3, p. 393–404.
- Tambe M. (1997). Towards flexible teamwork. *Journal of artificial intelligence research*, p. 83–124.
- Traum D., Swartout W., Marsella S., Gratch J. (2005). Fight, flight, or negotiate: Believable strategies for conversing under crisis. In *Intelligent virtual agents*, p. 52–64.
- Wong P. S., Cheung S. O., Ho P. K. (2005). Contractor as trust initiator in construction partnering—prisoner's dilemma perspective. *Journal of Construction Engineering and Management*, vol. 131, n° 10, p. 1045–1053.

