



HAL
open science

Activités collaboratives et génération de comportements d'agents : moteur décisionnel s'appuyant sur un modèle de confiance

Lucile Callebert

► To cite this version:

Lucile Callebert. Activités collaboratives et génération de comportements d'agents : moteur décisionnel s'appuyant sur un modèle de confiance. Autre [cs.OH]. Université de Technologie de Compiègne, 2016. Français. NNT : 2016COMP2299 . tel-01522687

HAL Id: tel-01522687

<https://theses.hal.science/tel-01522687>

Submitted on 15 May 2017

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

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

Par **Lucile CALLEBERT**

Activités collaboratives et générations de comportements d'agents : moteur décisionnel s'appuyant sur un modèle de confiance

Thèse présentée
pour l'obtention du grade
de Docteur de l'UTC



Soutenue le 18 octobre 2016

Spécialité : Technologies de l'Information et des Systèmes :
Unité de recherche Heudysiac (UMR-7253)

D2299

Thèse
pour l'obtention du grade de

Docteur de l'Université de Technologie de Compiègne
Spécialité : Technologies de l'Information et des Systèmes

Activités collaboratives et génération de
comportements d'agents :
moteur décisionnel s'appuyant sur un modèle de confiance

par
Lucile CALLEBERT

Soutenue le 18 octobre 2016 devant un jury composé de :

Mme Catherine PELACHAUD <i>Rapporteure</i>	Directrice de Recherche CNRS au LTCI, Télécom ParisTech
M. Nicolas SABOURET <i>Rapporteur</i>	Professeur, LIMSI-CNRS, Université Paris Sud
M. Didier BAZALGETTE <i>Examineur</i>	Direction de la Stratégie, Direction Générale de l'Armement
M. Pierre MORIZET-MAHOUDEAUX <i>Examineur</i>	Professeur, Heudiasyc, Université de Technologie de Compiègne
Mme. Magalie OCHS <i>Examinatrice</i>	Maître de conférences, LSIS, Polytech Marseille, Aix-Marseille Université
M. Jean-Paul BARTHÈS <i>Directeur</i>	Professeur émérite, Heudiasyc, Université de Technologie de Compiègne
Mme. Domitile LOURDEAUX <i>Directrice</i>	Maître de conférences, Heudiasyc, Université de Technologie de Compiègne





Résumé

Lorsqu'ils travaillent en équipe, les humains ont rarement des comportements optimaux : ils peuvent faire des erreurs, manquer de motivation ou de compétence. Dans les domaines des environnements virtuels ou des systèmes multi-agents, de nombreux travaux ont cherché à reproduire les comportements d'équipes humaines : un agent représente alors un membre de l'équipe. Cependant, ces travaux ont très souvent pour objectif la performance de l'équipe, et non la fidélité des comportements produits. Pour former un apprenant en environnement virtuel à prêter attention et à s'adapter aux autres, nous avons cherché dans cette thèse à reproduire des comportements humains réalistes et non-optimaux de travail d'équipe. Plus particulièrement, nous nous sommes intéressés aux équipes auto-organisées, c'est-à-dire aux équipes dans lesquelles le pouvoir de décision est réparti entre les membres, et dans lesquelles l'organisation est implicite. Dans de telles équipes, l'organisation se fait non pas au travers des communications mais par l'observation et l'anticipation des comportements des autres. Pour s'organiser, chaque agent doit se demander ce qu'il est préférable de faire en fonction de ce que pourraient faire les autres, et donc se poser des questions telles que *Ai-je confiance en la compétence de mon coéquipier pour faire cette tâche ?* Les relations de confiance permettent donc à chacun de prendre en compte les autres.

Pour générer de tels comportements, nous proposons un système permettant aux agents de raisonner d'une part sur un modèle de l'activité à effectuer et d'autre part sur les relations de confiance qui les lient aux autres agents de l'environnement. Dans ce cadre, notre première contribution porte sur l'augmentation du langage de description de l'activité Activity-DL de manière à permettre la description d'activités collectives. Nous proposons également des mécanismes de propagation de contraintes et d'informations qui faciliteront le raisonnement des agents. Ces contraintes et informations permettront par exemple aux agents de répondre à la question *Avons-nous les compétences nécessaires pour faire la tâche qui permettra de réaliser notre but collectif ?* Notre seconde contribution porte sur la proposition d'un modèle d'agent opérationnalisant le modèle de confiance de Mayer et al. (1995), sélectionné après une étude de la littérature en sciences humaines et sociales sur la confiance. La confiance d'un agent en un autre est décrite selon trois dimensions : l'intégrité, la bienveillance et les compétences. Chaque agent est donc défini par ces trois dimensions et a des croyances sur l'intégrité, la bienveillance et la compétence des autres agents. De plus chaque agent possède des buts qui lui sont personnels ainsi que des buts collectifs et devra donc choisir quel but privilégier. Finalement nous proposons un moteur décisionnel qui permet à chaque agent de calculer l'importance qu'il accorde à ses buts afin de sélectionner une tâche. Nous avons défini les mécanismes de calcul de l'importance des buts de manière à modéliser l'influence sur l'agent de ses croyances sur les autres, et pour sélectionner une tâche, l'agent raisonne à la fois sur les modèles d'activité et sur ses attentes à propos du comportement des autres, également générées à partir des croyances de l'agent sur les autres. Nous avons implémenté notre système et constaté qu'il répond à nos objectifs de génération de comportements d'équipe réalistes et non optimaux. Nous avons également conduit une évaluation perceptive préliminaire au cours de laquelle les participants ont notamment été capables de percevoir la confiance ou le manque de confiance d'un agent en un autre grâce à son comportement.



Abstract

When working in teams, humans rarely display optimal behaviors : they sometimes make mistakes, lack motivation or competence. In virtual environments or in multi-agent systems, many studies have tried to reproduce human teamwork : each agent acts as a team member. However, the main objective in those studies is the performance of the team : each agent should display optimal behavior, and the realism of those simulated behaviors is not a concern. To train someone in a virtual environment to pay attention to and to adapt to their teammates, we built a decision-making system for agents to display realistic and non-optimal behaviors. More specifically, we are interested in self-organized teams (i.e. teams where the decision power is decentralized among its members) and in implicit organization (i.e. when team members do not interact through communications but rather through the observation of others' behaviors). In such a team, each agent has to think about what it should do given what others could do. Agents then have to ask themselves questions such as *Do I trust my teammate's competence to perform this task ?* Trust relationships therefore allow agents to take others into account.

We propose a system that allows agents to reason, on the first hand, on models of the activity they have to do, and on the other hand, on trust relationships they share with others. In that context, we first augmented the Activity-Description Language so that it supports the description of collective activities. We also defined mechanisms for constraint generation that facilitates agent reasoning, by giving them the answer to questions like *Do we have the required abilities to perform the task which will achieve our goal ?* We then proposed an agent model based on the model of interpersonal trust of Mayer et al. (1995), that we selected after a study of trust in social science. This model describes trust relationship with three dimensions : the trustor trusts the trustee's integrity, benevolence and abilities. An agent is therefore defined through those three dimensions, and has a mental model of each other agent ; i.e. has trust beliefs about others' integrity, benevolence and abilities. Moreover each agent has both personal and collective goals (i.e. goals that are shared with other members of the team), and thus will have to decide which goal to focus on. Finally we proposed a decision-making system that allows an agent to compute the importance it gives to its goals and then to select a task. When computing goal importance, the agent is influenced by its trust beliefs about others, and to select a task, it reasons on the activity models and on its expectations about what others could do. Those expectations are generated from the agents' trust beliefs. We implemented our system and observed that it produces realistic and non-optimal behaviors. We also conducted a preliminary perceptive evaluation which showed that participants were able to recognize one agent's trust or lack of trust in another through the behaviors of the first one.



Remerciements

Cette thèse n'aurait pas vu le jour sans eux : je remercie tout d'abord mes directeurs de thèse Domitile Lourdeaux et Jean-Paul Barthès pour m'avoir donné l'occasion de faire mes premiers pas dans le monde de la recherche. Je les remercie pour leur présence, leur encadrement et leurs conseils pendant ces trois (longues) années. Je remercie également le laboratoire Heudiasyc et plus particulièrement l'équipe ICI pour m'avoir accueillie. Je remercie aussi bien sûr le Labex MST2 et la DGA, qui ont financé ces travaux.

Ma soutenance a été l'un des meilleurs moments de cette thèse : je remercie chacun des membres de mon jury pour leur présence, pour l'attention qu'ils ont portée à mon manuscrit, pour leurs remarques et questions pertinentes et pour les discussions enrichissantes que nous avons eues ce jour là. En particulier, un grand merci à mes rapporteurs Catherine Pelachaud et Nicolas Sabouret, pour votre relecture précise et vos critiques qui m'ont permis d'avancer, mais plus généralement pour m'avoir toujours ouvert votre porte pendant la thèse.

Pendant ces trois années de thèse, j'ai eu la chance de pouvoir enseigner à l'UTC. J'ai adoré cette expérience d'échange et de partage de connaissance, et je me suis vraiment épanouie dans cette activité. Les TD/TP étaient pour moi une petite bulle d'air chaque semaine (merci les étudiants!). Je remercie toutes les personnes avec qui j'ai pu travailler à cette occasion et en particulier les responsables d'UV qui m'ont fait confiance : Aziz Moukrim, Antoine Jouglet, Claude Moulin et Lynne Forest.

Les personnes avec qui j'ai partagé un bureau sont nombreuses, commençons par les "anciens" : Camille et Kevin, merci de m'avoir accueillie à bras ouverts et avec tant de bonne humeur, d'amour pour la recherche et d'étoiles plein les yeux! Juliette, bravo pour ta patience, ton recul et ta justesse d'analyse des situations, et merci de m'avoir permis de passer une journée de plus en conférence;-) Lauriane, on n'a passé que peu de temps dans le même bureau (juste assez pour que je te fasse peur, pardon), mais c'était un plaisir. Félicitations pour ton investissement dans la vie du labo et pour ton pouvoir conciliateur et fédérateur. Et évidemment un très grand merci pour avoir attentivement relu le chapitre 5 de ce manuscrit. Freddy... encore quelqu'un que je vais devoir féliciter pour son calme et son recul! Idir, je t'apprécie tellement que j'ai été obligée de changer de bureau pour venir te retrouver après quasiment un an de douloureuse séparation, que dire de plus! Et enfin... Mélody, je suis désolée de te priver de ton espace de travail favori :-P Merci d'avoir rendu cette dernière année beaucoup plus supportable. Merci pour ta présence, pour les repas (et les galères) partagés, pour les discussions interminables, et pour ton soutien psychologique sans faille.

Plus généralement, merci à tous ceux avec qui j'ai pu partager des discussions et des cafés (ou des bières) et qui m'ont écoutée râler : Azzeddine, Rémy, Yann, Florian, Nicolas.

Enfin, parce qu'ils m'ont accueillie à bras ouvert sans que je fasse partie de leur labo, je remercie toute la Greta-team.

... Et parce que quand il n'y en a plus, il y en a encore, je remercie ceux qui étaient là avant le commencement : Axel, David, Manu et Raphaëlle. La V2O reste le meilleur des bureaux!

Merci également à ceux avec qui je n'ai jamais partagé de bureau mais m'entourent depuis maintenant quelques années : Steph (bravo pour m'avoir écoutée pendant 2h!), Aurore, Pierre, merci de vous être déplacés quand j'ai décrété que j'avais trop de travail pour sortir de chez moi; Camille (tu es une super compagne de voyage!), et les autres aussi :-)

J'ai eu l'immense chance de pouvoir passer trois mois en stage à Northeastern University : merci Margot de m'avoir ouvert les portes de ce qui a été pour moi le Research Paradise! (Et soit dit en passant, un grand merci également, Margot et Émilie, pour m'avoir littéralement ouvert votre porte quand j'étais à la rue, et dans un moment si particulier pour vous!) Thanks Stacy for accepting me in the team. Thanks to both Stacy and Margot for criticizing my work and trying soooo hard to help me, I left Boston with so much more (to work on) than when I arrived. I LOVED working with you (I still have pictures of the white board!), but I also loved having coffee breaks, walking around the city, drinking beers, traveling to Cape Cod, testing scotch and clam cakes (no, they're not real cakes).

Ce manuscrit ne serait pas ce qu'il est sans eux : merci à Papa, Maman et Marine pour avoir relu *en entier* et corrigé d'innombrables fautes de français! Merci les parents pour m'avoir toujours poussée à faire de mon mieux dans mes études et à aller de l'avant. Et surtout, merci pour votre soutien en particulier cette dernière année, je sais que je vous en ai fait baver. Merci également, Papa, Maman et Maxime, pour votre présence à ma soutenance et pour avoir grandement contribué à la préparation (ou à la dégustation :-P) du pot! Plus généralement, merci ma petite famille (et aussi à Gisèle, Pascal et le reste de la famille Pecune/Lacrampe) pour votre présence toujours réconfortante, et pour les bons petits plats qui ont maintenu notre régime alimentaire à la limite de l'acceptable pendant la rédaction.

Et évidemment, le plus grand des MERCI (et tout mon amour) va à Florian. Parce que bien sûr, dans un couple, on se soutient, et on partage des hauts et des bas. Pour ça, tu as rempli ton rôle à merveille, tu m'as supportée et tu m'as écoutée me plaindre (parfois seulement hein, je reste géniale quand même ;-)). Mais tu as fait immensément plus : tu as critiqué mon travail (pour le meilleur), tu as fait preuve d'une patience infinie pour m'expliquer pourquoi telle chose n'allait pas ou comment faire telle autre. Tu as relu mes papiers, tu as pris du temps pour comprendre ce que je voulais faire et pour m'aider, en gérant ta propre thèse en parallèle (avec brio bien sûr). Plus que des discussions scientifiques, on a eu des sacrées disputes scientifiques (mais blâmons la génétique, c'est le caractère Callebert et je ne peux pas lutter :-P) à propos de ma thèse. Malgré tout, tu as eu l'immense courage (et générosité et honnêteté) de revenir à la charge chaque fois que j'en avais besoin, alors qu'il aurait été teeeeeellement plus facile

pour toi de dire *c'est bien*. Je t'ai déjà dit tout ça (et aussi que je pense que tu feras un jour un excellent directeur de thèse), mais ici je voudrais rajouter : merci de ne pas avoir écrit ma thèse.

Table des matières

1	Introduction	1
2	Comportements collaboratifs	7
2.1	Les approches pour la génération des comportements des personnages virtuels . . .	8
2.1.1	Approches scriptées	8
2.1.2	Approches décision centralisée	8
2.1.3	Approches personnages virtuels autonomes	9
2.1.4	Approches hybrides	10
2.1.5	Conclusion	11
2.2	Comportements collaboratifs	11
2.2.1	Formalisations pour la génération de comportements collaboratifs	12
2.2.2	Systèmes multi-agents pour la génération de comportements collaboratifs	13
2.2.3	Conclusion	18
2.3	Prise en compte des facteurs humains	19
2.3.1	De très nombreux facteurs	19
2.3.2	Approches pour la génération de comportements humains	20
2.3.3	Conclusion	24
2.4	Bilan du chapitre	24
3	Confiance – informatique	27
3.1	Personnages virtuels et confiance	27
3.1.1	Climat de confiance	28
3.1.2	La confiance comme support de décision	29
3.1.3	Conclusion	30
3.2	Systèmes de confiance numériques	30
3.2.1	Définition des systèmes de confiance numériques	32
3.2.2	Système de confiance de Marsh (1994)	32
3.2.3	AFRAS	33
3.2.4	REGRET	34
3.2.5	LIAR	35
3.2.6	Conclusion	35
3.3	Systèmes de confiance cognitifs	36
3.3.1	Repage et BDI+Repage	37
3.3.2	Modèle C&F de la confiance sociocognitive	41

3.3.3	ForTrust	45
3.3.4	Conclusion	48
3.4	Bilan du chapitre	48
4	Confiance – SHS	51
4.1	Confiance et comportements	51
4.1.1	Confiance et groupes	52
4.1.2	Confiance et individus	53
4.1.3	Conclusion	55
4.2	Mieux définir la confiance	56
4.2.1	Les différentes formes de confiance	56
4.2.2	Confiance contextualisée ou confiance décontextualisée ?	58
4.2.3	Croyances sur le futur	59
4.2.4	Conclusion	60
4.3	Modéliser la confiance	61
4.3.1	Les fondements de la confiance	61
4.3.2	Représentation de la confiance	64
4.3.3	Conclusion	66
4.4	Bilan du chapitre	67
5	Activité collective	69
5.1	Formalisme de description de l'activité	69
5.1.1	Besoins	70
5.1.2	De nombreuses possibilités pour la description de l'activité	71
5.1.3	Description de l'activité avec ACTIVITY-DL	73
5.2	Éléments liés aux aspects collectifs	76
5.2.1	Collaboration au niveau du scénario	77
5.2.2	Collaboration au niveau des actions	77
5.2.3	Spécification sur les tâches feuilles	78
5.2.4	Notations utilisées	78
5.2.5	<i>Exemple de description d'activité collective</i>	78
5.3	Raisonnement des agents sur l'arbre de tâches	79
5.3.1	Besoins	79
5.3.2	Aperçu de la solution proposée	81
5.4	Propagation des contraintes de faisabilité collective	82
5.4.1	Contrainte de faisabilité collective en termes de nombre de personnes	82
5.4.2	Contrainte de faisabilité collective en termes de compétences	83
5.4.3	<i>Exemple de propagation de contraintes de faisabilité collective sur un arbre de tâches ACTIVITY-DL</i>	90
5.4.4	Conclusion	90
5.5	Propagation dynamique d'informations	91
5.5.1	Information sur le nombre maximal de participants	92
5.5.2	Information sur les compétences	92
5.5.3	<i>Exemple de propagation dynamique d'informations</i>	93

5.5.4	Conclusion	95
5.6	Bilan du chapitre	96
6	Moteur décisionnel	99
6.1	Modèle d'agent	99
6.1.1	Buts des agents	100
6.1.2	Dimensions des agents	101
6.1.3	Croyances	103
6.1.4	<i>Exemple</i>	104
6.2	Fonctionnement général du moteur décisionnel	107
6.3	Importance des buts	109
6.3.1	Calcul initial de l'importance des buts	109
6.3.2	Prise en compte des autres	110
6.4	Utilité des tâches	116
6.4.1	Utilité pour x de faire la tâche lui-même	117
6.4.2	Utilité des tâches pour les autres selon x	120
6.5	Sélection d'une tâche	121
6.5.1	Génération des répartitions de tâches	121
6.5.2	Calcul de l'utilité des répartitions	123
6.5.3	Sélection de la meilleure répartition	123
6.5.4	<i>Exemple</i>	124
6.6	Processus récursif	125
6.6.1	Ensemble de tâches sur lesquelles x raisonne	125
6.6.2	Ensemble des agents sur lesquels x raisonne	126
6.7	Bilan du chapitre	127
7	Cas d'usages et résultats	131
7.1	Implémentation	131
7.1.1	Description de l'architecture	131
7.1.2	Exécution du système	134
7.2	Cas d'usage	136
7.2.1	Influence de l'intégrité – scénarios $S2a$ et $S2b$	136
7.2.2	Influence de la bienveillance – scénario $S3$	140
7.2.3	Importance des compétences – scénario $S4$	142
7.2.4	Confiance mal placée – scénario $S5$	144
7.2.5	Conclusion	146
7.3	Évaluations perceptives préliminaires	147
7.3.1	Procédure	148
7.3.2	Première évaluation préliminaire	148
7.3.3	Seconde évaluation préliminaire	152
7.3.4	Conclusion	155
7.4	Bilan du chapitre	155

8 Conclusion générale	157
8.1 Résumé des travaux	157
8.1.1 Description de l'activité et traitement des arbres de tâches	157
8.1.2 Moteur décisionnel	158
8.2 Limites	159
8.2.1 Limites liées à la description et au traitement de l'activité	159
8.2.2 Limites liées à notre moteur décisionnel	161
8.2.3 Limites liées à l'implémentation proposée	162
8.3 Perspectives	163
8.3.1 Perspectives à court terme	163
8.3.2 Perspectives à long terme	164
A Publications	169
B Représentation de l'activité	171
B.1 Agrandissements	171
B.2 ACTIVITY-DL : XML	179
B.3 ACTIVITY-DL : workflow	181
B.3.1 Représentation graphique	181
B.3.2 <i>Exemple</i>	181
C Notations	183
D Traitement de l'activité : autres cas	187
D.1 Autres cas de génération des contraintes de faisabilité collective	188
D.2 Autres cas de propagation d'informations	188
E Prise de décision	191
E.1 Scénario $S2a$	191
E.1.1 Importance des buts	191
E.1.2 Utilité des tâches	191
E.1.3 Sélection d'une tâche	191
E.2 Scénario $S2b$	194
E.2.1 Importance des buts	194
E.2.2 Utilité des tâches	194
E.2.3 Sélection d'une tâche	194
E.3 Scénario $S3$	196
E.3.1 Importance des buts	196
E.3.2 Utilité des tâches	196
E.3.3 Sélection d'une tâche	196
E.4 Scénario $S4$	199
E.4.1 Importance des buts	199
E.4.2 Sélection d'une tâche	199
E.5 Scénario $S5$	202
E.5.1 Sélection d'une tâche	202

F Évaluations préliminaires	205
Bibliographie	207

Table des figures

2.1	Schémas de (Hubner et al., 2007, pp. 7, 8) : spécification fonctionnelle à gauche et spécification structurelle à droite avec $\mathcal{M}OISE^+$	15
2.2	Schéma de (Hubner et al., 2007, p. 12) : les composants de $\mathcal{S} - \mathcal{M}OISE^+$	15
2.3	Schéma de (Tambe, 1997, p. 14) : hiérarchie d'opérateurs décrivant une activité de repérage qui pourrait être exécutée par une équipe d'hélicoptères dans le domaine militaire.	17
2.4	Architecture modulaire de PMFserv (Silverman et al., 2006b) utilisée pour la génération de comportements humains.	21
3.1	Exemple d'une FCM décrivant la confiance de l'agent <i>malade</i> en l'agent <i>médecin</i> (Castelfranchi and Falcone, 2010).	44
5.1	Exemple d'un arbre de tâches décrit en ACTIVITY-DL.	76
5.2	ACTIVITY-DL : éléments de nombre de personnes et de compétences.	80
5.3	Traitement des arbres de tâches : contraintes de faisabilité collective en termes de compétences associées aux tâches feuilles.	84
5.4	Traitement des arbres de tâches : contraintes de faisabilité collective en termes de nombre minimal de personnes et de compétences nécessaires.	90
5.5	Traitement des arbres de tâches : nombre maximal de personnes pouvant participer à chaque tâche et les informations sur les compétences à t_0	95
5.6	Traitement des arbres de tâches : nombre maximal de personnes pouvant participer à chaque tâche et les informations sur les compétences à t_1	95
5.7	Traitement des arbres de tâches : nombre maximal de personnes pouvant participer à chaque tâche et les informations sur les compétences à t_2	96
6.1	Arbres de tâches décrits en ACTIVITY-DL correspondant au scénario détaillé en 6.1.4	105
7.1	Arbres de tâches décrits en ACTIVITY-DL correspondants au scénario détaillé en 6.1.4.	136
7.2	Graphe dépeignant les comportements des agents Rose, Marine et Jade dans le scénario <i>SA</i>	149
7.3	Arbre de tâches ACTIVITY-DL représentant l'activité collective <i>sauver la victime</i>	153
B.1	Exemple d'un arbre de tâches décrit en ACTIVITY-DL.	172
B.2	ACTIVITY-DL : éléments de nombre de personnes et de compétences.	173

B.3	Traitement des arbres de tâches : contraintes de faisabilité collective en termes de compétences associées aux tâches feuilles.	174
B.4	Traitement des arbres de tâches : contraintes de faisabilité collective en termes de nombre minimal de personnes et de compétences nécessaires.	175
B.5	Traitement des arbres de tâches : nombre maximal de personnes pouvant participer à chaque tâche et les informations sur les compétences à t_0	176
B.6	Traitement des arbres de tâches : nombre maximal de personnes pouvant participer à chaque tâche et les informations sur les compétences à t_1	177
B.7	Traitement des arbres de tâches : nombre maximal de personnes pouvant participer à chaque tâche et les informations sur les compétences à t_2	178
B.8	Représentation de l'activité sous forme de workflow.	182
F.1	<i>Évaluation préliminaire</i> – Exemple d'une répartition de tâche entre agents.	206

Liste des tableaux

6.1	<i>S1</i> – Caractéristiques (en termes de buts et de dimensions personnelles) des agents.	106
6.2	<i>S1</i> – Importance initiale des buts calculée par Anna.	110
6.3	Valeurs de $\delta_{x,y_i}(\gamma)$ en fonction des valeurs de b_{x,y_i} et de la comparaison entre $Imp_{y_i}^x(\gamma)$ et $Imp_x(\gamma)$	112
6.4	<i>S1</i> – Influence de chacun des autres agents sur l’importance des buts pour Anna.	113
6.5	<i>S1</i> – Ce qu’Anna pense être l’influence des autres agents sur l’importance des buts pour Béa, Cara et Dora.	116
6.6	<i>S1</i> – Valeurs d’importance finale des buts pour tous les agents selon Anna.	117
6.7	<i>S1</i> – Valeurs d’utilité pour toutes les tâches et pour tous les agents calculées par Anna.	121
7.1	<i>S2a</i> – Caractéristiques (en termes de buts et de dimensions personnelles) des agents.	137
7.2	<i>S2a</i> – Valeurs d’importance finale des buts pour tous les agents selon Anna dans les scénarios <i>S1</i> et <i>S2a</i>	138
7.3	<i>S2b</i> – Caractéristiques (en termes de buts et de dimensions personnelles) des agents.	139
7.4	<i>S2b</i> – Valeurs d’importance finale des buts pour tous les agents selon Anna.	140
7.5	<i>S3</i> – Caractéristiques (en termes de buts et de dimensions personnelles) des agents.	141
7.6	<i>S3</i> – Valeurs d’importance finale des buts pour tous les agents selon Anna.	141
7.7	<i>S4</i> – Caractéristiques (en termes de buts et de dimensions personnelles) des agents.	142
7.8	<i>S4</i> – Valeurs d’utilité des tâches pour tous les agents selon Anna.	143
7.9	<i>S5</i> – Caractéristiques (en termes de buts et de dimensions personnelles) des agents.	144
7.10	<i>S5</i> – Valeurs d’utilité des tâches pour tous les agents selon Anna.	145
7.11	Moyennes m et écarts types σ obtenus pour chacune des affirmations et chacun des scénarios de rangement de bureau.	151
7.12	Valeurs t obtenues par les tests de Bonferroni sur chaque combinaison de variantes et sur (i) l’affirmation A1 et (ii) l’affirmation A2.	152
7.13	Moyennes m et écarts types σ obtenus pour chacune des affirmations et chacune des variantes du scénario de sauvetage de victime	155
C.1	Récapitulatif des notations utilisées dans le manuscrit.	183
E.1	<i>S2a</i> – Importance initiale des buts calculée par Anna.	191
E.2	<i>S2a</i> – Ce qu’Anna pense être l’influence des autres agents sur l’importance des buts pour chaque agent.	192

E.3	<i>S2a</i> – Valeurs d'utilité pour toutes les tâches et pour tous les agents calculées par Anna.	192
E.4	<i>S2b</i> – Importance initiale des buts calculée par Anna.	194
E.5	<i>S2b</i> – Ce qu'Anna pense être l'influence des autres agents sur l'importance des buts pour chaque agent.	195
E.6	<i>S2b</i> – Valeurs d'utilité pour toutes les tâches et pour tous les agents calculées par Anna.	195
E.7	<i>S3</i> – Importance initiale des buts calculée par Anna.	196
E.8	<i>S3</i> – Ce qu'Anna pense être l'influence des autres agents sur l'importance des buts pour chaque agent.	197
E.9	<i>S3</i> – Valeurs d'utilité pour toutes les tâches et pour tous les agents calculées par Anna.	197
E.10	<i>S4</i> – Importance initiale des buts calculée par Anna.	199
E.13	<i>S4</i> – Tableau des répartitions de tâches générées par Anna.	199
E.11	<i>S4</i> – Ce qu'Anna pense être l'influence des autres agents sur l'importance des buts pour chaque agent.	201
E.12	<i>S4</i> – Valeurs d'importance finale des buts pour tous les agents selon Anna.	201
E.14	<i>S5</i> – Tableau des répartitions de tâches générées par Anna.	202

Chapitre 1

Introduction

Environnements virtuels pour la formation

Les environnements virtuels sont aujourd'hui largement utilisés pour la formation car ils apportent de nombreux avantages : ils permettent par exemple de réduire les coûts de formation et de contrôler les situations d'apprentissage dans lesquelles évoluent les apprenants. Les domaines d'application couverts par les environnements virtuels pour la formation sont très variés. Les projets GVT (Gerbaud et al., 2008) et V3S (Barot et al., 2013) visent la formation à des procédures de travail dans des environnements sociotechniques complexes. Les projets STRIVE (Rizzo et al., 2011) et VICTEAMS (Huguet et al., 2016) s'intéressent à la gestion du stress dans les situations de crises : le premier vise la formation des militaires à des techniques de gestion de stress qui pourront être appliquées en situation de combat, alors que le second vise l'entraînement des leaders d'équipe médicale à la prise de décision en situation dégradée. Pour ces quatre exemples, les environnements virtuels offrent un avantage de taille : ils permettent de reproduire des environnements qui seraient difficiles d'accès et des situations qui seraient dangereuses dans la réalité tout en garantissant la sécurité des apprenants. Ils permettent un contrôle précis du déroulé du scénario de formation de manière à limiter la charge cognitive et émotionnelle ressentie par les apprenants.

Un autre des avantages majeurs de la formation en environnement virtuel, et dont tous ces projets tirent profit, est qu'ils réduisent le nombre de personnes nécessaires pour la formation. Par exemple, pour la formation des leaders d'équipes médicales de secours en situation de crise, un très grand nombre de personnes sont mobilisées : des "acteurs" qui jouent les victimes à secourir, une équipe médicale qui entoure le leader en formation. En environnement virtuel, ces victimes et membres de l'équipe de secours peuvent être joués par des personnages virtuels. Diminuer le nombre de personnes nécessaires à la formation permet d'en réduire les coûts et d'en faciliter l'accès en parant au problème lié à la disponibilité de ces personnes. Par ailleurs, les apprenants peuvent ressentir de l'appréhension à l'idée d'être jugés par d'autres personnes, et faire face à un personnage virtuel permet de réduire cette appréhension. Le projet TARDIS (Anderson et al., 2013) tire parti de cet avantage pour l'entraînement à un entretien d'embauche : les personnes font face à un recruteur virtuel. Les agents virtuels jouent donc un rôle central dans les environnements virtuels puisque l'apprenant est amené à observer leurs comportements voire à interagir avec eux.

Environnements virtuels collaboratifs

Certains environnements virtuels de formation sont dits collaboratifs : l'apprenant n'est pas seul mais entouré d'autres humains, réels ou artificiels, avec lesquels une forme de collaboration apparaît (Gerbaud and Arnaldi, 2009). C'est le cas par exemple dans le projet SECUREVI qui vise la formation des pompiers en environnement virtuel (Querrec et al., 2003). Dans ce cas, les personnages virtuels doivent être capables de s'organiser et de se répartir les tâches pour travailler ensemble. La génération de comportements collaboratifs pour les personnages autonomes est un problème difficile qui a été abordé à la fois dans le domaine des environnements virtuels de formation et dans le domaine des systèmes multi-agents (SMA).

La première difficulté est d'identifier et de décrire les activités que les agents doivent réaliser. En effet, pour que leurs comportements soient représentatifs de ceux observés sur le terrain, il est nécessaire de leur fournir une telle description de l'activité. Il s'agit alors d'identifier un formalisme de description de l'activité qui soit à la fois expressif et interprétable par les agents. Lorsque le but est de générer des comportements collaboratifs représentatifs des comportements humains, l'activité est souvent découpée sous forme de rôles (Chevaillier et al., 2012; Claude et al., 2015; Hubner et al., 2007; Tambe, 1997). Chaque agent joue alors un (ou éventuellement plusieurs) rôle(s) et doit effectuer les séquences d'actions correspondant à son (ses) rôle(s). Les équipes d'agents sont alors pré-organisées.

Équipes auto-organisées et collaboration émergente

Nous nous intéressons à une autre forme d'activités collaboratives : celles pour lesquelles les rôles ne sont pas prédéfinis et où l'organisation des membres de l'équipe doit se faire dynamiquement et selon la volonté de ses membres. Plovnick et al. (1975) définissent en effet une équipe comme un ensemble de deux personnes ou plus dont les activités doivent être coordonnées pour atteindre un but commun : la notion de rôle n'intervient pas dans cette définition.

Nous nous intéressons plus particulièrement aux petites équipes auto-organisées, c'est-à-dire aux équipes dans lesquelles les membres sont adaptatifs et dans lesquelles la prise de décision est décentralisée et répartie entre les membres : il n'y a pas de chef dirigeant l'ensemble de l'équipe. Dans ce type d'équipes, la collaboration n'est pas prédéfinie mais émergente (Anderson and Franks, 2003). L'entraînement à des activités collectives ne se limite plus alors pour chaque personne à accomplir une séquence de tâches et à attendre de leurs coéquipiers qu'ils fassent de même : il est également souhaitable d'entraîner l'apprenant à prêter attention aux autres membres de l'équipe de manière à mieux s'organiser voire à rattraper les erreurs des autres (Marc and Amalberti, 2002).

Organisation implicite

Les travaux de cette thèse portent sur la génération de comportements d'agents virtuels formant des équipes auto-organisées. La génération de tels comportements d'équipe a fait l'objet de nombreux travaux dans le domaine des SMA (Fan and Yen, 2004) : l'accent est alors mis sur la performance de l'équipe. Cet objectif de performance est atteint par la mise en place de protocoles permettant aux agents de communiquer leurs intentions aux autres de manière à assurer l'atteinte de l'objectif collectif. Ces protocoles reposent en grande partie sur des mécanismes

de communication entre les agents pour établir les responsabilités de chacun : l'organisation de l'équipe est explicite.

Néanmoins [Anderson and Franks \(2003\)](#) mettent en avant le caractère non nécessaire des communications lors du travail d'équipe : l'équipe peut s'organiser de manière implicite. Chaque agent tient alors compte des autres avec lesquels il interagit sans communication directe mais au travers de l'observation de l'état de l'environnement et des comportements des autres ([Cao et al., 1997](#)). Ce principe a notamment été appliqué dans des travaux en robotique : [Brown and Jennings \(1995\)](#) ont montré que deux robots peuvent coopérer et travailler ensemble sans communiquer. Nous nous positionnons dans ce cadre : par la génération de comportements individuels d'agents et sans communication explicite, nous souhaitons reproduire des équipes auto-organisées et voir la collaboration émerger.

Comportements optimaux, non optimaux et prise en compte des autres

Notre priorité n'est pas à la performance de l'équipe d'agents : notre objectif est avant tout de reproduire des comportements représentatifs des comportements humains. Toutefois de manière à pouvoir générer un large panel de situations d'entraînement dans lesquelles l'apprenant pourra être confronté à différents niveaux de difficulté, nos agents doivent à la fois être capables d'adopter des comportements (1) optimaux et (2) non optimaux du point de vue de l'équipe. (1) La simulation d'une équipe dont les comportements des agents sont optimaux pourrait être observée par les apprenants à titre d'exemple, alors que (2) pour des sessions d'entraînement l'apprenant pourrait être confronté à une équipe dont l'un des membres ne se comporte pas de manière optimale : l'apprenant devrait alors s'adapter voire rattraper les erreurs de ce coéquipier. Nous entendons par "se comporter de manière optimale", pour les membres de l'équipe, le fait de (i) privilégier la réalisation des objectifs collectifs par rapport à la réalisation de leurs objectifs personnels et de (ii) se répartir les tâches de manière équilibrée et de manière à ce que l'activité collective soit réalisée au mieux (i.e. les agents devront privilégier la réalisation des tâches pour lesquelles ils sont compétents). Si le fait de (i) privilégier les objectifs collectifs se rapporte à la motivation et à l'importance que les agents accordent à ces objectifs, c'est la capacité des agents à s'organiser qui détermine (ii) la bonne répartition des tâches.

De manière à reproduire des comportements représentatifs des comportements humains, les agents doivent doublement tenir compte des autres. (i) Premièrement, dans un groupe, la motivation d'une personne pour réaliser les buts collectifs est en effet influencée par sa perception de la motivation des autres ([Kerr, 1983](#)). Deuxièmement, (ii) se répartir les tâches de manière équilibrée requiert pour chaque individu de prendre les autres en compte.

Confiance

Cette prise en compte des autres peut se faire au travers d'un raisonnement sur les relations de confiance qui lient les agents. En effet, si [Castelfranchi and Falcone \(2010\)](#) considèrent la confiance comme "la glu de la société", [Gambetta \(2000\)](#) propose que la confiance influence toutes nos décisions, et en particulier celles prises dans le cadre d'activités collaboratives : chaque agent devra se poser des questions telles que *Ai-je confiance en la compétence de mon coéquipier pour faire cette tâche ?* Cela permettra à chaque agent d'anticiper les comportements

des autres et ainsi de faire lui-même un choix informé quant à l'action à sélectionner.

Qu'un agent ait un comportement optimal ou non, ces comportements doivent dans tous les cas être cohérents, crédibles et explicables (Kenny et al., 2007) : un manque de cohérence dans le comportement des agents peut nuire à la compréhension du scénario par les apprenants (Si et al., 2010). Or raisonner sur des relations de confiance permettra de générer des comportements non optimaux du point de vue de l'équipe mais cohérents et explicables du point de vue de l'agent : par exemple si la confiance d'un agent en un autre est surévaluée, les comportements de ce dernier ne seront pas à la hauteur des attentes du premier. Du fait que le premier sélectionne ses propres actions en fonction des attentes qu'il a fondées sur les comportements du second, l'efficacité globale de l'équipe s'en trouvera limitée (Erdem, 2003).

Problématique

Nous souhaitons donc générer des comportements d'agent virtuels qui puissent prendre part à des activités collaboratives. Pour que les comportements soient représentatifs de ceux observés sur le terrain, nous serons amenés à travailler sur un formalisme de description de l'activité. Nous souhaitons que nos agents soient capables de former des équipes s'auto-organisant de manière implicite : chaque agent devra prendre en compte les autres pour décider de ses propres comportements.

En particulier, nous devons être à la fois capables de produire des comportements d'agents optimaux et non optimaux. Produire de tels comportements est un problème complexe qui est rarement adressé dans le domaine des environnements virtuels ou dans celui des SMA où la priorité est souvent donnée à la performance. La difficulté principale réside dans le fait que, bien que ces comportements soient non optimaux, ils doivent néanmoins être cohérents et explicables de manière à ce que l'apprenant les comprenne.

Les travaux présentés dans cette thèse s'appuient donc sur un cadre pluridisciplinaire mêlant informatique et sciences humaines et sociales (SHS) : nous souhaitons générer des comportements d'agents virtuels qui soient cohérents, explicables et représentatifs des comportements humains. Nous appuyons pour cela notre proposition sur des modèles issus d'études en SHS. Nous nous plaçons dans le cadre des environnements virtuels collaboratifs, et nous souhaitons en particulier que nos agents soient capables de former des équipes et de s'auto-organiser dynamiquement et implicitement. Dans ce contexte, chaque individu de l'équipe constitue un système autonome, adaptable et complexe, capable de réflexion sur lui-même et sur les autres individus de l'équipe afin de prendre une décision informée. L'équipe constitue par conséquent un système de systèmes et la collaboration émerge de décisions individuelles et décentralisées.

Organisation du document

Nous commençons ce manuscrit par un état de l'art sur les systèmes permettant la génération de comportements d'agents virtuels et en particulier la génération de comportements collaboratifs, aussi bien dans le domaine des environnements virtuels que dans le domaine des SMA. Nous verrons qu'il existe de très nombreuses manières de générer des comportements collaboratifs, et si la majorité des systèmes sont concernés en priorité par la performance de l'équipe d'agents,

d'autres s'intéressent avant tout aux facteurs humains qui influencent la prise de décision. Cet état de l'art est présenté dans le chapitre 2.

Nous avons abordé dans cette introduction l'importance de la confiance dans les activités collaboratives : le chapitre 3 est dédié à un état de l'art sur l'utilisation de la confiance dans le domaine des environnements virtuels ainsi que dans le domaine des SMA. La confiance est en effet utilisée de différentes manières dans les environnements virtuels, par exemple dans des scénarios de négociation ou comme variable d'évaluation de la qualité des comportements produits. Dans le domaine des SMA, les systèmes de confiance visent principalement à la régularisation des échanges entre les agents.

Le chapitre 4 est consacré à l'étude de travaux portant sur la confiance dans le domaine des SHS. Cette étude nous permettra de mieux définir cette notion complexe et de mieux comprendre les influences de la confiance sur les comportements humains. Nous décrirons le modèle de confiance sur lequel nous appuierons notre proposition de moteur décisionnel.

Nous abordons dans le chapitre 5 notre première contribution portant sur l'augmentation du langage de description de l'activité *ACTIVITY-DL* pour permettre la description d'activités collaboratives. Nous proposerons également des mécanismes de traitement automatique de l'activité permettant de générer des contraintes et informations sur lesquelles s'appuiera le raisonnement des agents.

Le moteur décisionnel présenté dans le chapitre 6 constitue notre deuxième contribution. Ce moteur décisionnel s'appuie sur un modèle de confiance issu d'études en SHS et permet aux agents de raisonner dans un premier temps sur l'importance qu'ils accordent à leurs buts personnels et à leurs buts collectifs et dans un second temps sur les activités décrites en *ACTIVITY-DL* de manière à sélectionner une tâche. Les mécanismes de prise de décision sont définis de sorte que les agents tiennent compte des autres, et il sera ainsi possible de produire, selon le paramétrage des agents, à la fois des comportements optimaux et non optimaux, mais toujours cohérents et explicables.

L'architecture de notre système ainsi que des cas d'usage illustrant le fonctionnement du moteur décisionnel proposé seront présentés dans le chapitre 7. Nous décrirons également dans ce chapitre deux évaluations perceptives préliminaires que nous avons menées.

Nous concluons sur nos travaux dans le chapitre 8. Après avoir résumé nos contributions, nous détaillerons les limites de nos propositions ainsi que les perspectives de travaux futurs.

Chapitre 2

Génération de comportements de personnages virtuels collaboratifs

De nombreux environnements virtuels utilisent des personnages virtuels. Dans les environnements virtuels pour la formation, plusieurs rôles ont été attribués aux personnages virtuels (Luna et al., 2012). Ils peuplent parfois l’environnement, en jouant par exemple les passagers d’un avion dans un scénario de simulation d’évacuation en urgence (Sharma and Otunba, 2012). Ils peuvent aussi jouer le rôle de tuteur ou d’assistant pour l’apprenant, comme le font par exemple l’agent tuteur développé par Graesser and McNamara (2010) ou les agents Steve (Johnson and Rickel, 1997; Rickel et al., 2001) et Adele (Johnson, 2001; Johnson et al., 2003).

Certains environnements virtuels pour la formation sont dits collaboratifs car ils permettent une forme de collaboration. On distingue trois types d’environnements virtuels collaboratifs (Gerbaud and Arnaldi, 2009) :

- Plusieurs apprenants peuvent apprendre ou s’entraîner collaborativement à une procédure individuelle. Dans ce cas, l’un des apprenants exécute la procédure en question dans l’environnement virtuel, alors que les autres observent.
- Un apprenant peut apprendre ou s’entraîner individuellement à une procédure collective. L’apprenant est immergé dans l’environnement virtuel et entouré de personnages non-joueurs avec lesquels il collabore.
- Enfin, plusieurs apprenants peuvent apprendre ou s’entraîner collaborativement à une procédure collaborative. Ils sont alors immergés dans l’environnement virtuel où ils collaborent.

Les environnements virtuels collaboratifs où un apprenant collabore avec plusieurs personnages non-joueurs constituent un outil précieux pour la formation : ils permettent l’entraînement d’une personne sans mobiliser de formateur ou d’acteurs qui jouent le rôle des coéquipiers. Nous nous intéressons dans le cadre de nos travaux aux personnages non-joueurs qui vont peupler l’environnement virtuel de formation et collaborer avec l’apprenant. Pour que la formation soit effective, ces personnages virtuels doivent avoir des comportements (i) cohérents et explicables, de manière à ce que l’apprenant puisse les comprendre et les anticiper. En effet, Si et al. (2010) ont démontré que des comportements incohérents nuisent à la compréhension de la situation par le sujet. Ces comportements doivent également être (ii) crédibles, c’est-à-dire qu’un humain réel pourrait adopter de tels comportements, et ce pour que l’utilisateur soit plongé dans le scénario

(Kenny et al., 2007). Par exemple, on considère qu’une équipe dont les agents ne font *jamaïs* d’erreur n’est pas crédible : comme le dit l’expression, l’erreur est humaine, et l’on cherche à reproduire des comportements semblables à ceux des humains. Enfin ces comportements doivent être (iii) représentatifs de ceux observés sur le terrain, de manière à ce que l’activité et les processus psychologiques activés soient identiques en environnement réel et en environnement virtuel (on peut alors parler de fidélité psychologique (Barot, 2014)). Nous nous intéressons à la génération de ces comportements. De cette manière, les apprentissages acquis en environnement virtuel pourront être transposés au monde réel.

2.1 Les approches pour la génération des comportements des personnages virtuels

Il existe plusieurs approches pour générer le comportement des personnages virtuels dans les environnements virtuels. Ces approches répondent à différents objectifs. Nous détaillons ces approches dans les sections suivantes.

2.1.1 Approches scriptées

Dans certains environnements virtuels collaboratifs pour la formation, le comportement des personnages virtuels est scripté. C’est le cas par exemple dans les travaux de Sharma and Otunba (2012). Dans ce cas, un petit nombre de scénarios est prévu, car il est fastidieux et coûteux de décrire le comportement de chaque personnage virtuel dans chaque scénario. Les possibilités de rejeu sont donc limitées. Cette approche a cependant ses avantages : elle permet un grand contrôle des scénarios d’apprentissage, assure l’atteinte des objectifs pédagogiques, et assure la cohérence des comportements de personnages virtuels (dans la limite de la cohérence des comportements décrits par le game designer).

2.1.2 Approches décision centralisée

Dans certains environnements virtuels, les décisions relatives aux comportements des personnages virtuels sont centralisées dans un seul module. Les motivations à adopter une telle approche sont doubles :

- Certains environnements virtuels sont centrés sur la narration ou sur l’intrigue (Cavazza et al., 2002) : un module de scénarisation assure le passage par certaines situations clés qui permettent d’atteindre des objectifs narratifs ou pédagogiques (Carpentier, 2015). Le module de scénarisation a alors la responsabilité du comportement des personnages virtuels.

Dans Cristal Island, l’apprenant mène une enquête qui lui permet d’améliorer ses connaissances en microbiologie (Mott and Lester, 2006). L’architecture U-DIRECTOR est utilisée et vise à produire dynamiquement un scénario cohérent en fonction de ce que fait l’apprenant. Des réseaux bayésiens dynamiques sont utilisés pour décider des événements ainsi que des comportements des personnages non-joueurs.

- Dans d’autres environnements virtuels où une équipe de personnages virtuels doit réaliser une tâche collaborative, il s’agit avant tout de garantir une organisation cohérente de

l'équipe. Un moteur de décision centralisé permet alors de choisir les actions à réaliser par chaque personnage de manière à ce que l'ensemble soit cohérent.

Dans le projet 3D Virtual Operating Room (3DVOR), un apprenant fait partie d'une équipe médicale qui réalise une intervention chirurgicale en bloc opératoire. La génération des comportements de personnages non-joueurs est assurée par l'application de la méthode Monte Carlo Tree Search (Sanselone et al., 2014) : un plan global est généré dans lequel des actions sont assignées à différents personnages non-joueurs. La génération du plan par la méthode Monte Carlo Tree Search s'appuie sur une description de l'activité à réaliser à l'aide de diagrammes Business Process Model Notation. La prise de décision quant au comportement des personnages virtuels est donc centralisée, ce qui assure une organisation optimale de l'équipe de personnages virtuels.

Les approches décision centralisée assurent donc la cohérence globale du scénario et/ou l'organisation cohérente d'une équipe de personnages virtuels ; cependant elles n'assurent pas nécessairement la cohérence des comportements individuels des personnages virtuels. Szilas (2007) propose également un moteur de scénarisation *IDtension*, notamment utilisé dans le jeu sérieux Nothing For Diner (Habonneau et al., 2012) qui a pour but l'atteinte d'objectifs narratifs tout en garantissant la consistance des comportements des personnages non-joueurs. Cependant dans ces travaux les règles qui permettent la génération des comportements des personnages virtuels et notamment le calcul de leurs motivations sont *ad hoc* : il est alors difficile d'assurer la crédibilité et l'explicabilité des comportements produits.

2.1.3 Approches personnages virtuels autonomes

Il est également possible d'utiliser des personnages virtuels autonomes pour peupler l'environnement virtuel, comme par exemple dans les projets EmoEmma (Pizzi and Cavazza, 2007), MRE (Hill Jr et al., 2003) ou FearNot! (Aylett et al., 2006). Utiliser des personnages virtuels autonomes permet d'obtenir une multitude de simulations différentes à partir du même contenu : comme il n'est pas nécessaire de spécifier tous les comportements de tous les personnages virtuels pour chaque situation, l'effort de programmation à fournir est moindre par rapport aux approches où les comportements sont complètement scriptés. Cette approche représente donc une solution à notre problématique.

Dans le cadre de l'utilisation de personnages autonomes et par rapport à notre problématique, deux aspects sont à prendre en compte :

- *les aspects collaboratifs*. Lorsque plusieurs personnages autonomes doivent accomplir ensemble une tâche collaborative, il est nécessaire de définir les mécanismes de collaboration qui vont leur permettre d'interagir et de s'organiser. En effet, les comportements n'étant pas scriptés, il faut prévoir les mécanismes qui permettront aux agents de s'entendre sur le but à atteindre, sur la manière d'atteindre ce but, sur une répartition des tâches, etc. Cette problématique a notamment été abordée dans le domaine de systèmes multi-agents (SMA) : certains SMA ont été définis pour la génération de comportements d'équipe. Nous abordons ce point dans la section 2.2.
- *les facteurs humains*. Pour produire des comportements humains, on peut définir un moteur décisionnel *ad hoc*, c'est-à-dire que l'on produit soi-même les règles de décision en fonction de ce que l'on cherche à obtenir et de ce que l'on pense être la meilleure solution

pour obtenir des comportements proches de ceux des humains. C’est le cas par exemple dans les travaux de [Johnson and Rickel \(1997\)](#) sur l’agent tuteur Steve. À l’inverse, en particulier si l’on s’intéresse aux facteurs humains (e.g. stress, confiance, émotions, etc.), on peut s’appuyer sur des théories en sciences humaines et sociales (SHS) qui visent à expliquer le comportement humain de manière à essayer de reproduire du mieux possible les comportements humains¹. Opérationnaliser de telles théories représente un gros travail de recherche pour trouver la théorie adaptée à la situation ou au type de comportements que l’on essaie de reproduire, et des efforts conséquents de compréhension, d’interprétation et d’adaptation de cette théorie qui à l’origine n’a pas été pensée pour être implémentée. Cependant, procéder de la sorte permet d’obtenir des comportements plus crédibles et explicables.

Nous nous intéressons à la prise en compte des facteurs humains pour la génération des comportements d’agents (dans les SMA) ou de personnages virtuels autonomes dans la section 2.3.

En nous intéressant à la fois aux aspects collaboratifs et aux facteurs humains, nous cherchons à produire, pour des personnages virtuels autonomes, des comportements cohérents, crédibles, explicables et représentatifs de ceux observés sur le terrain.

2.1.4 Approches hybrides

Les approches dites hybrides tentent de combiner les avantages à la fois des approches décision centralisée et des approches décision décentralisée. Dans ces approches, les personnages sont autonomes, mais sont d’une certaine manière manipulés par un module externe. Deux catégories d’approches hybrides sont à distinguer :

- Un module externe peut proposer une distribution d’actions optimale aux personnages virtuels. Ceux-ci peuvent ensuite choisir de suivre le conseil reçu ou de faire une autre action.

L’architecture GVT (Generic Virtual Trainer) permet l’entraînement individuel à des procédures collaboratives ([Claude et al., 2015](#); [Gerbaud et al., 2009](#)). Un module *action filter* de distribution d’action analyse les actions qui doivent être réalisées pour respecter le scénario décrit avec LORA – *Language for Object-Relation Application* ([Mollet and Arnaldi, 2006](#)) et les met en correspondance avec les personnages qui seraient aptes à les réaliser (en fonction des compétences, des qualifications et des ressources que possède chaque personnage). Ce module génère ensuite une distribution d’actions qui est proposée à chaque personnage autonome.

- Sans proposer d’action au personnage virtuel, un module externe peut déclencher des événements dans le monde et/ou modifier certains paramètres internes des personnages virtuels (e.g. leurs buts) pour orienter le choix d’action de ces derniers.

Dans la plateforme HUMANS, les personnages virtuels sont autonomes ([Lanquepin et al., 2013](#); [Lhommet, 2012](#)), mais le module SELDON assure d’une part la génération dynamique d’objectifs scénaristiques avec le moteur TAILOR ainsi que la réalisation de ces objectifs scénaristiques avec le moteur DIRECTOR. Ce dernier propose dynamiquement

1. Néanmoins, il est tout à fait possible de s’intéresser aux facteurs humains tels que le stress ou les émotions et proposer un moteur décisionnel *ad hoc* qui intègre ces facteurs ([Barthès et al., 2016](#)).

des ajustements à apporter à l’environnement virtuel pour diriger le scénario vers l’objectif voulu. Ces ajustements peuvent concerner le déclenchement d’un événement dans l’environnement virtuel ou le *late commitment* de l’état d’un objet ou d’un personnage virtuel, c’est-à-dire l’ajout, au cours de la simulation, d’un fait qui sera considéré rétrospectivement comme faisant partie des conditions initiales (e.g. on décide en cours de simulation que la porte, dont l’état était incertain jusqu’à cet instant, est ouverte) (Barot, 2014).

Ces approches permettent de combiner les avantages des deux précédentes. Nous nous intéressons dans nos travaux à la génération des comportements des personnages autonomes, néanmoins notre moteur de génération des comportements de personnages virtuels pourrait être couplé à un module de scénarisation comme SELDON.

2.1.5 Conclusion

Les approches qui utilisent des personnages virtuels autonomes permettent donc de générer un très grand nombre de scénarios de formation puisque chaque scénario n’a pas besoin d’être complètement spécifié. De plus la cohérence du comportement des personnages est assurée et un module externe peut être ajouté pour influencer le comportement des personnages si besoin.

Certains systèmes utilisant des personnages autonomes ignorent complètement les facteurs humains, ce qui peut nuire à la qualité des comportements produits ainsi qu’à celle de la formation. En effet, prendre en compte ces facteurs permettra d’obtenir des comportements plus proches de la réalité, ce qui favorisera l’apprentissage. De plus, de nouveaux types d’apprentissage peuvent être envisagés. Par exemple, si les personnages virtuels peuvent ressentir et exprimer des émotions, il sera possible de proposer un entraînement à la prise de décision dans des situations à forte charge émotionnelle (Hill Jr et al., 2003). Nous aborderons la problématique de la prise en compte des humains plus en détail dans la section 2.3.

Nous nous intéressons dans le cadre de cette thèse à des activités collaboratives. Si nous utilisons des personnages autonomes, il est alors nécessaire de définir les mécanismes qui leur permettront de collaborer. Nous développons dans la section suivante la génération de comportements collaboratifs pour des personnages autonomes.

2.2 Comportements collaboratifs

Nos travaux sont centrés sur la génération de comportements collaboratifs pour des personnages virtuels en environnement virtuel de formation. Après avoir détaillé les mécanismes de génération des comportements des personnages virtuels, nous nous intéressons ici à la nature collaborative de ces comportements. Cette problématique a été abordée dans le domaine des environnements virtuels pour la formation (Barange et al., 2014; Gerbaud et al., 2009), mais surtout dans le domaine des SMA. En effet, les SMA ont été utilisés pour la modélisation des comportements humains et du travail d’équipe.

2.2.1 Formalisations pour la génération de comportements collaboratifs

La collaboration dans les SMA est un problème à part entière sur lequel de nombreux chercheurs se sont penchés. Plusieurs formalisations pour la génération de comportements collaboratifs ont été proposées, dont notamment les formalisations Joint Intentions et SharedPlan qui ont été d'une influence majeure pour les travaux ultérieurs. D'autres formalisations existent, mais les deux précédemment citées nous semblent être les plus importantes et ce sont celles que nous développons ici.

Joint Intentions

La théorie des *Joint Intentions* capture la nécessité, dans des situations de collaboration, non seulement d'avoir un but commun sur lequel chacun des membres d'une équipe s'est engagé, mais aussi de s'engager à communiquer les informations relatives à ce but commun (Cohen and Levesque, 1990, 1991; Levesque et al., 1990).

Les auteurs définissent la notion de *joint persistent goal* : "deux agents x et y ont le joint persistent goal p sous la condition q " signifie que x et y ont la croyance mutuelle² que p est faux mais ont le but mutuel que p soit vrai³ et x et y ne peuvent abandonner ce but que sous trois conditions : (i) x et y ont la croyance mutuelle que p est vrai ou (ii) x et y ont la croyance mutuelle que p est irréalisable ou (iii) x et y ont la croyance mutuelle que q est fausse. Dans ce cas, la notion de croyance mutuelle implique que si x ou y se rend compte (i.e. ont la croyance individuelle) que p est vrai, irréalisable ou que q est fausse, alors l'agent a le but de le faire savoir à l'autre, i.e. d'établir la croyance mutuelle. La notion de *joint intention* est alors définie par rapport à celle de joint persistent goal : " x et y ont la joint intention de faire a sous la condition q " signifie que x et y ont le joint persistent goal de faire a consciemment⁴ tant que q est vraie.

Les notions de joint persistent goal et de joint intention permettent de spécifier les éléments en termes de croyances et d'intentions qui sont nécessaires pour que la collaboration émerge et se déroule correctement : chaque agent est engagé sur l'activité collective et ne peut se démettre de ses responsabilités que dans des conditions spécifiques, et doit alors en informer les autres. Chaque agent a également l'assurance que les autres sont engagés sur cette activité, et que, si un membre de l'équipe se désengage de l'activité commune, il en informera les autres.

Cette formalisation est notamment utilisée dans le système STEAM.

SharedPlans

La notion de plan partagé a été définie pour formaliser les éléments nécessaires en termes de croyances pour l'émergence de la collaboration dans un groupe d'agents (Grosz and Kraus, 1996; Grosz and Sidner, 1988; Lochbaum et al., 1990).

De manière similaire à la théorie des Joint Intentions, le but de la théorie SharedPlan est de formaliser les éléments de connaissance de chaque individu du groupe nécessaires pour assurer le

2. Dire que deux agents x et y ont la croyance mutuelle que φ est vrai signifie que : x pense que φ est vrai, y pense que φ est vrai, x pense que y pense que φ est vrai, y pense que x pense que φ est vrai, x pense que y pense que x pense que φ est vrai, etc.

3. " x et y ont le but mutuel p " signifie que x et y ont la croyance mutuelle que chacun d'eux a le but p .

4. " x fait a consciemment" signifie " x pense faire a puis x fait a ".

succès d'une action collective complexe (i.e. composée d'une séquence de sous-actions). Chaque individu doit être engagé sur l'activité collective et avoir les connaissances et capacités pour mener à bien une partie de l'activité collective, tout en étant assuré que les autres membres du groupe sont également engagés sur l'action collective et ont également les connaissances et capacités nécessaires. Cependant, la formalisation SharedPlan est plus souple que la théorie des Joint Intentions quant à la communication : dans la théorie des Joint Intentions, un agent *doit* communiquer dès lors qu'il abandonne une joint intention, alors qu'avec SharedPlan, l'agent *peut* communiquer s'il abandonne une intention. Par ailleurs, la principale contribution de la théorie des SharedPlan est qu'elle permet une meilleure spécification de la manière dont l'action doit être réalisée, alors que dans la théorie des Joint Intentions, on ne s'intéresse qu'au résultat.

Dans la théorie SharedPlans, il existe une distinction entre deux types d'intentions : l'*intention de* désigne l'intention de faire une action, et l'agent qui a cette intention s'engage à faire cette action. L'*intention que* est beaucoup moins contraignante puisqu'elle désigne l'intention qu'une proposition soit rendue vraie, et l'agent s'engage à faire ce qu'il peut pour que cette proposition soit rendue vraie (i.e. dans certains cas, rien).

La définition d'un plan partagé est alors la suivante : lorsqu'un groupe d'agents Gr a le plan partagé $SP = \text{SharedPlan}(Gr, A)$ d'exécuter l'action complexe (i.e. action composée) A , alors :

- Les agents de Gr ont la croyance mutuelle qu'ils ont l'intention d'exécuter A ;
- Les agents de Gr ont la croyance mutuelle qu'ils connaissent un moyen (recipe) R_A de réaliser A ;
- Pour chaque sous-action A_j de R_A , (i) les agents de Gr ont la croyance mutuelle que Gr_j , un sous-groupe de Gr , a un plan partagé SP_j pour faire A_j ^{5,6}, (ii) Gr_j a effectivement le plan partagé SP_j de faire A_j et (iii) les agents de Gr ont la croyance mutuelle que chacun d'eux a l'*intention que* Gr_j puisse faire A_j .

Le plan partagé peut être complètement spécifié, on parle alors de Full SharedPlan, ou partiellement spécifié, auquel cas on parle de Partial SharedPlan. Lorsque le plan est partiellement spécifié, cela signifie que des grandes lignes du plan sont dessinées, mais les détails (i.e. la manière de réaliser un objectif précis du plan) ne sont pas définis. Dans ce cas, il revient aux agents exécutant le plan de choisir comment atteindre l'objectif en question.

Nous nous intéressons dans la section suivante aux SMA dans lesquels les agents peuvent avoir des comportements collaboratifs. Nous verrons que si certains systèmes s'appuient sur les formalisations type Joint Intentions ou SharedPlan, il existe d'autres approches pour générer de tels comportements.

2.2.2 Systèmes multi-agents pour la génération de comportements collaboratifs

Il existe de nombreux SMA (SMA) qui ont pour objectif de simuler des comportements collaboratifs d'équipe : on peut citer par exemple GRATE* (Jennings, 1995), CAST (Yen et al., 2001) ou RESTINA-MAS (Giampapa and Sycara, 2002). Nous renvoyons le lecteur aux revues de Fan and Yen (2004) et de Sycara and Sukthankar (2006) pour un état de l'art plus détaillé

5. Si Gr_j n'est composé que d'un seul agent alors cet agent n'a pas un plan partagé mais un plan individuel.

6. Il n'est pas nécessaire pour les agents g de Gr n'appartenant pas Gr_j de connaître le sous-plan partagé SP_j .

sur ces systèmes. Nous présentons ici trois de ces systèmes qui adoptent chacun une approche différente pour la génération des comportements collaboratifs. Nous pourrions ainsi comprendre les objectifs visés par chaque approche et les avantages et inconvénients de chaque approche à la lumière de ces exemples.

MOISE, MOISE⁺ et \mathcal{S} – MOISE⁺

MOISE – *Model of Organization for multi-agent SystEms* (Hannoun et al., 2000), enrichi dans MOISE⁺ (Hübner et al., 2002), est un méta-modèle pour la description d’organisation. \mathcal{S} – MOISE⁺ (Hubner et al., 2007) est un framework pour la construction de SMA qui permet d’assurer que les agents suivent les spécifications prescrites dans MOISE⁺.

MOISE⁺ pour la spécification d’organisation. MOISE⁺ (Hübner et al., 2002) distingue trois types de spécifications différentes :

- La *spécification structurelle* vise à décrire la structure de la société d’agents, en termes de rôles et/ou de groupes d’agents.

Dans MOISE⁺, on distingue trois niveaux qui composent la spécification structurelle :

- (i) *Niveau individuel*. Les rôles définissent les champs de compétence et de responsabilités des agents qui les jouent.
- (ii) *Niveau social*. Les rôles sont liés entre eux. Plusieurs types de liens existent entre les rôles (e.g. les liens de communication ou les liens d’autorité).
- (iii) *Niveau collectif*. Un agent pouvant jouer plusieurs rôles, il existe des règles qui définissent la compatibilité des rôles : des groupes définissent les rôles pouvant être joués ensemble. Un exemple de spécification structurelle avec MOISE⁺ est donné figure 2.1 (droite).

- La *spécification fonctionnelle* contient les éléments qui décrivent le fonctionnement de l’organisation. Par exemple, une telle spécification peut contenir des éléments de description de l’activité qui sera réalisée collectivement par les agents, des mécanismes qui leur permettront de se coordonner, etc.

L’activité que les agents doivent réaliser pour atteindre leurs buts est ici décrite sous forme de plans qui consistent en une décomposition hiérarchique de buts. Ces buts sont ordonnés grâce à l’utilisation des opérateurs *séquence*, *choix* ou *parallélisme*. De plus ces buts sont regroupés sous forme de mission. Un exemple de spécification fonctionnelle avec MOISE⁺ est donné figure 2.1 (gauche).

- La *spécification déontique* contient les éléments qui permettent de lier les spécifications structurelle et fonctionnelle, c’est-à-dire les droits et les obligations (en termes de réalisation de but) liés aux différents rôles et/ou groupes.

Dans MOISE⁺, les rôles et obligations sont définis de la manière suivante : le rôle ρ aura la permission $per(\rho, m_1, tc_1)$ de faire la mission m_1 pendant la période tc_1 (i.e. tc pour *time constraint*), et aura l’obligation $obl(\rho, m_2, tc_2)$ de faire la mission m_2 pendant la période tc_2 .

\mathcal{S} – MOISE⁺ pour la simulation d’organisations. Le framework \mathcal{S} – MOISE⁺, illustré dans la figure 2.2, permet d’assurer que les agents suivent les spécifications prescrites dans MOISE⁺. Ce framework est notamment composé d’un super-agent OrgManager qui est chargé

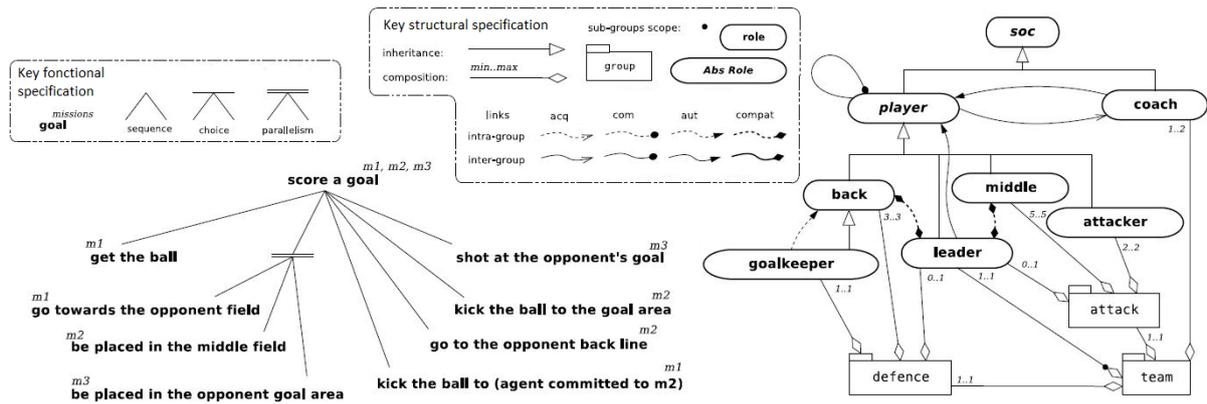


FIGURE 2.1 – Schémas de (Hubner et al., 2007, pp. 7, 8) : spécification fonctionnelle à gauche et spécification structurelle à droite avec \mathcal{MOISE}^+ .

de maintenir la cohérence de l’organisation au regard des spécifications décrites avec \mathcal{MOISE}^+ . Par exemple, lorsqu’un agent souhaite changer de rôle, s’engager dans une mission ou former un nouveau groupe, il adresse une requête au OrgManager qui décidera d’accepter ou non la proposition. L’OrgManager informe aussi les agents concernés lorsqu’un nouveau but lié à leur rôle est débloqué, ce qui facilite la coordination. Toutes ces communications entre agents et OrgManager se font par l’intermédiaire d’agents OrgBox : chaque agent possède un agent OrgBox qui lui sert d’interface de communication.

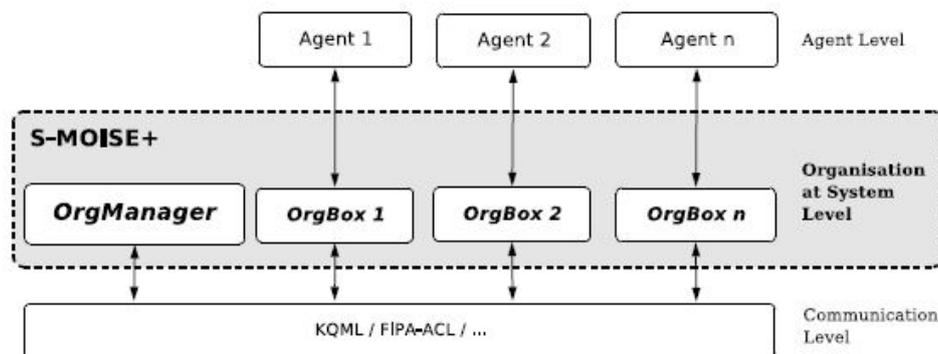


FIGURE 2.2 – Schéma de (Hubner et al., 2007, p. 12) : les composants de $\mathcal{S} - \mathcal{MOISE}^+$.

Avantages et limites. Le système $\mathcal{MOISE}^+/\mathcal{S} - \mathcal{MOISE}^+$ adopte une approche centrée organisation : les mécanismes de coopération sont décrits dans la spécification de l’organisation et la coopération est assurée. En effet, à l’opposé des approches centrées agents dans lesquelles les agents sont au cœur du problème (i.e. les mécanismes de coopération sont définis au niveau des agents, qui sont libres de coopérer ou non), dans les approches centrées organisation l’accent est mis sur l’organisation du collectif d’agents plutôt que sur les individus qui le composent (Hübner et al., 2002). Les agents ont une certaine liberté puisqu’ils peuvent demander de jouer un rôle qu’ils sélectionnent eux-mêmes mais cette liberté est très limitée. En effet la décision finale ne leur revient pas puisque c’est le super-agent OrgManager qui accepte ou refuse leur proposition. Ici l’accent est donc mis sur la performance finale : on souhaite obtenir une équipe performante

et pour cela on limite l'autonomie des agents qui la composent.

Dans le cadre de nos travaux, nous ne cherchons pas à proposer une équipe d'agents qui soit la plus performante possible : nous souhaitons mettre l'apprenant face à une équipe d'agents dont les comportements soient proches de ceux des humains, i.e. une équipe dont les agents font parfois des erreurs, ne sont pas toujours enclins à coopérer, etc. Nous souhaitons voir la coopération émerger, et nous nous intéressons autant à chaque agent du système qu'au système lui-même. Les approches centrées organisation ne conviennent donc pas pour nos travaux.

STEAM

STEAM – *Shell for TEAMwork* (Tambe, 1997) est un modèle générique de travail d'équipe et permet la coordination dynamique de plusieurs agents ayant des rôles complémentaires.

Génération des comportements collaboratifs dans STEAM. STEAM s'appuie sur la théorie des Joint Intentions. Les agents établissent dans un premier temps une joint intention de réaliser un joint persistent goal ϕ . Ainsi, chaque agent a l'intention de réaliser ϕ et sait que chaque autre agent de l'équipe a l'intention de réaliser ϕ également. Le but ϕ est ensuite découpé en sous-buts, et un joint persistent goal est formé récursivement pour chaque sous-but, ce qui revient à établir un plan commun \mathbf{p} , à la manière des plans partagés SharedPlans (Grosz and Kraus, 1996; Grosz and Sidner, 1988). STEAM possède également un mécanisme de réparation qui force les agents à chercher une autre solution pour atteindre le but ϕ lorsque le plan \mathbf{p} ne peut plus être achevé.

Dans STEAM, chaque plan est construit par un assemblage hiérarchique d'opérateurs sous forme d'arbre où l'opérateur racine correspond au but à atteindre, et où chaque but est décomposé en sous-buts jusqu'aux opérateurs feuilles de l'arbre qui correspondent à des actions, cette hiérarchie étant décrite par le concepteur du système. Les opérateurs peuvent être individuels (i.e. cette partie du plan se fait individuellement) ou d'équipe (i.e. cette partie du plan doit être faite par l'équipe ou par une sous-équipe). Lorsqu'un opérateur d'équipe est sélectionné, une joint intention est automatiquement instanciée. Des contraintes logiques spécifient les conditions de validation d'un opérateur ayant des sous-opérateurs : la *AND-combination* spécifie que tous les sous-opérateurs doivent être réalisés et la *OR-combination* spécifie qu'un seul des sous-opérateurs a besoin d'être réalisé. Enfin STEAM utilise la notion de rôle pour associer les opérateurs devant être effectués par le même groupe d'agents. La figure 2.3 illustre la décomposition hiérarchique du but en sous-buts. Les *AND-combination*, les *OR-combination* et les rôles ne figurent pas sur le schéma.

L'équipe est également décrite par le concepteur du système qui précise les éventuelles sous-équipes, les liens de communication, le leader, etc. Il n'y a cependant pas de mémoire partagée et chaque agent possède sa propre copie des informations relatives au plan et à l'état de l'équipe en cours de simulation.

Avantages et limites. On peut caractériser STEAM comme un système hybride entre les approches orientées agents et celles orientées organisation : des mécanismes pour l'émergence de la collaboration sont définis au niveau des agents avec l'utilisation des Joint Intentions, et une spécification fonctionnelle est donnée aux agents par la description de l'activité à réaliser

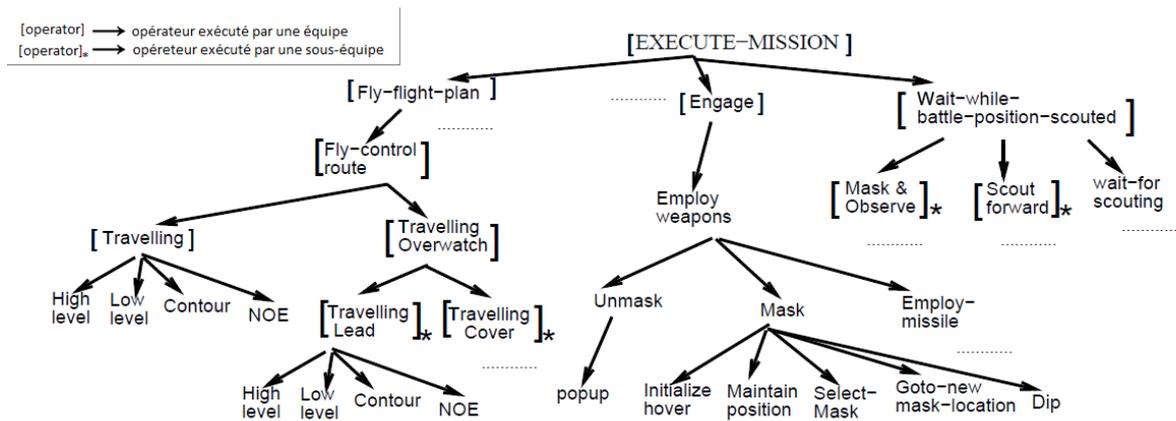


FIGURE 2.3 – Schéma de (Tambe, 1997, p. 14) : hiérarchie d’opérateurs décrivant une activité de repérage qui pourrait être exécutée par une équipe d’hélicoptères dans le domaine militaire.

sous forme d’une hiérarchie d’opérateurs prédéfinie par le concepteur du système. Les agents déterminent un plan p qui leur permette d’atteindre le but ϕ grâce cette description de l’activité.

Ce type d’approche semble adapté à notre problématique puisqu’elle permet d’obtenir des comportements individuels cohérents (i.e. chaque agent est motivé par ses propres buts) et représentatifs de ceux observés sur le terrain grâce à la description de l’activité sur laquelle les agents raisonnent.

Cependant, de par l’appui théorique des théories des Joint Intentions et des SharedPlan, l’accent est encore une fois mis sur la performance de l’équipe d’agents et la collaboration passe par la communication. Pourtant Castelfranchi (1998) met en évidence qu’un agent n’est pas social parce qu’il communique, mais communique parce qu’il a des comportements sociaux : les agents interfèrent, dépendent les uns des autres, et s’influencent mutuellement. La communication n’est donc pas nécessairement un élément indispensable pour produire des comportements sociaux. Des agents pourraient collaborer sans communiquer, simplement en observant et interprétant les comportements des autres. Par exemple, si l’on imagine un scénario où deux agents pompiers assistent à l’effondrement d’une maison, chaque agent sait sans avoir besoin de communiquer que l’autre va vouloir intervenir pour sauver la personne prise dans les décombres. C’est aussi en permettant cette collaboration implicite que l’on pourra obtenir des situations où les agents ne s’organisent pas bien ; or c’est ce type de situation qui nous intéresse dans le cadre de cette thèse.

DEPNET

DEPNET est un simulateur proposé par Sichman et al. (1998) dont l’objectif est de produire des agents ayant des comportements sociaux. Pour cela DEPNET opérationnalise le *raisonnement social*.

Raisonnement social dans DEPNET. Le raisonnement social est défini par les mêmes auteurs comme un raisonnement dans lequel un agent utilise une croyance à propos d’un autre agent pour en tirer une conclusion. Pour que les agents soient capables d’un tel raisonnement,

chaque agent ag_i a des croyances sur chaque autre agent ag_j sous forme de *description externe* $Ext_{ag_i}(ag_j)$, composée (i) des buts de ag_j , (ii) des actions que ag_j peut réaliser, (iii) des ressources dont ag_j dispose et (iv) des plans de ag_j pour réaliser ses buts⁷. Les agents peuvent ainsi raisonner sur les croyances qu'ils ont sur les autres, c'est-à-dire simuler le raisonnement des autres⁸.

Les auteurs s'intéressent aux liens de dépendance entre les agents. Un agent ag_i peut être dépendant d'un agent ag_j pour réaliser son but g_k par le plan P_k pour principalement deux raisons : (1) ag_i ne peut pas faire une ou plusieurs actions du plan P_k , ou (2) ag_i n'a pas une ou plusieurs ressources nécessaires pour réaliser le plan P_k . En raisonnant sur ses buts, actions, ressources et plans et sur ses descriptions externes des autres, un agent peut donc construire un *réseau de dépendances*. Les agents peuvent ainsi identifier :

- les situations de coopération, dans lesquelles ag_i et ag_j sont mutuellement dépendants l'un de l'autre pour la réalisation du même but g_k ;
- les situations d'échange social, dans lesquelles ag_i et ag_j sont réciproquement dépendants l'un de l'autre pour la réalisation de leurs buts respectifs g_i et g_j ;
- les situations de dépendance unilatérale, dans lesquelles ag_i dépend de ag_j pour réaliser son but g_i mais ag_j ne dépend pas de ag_i pour réaliser son but g_j ;
- les situations d'indépendance, où ag_i n'est dépendant d'aucun autre agent pour réaliser son plan.

Avantages et limites. Avec cette formalisation de raisonnement social, les auteurs se situent dans une approche émergente de la collaboration, ou une approche centrée agent (Hübner et al., 2002). Ce système est intéressant puisqu'il met en évidence que la collaboration n'est pas acquise : chaque agent a des buts personnels à réaliser, il n'y a pas de notion d'équipe : on s'intéresse avant tout aux individus et non au système. L'autonomie des agents est totale : les agents peuvent ne pas collaborer avec d'autres s'ils n'y voient pas d'intérêt, mais on peut supposer que, si les agents sont dépendants les uns des autres pour réaliser leurs buts, la collaboration émergera.

Toutefois les mécanismes décisionnels qui permettent à l'agent de choisir quoi faire en fonction de ce réseau de dépendances ne sont pas précisés ; il est donc difficile d'avoir une idée précise des comportements produits par ce système.

2.2.3 Conclusion

Nous nous sommes tout d'abord intéressés aux formalisations des Joint Intentions et des SharedPlans : de telles formalisations procurent un support pour la définition de SMA dont les agents ont des comportements collaboratifs. Elles permettent de spécifier les éléments nécessaires au bon déroulement de cette collaboration, notamment en ce qui concerne la nécessité d'établir des croyances mutuelles.

Nous avons ensuite étudié trois SMA pour la génération de comportements collaboratifs. Ces trois systèmes adoptent chacun une approche différente : $\mathcal{MOISE}^+/\mathcal{S}$ – \mathcal{MOISE}^+ adopte

7. Dans l'implémentation de leur modèle théorique, Sichman et al. (1998) n'utilisent que des croyances qui correspondent à la réalité ; on peut alors parler de connaissances, i.e. les agents ne se trompent pas sur les autres.

8. On dit que ces agents sont dotés de la théorie de l'esprit (voir p. 59).

une approche centrée organisation dans laquelle la collaboration est prédéfinie et donc assurée. DEPNET adopte à l’opposé une approche centrée agents, dans laquelle la collaboration est entièrement émergente : les agents sont capables d’avoir un raisonnement social et ont la possibilité de ne pas collaborer ; ce qui doit également être le cas dans nos travaux. Enfin, STEAM adopte une approche hybride : les agents ont la liberté de collaborer ou non, et une description de l’activité à réaliser (produite par le concepteur du système) permet d’obtenir des comportements représentatifs de ceux observés sur le terrain.

Les approches hybrides sont donc les plus adaptées à notre problématique. Pour obtenir des comportements collaboratifs de personnages virtuels qui soient représentatifs de ceux observés sur le terrain, nous avons besoin, comme dans STEAM, d’une description de l’activité qui pourra être réalisée. Cependant, nous souhaitons que les comportements générés soient crédibles pour chaque individu. Un agent doit pouvoir faire des erreurs, choisir de réaliser une action même si cela n’est pas le choix optimal pour l’équipe : nous souhaitons obtenir des comportements les plus semblables possibles à ceux des humains.

Les travaux en SHS sur les comportements humains peuvent nous éclairer quant à la manière de générer de tels comportements. Nous passons donc en revue dans la section suivante quelques travaux dans lesquels sont opérationnalisées des théories de SHS sur le comportement humain et ce dans le cadre de la génération de comportements d’agents virtuels.

2.3 Prise en compte des facteurs humains

Pour la génération de comportements crédibles et explicables, une approche consiste à étudier les théories en SHS sur les comportements humains et à opérationnaliser ces théories. Nous nous intéressons ici aux travaux dans le domaine des SMA ou dans le domaine des personnages virtuels qui tiennent compte de facteurs humains pour la génération des comportements des agents.

2.3.1 De très nombreux facteurs

De nombreuses études en SHS portent sur l’analyse des comportements humains, et on peut trouver dans la littérature une multitude de facteurs qui influencent ces comportements.

Par exemple, de nombreux travaux ont pointé la nécessité pour un groupe d’avoir des buts, connaissances et plans partagés pour coopérer de manière efficace. La notion de modèle mental en particulier a été étudiée : par exemple Cannon-Bowers et al. (1993) expliquent la nécessité pour une équipe d’avoir un modèle mental partagé de la tâche à effectuer. Mathieu et al. (2000) montrent que le degré de similitude entre les modèles mentaux de la tâche à effectuer de chacun des membres d’une équipe prédit la performance de cette équipe. Ce type d’expérience a été reproduit sur une équipe d’agents par Harbers et al. (2012) et donne les mêmes résultats. Ces travaux sur les modèles mentaux partagés ont notamment été utilisés dans le domaine des environnements virtuels de formation par Mancheva and Dugdale (2016) qui s’intéressent aux erreurs produites en bloc opératoire à cause de mauvaises communications. Ces travaux s’appuient sur le principe de la conscience mutuelle de la situation (*mutual situation awareness*) : chaque agent maintient une représentation mentale de la situation qui est mise à jour à chaque communication et, dans l’idéal, les représentations de tous les agents concordent. C’est grâce à cette conscience mutuelle de la situation que les comportements coopératifs peuvent émerger.

De plus les comportements déviants et les erreurs sont interprétés en termes d'écart entre les représentations mentales de chaque agent.

Un autre exemple d'opérationnalisation d'une théorie de la discipline des SHS pour la génération de comportements d'agents est celui de la théorie de l'identité sociale de [Tajfel and Turner \(1979\)](#). Selon cette théorie, un individu peut exprimer plusieurs identités : l'identité personnelle est exprimée en présence des membres du groupe auquel on appartient, ce qui nous permet de mettre en avant nos différences avec ceux-ci ; alors qu'une identité sociale est exprimée lorsque l'on est confronté à des personnes extérieures à notre groupe, pour mettre en avant les caractéristiques que l'on partage avec notre groupe. Ces différentes identités sont exprimées au travers des comportements adoptés. [Dimas and Prada \(2013\)](#) opérationnalisent cette théorie et proposent un modèle dynamique d'identité pour des agents capables d'ajuster leurs comportements au contexte social dans lequel ils se trouvent.

Dans d'autres travaux, plusieurs facteurs d'influence sont opérationnalisés dans un même agent. Par exemple, les émotions, la personnalité et les relations sociales sont souvent traitées ensemble : [Ochs et al. \(2009\)](#) notamment s'appuient sur le modèle de personnalité OCEAN ([McCrae and John, 1992](#)) et utilisent les traits de névrotisme et d'extraversion, qui influencent respectivement le ressenti des émotions positives (joie, fierté, soulagement) et négatives (tristesse, peur, honte, colère, déception). Les relations sociales sont décrites selon les dimensions d'appréciation, de dominance, de solidarité et de familiarité. Leur dynamique est définie en fonction des émotions ressenties par les personnages virtuels (e.g. si i cause de la joie à j , l'appréciation de i pour j augmente). Ces facteurs sont utilisés dans d'autres travaux : on peut par exemple citer [Lhomme \(2012\)](#) qui s'intéresse à la contagion émotionnelle entre agents virtuels et prend également en compte la personnalité, les émotions et les relations sociales des agents, ou [Heudin \(2016\)](#) qui propose un modèle pour la génération d'attitudes affectives tenant compte de la personnalité, des émotions et de l'humeur.

Énumérer les facteurs qui influencent les comportements humains ou énumérer les travaux en informatique qui s'intéressent à cette problématique n'est pas possible car ils sont trop nombreux. Cependant, nous abordons dans la section suivante les deux grandes approches qui existent pour opérationnaliser dans des agents virtuels l'influence de facteurs humains sur le comportement.

2.3.2 Approches pour la génération de comportements humains

Pour générer des comportements de personnages virtuels semblables à ceux des humains, deux approches sont donc possibles : intégrer de multiples facteurs dans un même personnage autonome, ou intégrer un seul ou un petit nombre de facteurs. Nous développons un exemple de ces deux approches dans les sections suivantes pour mieux en comprendre les avantages et les limites.

Intégration de multiples facteurs

[Silverman](#) s'est intéressé à l'étude de facteurs dits modérateurs de performance : en influençant la prise de décision, ces facteurs modulent la performance des individus.

[Silverman et al. \(2006b\)](#) identifient certains facteurs biologiques, cognitifs, émotionnels et sociaux qui influencent la prise de décision individuelle et l'émergence des comportements sociaux

de groupe. Les théories et études sur l'influence de chacun de ces facteurs sur le comportement sont nombreuses, mais le but avancé par Silverman et al. (2006b) n'est pas d'étudier un facteur en particulier ou d'argumenter en faveur de l'une de ces théories mais plutôt de proposer un modèle alternatif qui intègre un ensemble de théories. Ce modèle devrait également permettre d'étudier les relations, interactions et influences de chacun de ces facteurs sur les autres et les auteurs mettent en avant la possibilité d'intégrer facilement de nouveaux facteurs modérateurs de performance ainsi que les règles qui détermineront leurs influences sur et interactions avec les autres facteurs.

Approche modulaire. Pour répondre à cette problématique, les auteurs proposent l'architecture cognitive *Unified Behavior Architecture*, implémentée dans PMFserv (PMF pour *Performance Moderator Function*) et décrite dans la figure 2.4.

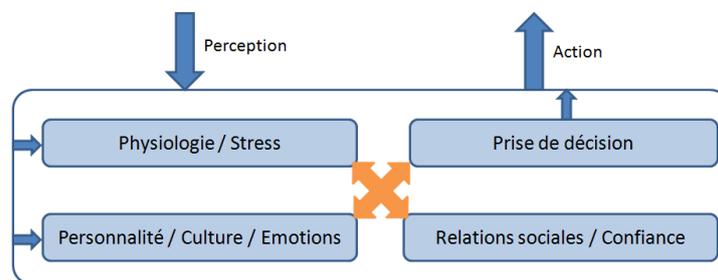


FIGURE 2.4 – Architecture modulaire de PMFserv (Silverman et al., 2006b) utilisée pour la génération de comportements humains.

PMFserv possède une structure modulaire. Ses principaux modules sont les suivants :

- Le module *Physiologie / Stress* traite de l'influence des facteurs biologiques sur la performance. Les facteurs choisis sont ceux identifiés par Gillis and Hursh (1999)⁹ : la perception d'un événement stressant, la pression temporelle et la fatigue. La combinaison de ces facteurs par des règles *ad hoc* détermine un style de coping (i.e. opérationnalisation de la théorie de Janis and Mann (1977)⁹).
- Le module *Personnalité / Culture / Émotions* vise à déterminer un état émotionnel pour le personnage virtuel. Des traits de personnalité et des valeurs culturelles déterminent ses buts, standards et préférences, et la théorie OCC (Ortony et al., 1990) est opérationnalisée pour calculer l'état émotionnel suite à l'occurrence d'un événement et en fonction des buts, standards et préférences du personnage.
- Le module *Relations sociales / Confiance* vise à permettre au personnage virtuel de raisonner à propos des autres personnages. Pour cela, il est nécessaire de connaître la relation qu'il entretient avec eux. Des groupes d'agents sont identifiés, et chaque autre agent de l'environnement peut faire partie du même groupe ou être ennemi. En appliquant la théorie de l'esprit pour raisonner sur les intentions des autres, l'agent est capable de déterminer s'il peut leur faire confiance.
- Le module *Prise de décision* permet au personnage virtuel de sélectionner une action. La sélection d'action repose sur une fonction de maximisation de l'utilité qui prend en compte

9. Cité par Silverman et al. (2006b).

l'état émotionnel et les relations sociales ainsi que les événements passés. Différentes fonctions de maximisation d'utilité doivent être appliquées selon le mode de coping actif pour le personnage, mais à notre connaissance aucune n'est présentée par les auteurs. Ces modules ne constituent pas des "boîtes" indépendantes ; chaque élément calculé par un module peut avoir une influence sur un autre. Par exemple, le style de coping et les relations sociales d'un personnage influencent le calcul de son état émotionnel, alors que l'état émotionnel influence à son tour le choix du style de coping.

Dans des travaux plus récents, [Nye and Silverman \(2013\)](#) s'intéressent au processus d'adoption d'un nouveau comportement observé. L'architecture PMFserv est utilisée pour générer les comportements des agents, et l'approche adoptée pour la modélisation du processus d'adoption est la même que précédemment : de très nombreux facteurs influençant le fait qu'un individu décide d'adopter un comportement ou non sont identifiés. Ces facteurs correspondent à des théories issues de disciplines différentes (e.g. psychologie sociale, psychologie expérimentale, sciences cognitives, etc.).

Les auteurs mettent en avant la simplicité d'utilisation de leur architecture et la possibilité d'ajouter ou d'enlever très simplement des facteurs modérateurs de performance de par sa structure modulaire ([Silverman et al., 2006b](#)). Leur architecture est notamment utilisée dans des projets militaires ([Silverman et al., 2002, 2006a, 2012](#)) et pour la santé ([Silverman et al., 2016](#)).

Avantages et limites. Ces travaux présentent une approche intéressante : de par le caractère modulaire de l'architecture, il est facile d'ajouter ou d'enlever des facteurs influençant la prise de décision. Cependant des questions se posent quant à la validité du modèle. D'une part, les auteurs choisissent d'opérationnaliser différentes théories issues des sciences humaines. Ces modèles n'étant originellement pas destinés à faire l'objet d'une implémentation, les auteurs ont fourni de gros efforts d'interprétation pour obtenir une opérationnalisation mais n'ont ensuite pas évalué cette opérationnalisation. De même, l'influence de chacun des facteurs identifiés sur les autres n'est pas toujours étudiée dans la littérature et les interactions de chacun de ces facteurs sont donc définies par les auteurs mais ne s'appuient pas toujours sur une étude valide et ne font pas l'objet d'une évaluation.

D'autre part, les théories utilisées recoupent de nombreux domaines : [Janis and Mann \(1977\)](#) qui ont travaillé sur les styles de coping sont des chercheurs en psychologie sociale ; les travaux de [Gillis and Hursh \(1999\)](#) s'inscrivent dans la discipline des sciences du comportement ; etc. Rien ne garantit que ces théories soient compatibles et qu'elles s'intègrent dans un même cadre théorique cohérent.

Enfin, de par les nombreux facteurs pris en compte et les influences multiples de chaque facteur, le modèle peut s'avérer très complexe à paramétrer et donc difficilement utilisable dans un contexte de formation. Par exemple, il est difficile de comprendre exactement comment la confiance a été opérationnalisée et comment elle influence la prise de décision.

Petit nombre de facteurs

Dans d'autres travaux, les chercheurs se sont concentrés sur un petit nombre de facteurs pour produire des personnages virtuels ayant des comportements humains. L'enjeu est alors de respecter un cadre théorique valide, d'expliquer les choix d'opérationnalisation de chaque

facteur et des liens qui permettent de les associer pour produire un modèle représentatif des comportements humains et des processus cognitifs en jeu. Le modèle produit peut ensuite être utilisé pour évaluer la théorie sous-jacente par simulation informatique. C'est l'objectif visé par le modèle computationnel d'émotions et des processus cognitifs sous-jacents EMA – pour *EMotion and Adaptation* (Gratch and Marsella, 2004; Marsella and Gratch, 2009).

Un fort cadre théorique. L'objectif des travaux de Gratch and Marsella est de proposer un modèle computationnel des processus cognitifs liés aux émotions et à leurs expressions et influences sur le comportement. EMA est inspiré de la théorie de l'évaluation cognitive de Smith et Lazarus (1990). Selon cette théorie, une évaluation cognitive permet à l'individu de construire, par le processus d'*appraisal*, une représentation mentale et une interprétation d'un événement à la lumière de ses buts et de ses croyances. L'événement est interprété selon différents critères, appelés variables d'*appraisal* (e.g. *désirabilité* : cet événement facilite-t-il ou menace-t-il mes buts?). Dans la phase de *coping*, l'individu choisit une réponse aux événements qui ont été interprétés lors de la phase d'*appraisal*. Deux grands types de stratégies de coping sont identifiés (Lazarus, 1991) : le coping centré problème, dans lequel l'individu agit sur l'environnement, et le coping centré émotion, dans lequel l'individu modifie sa réponse émotionnelle à l'événement en le réinterprétant différemment. Pour chacun de ces types, plusieurs stratégies ont été identifiées (e.g. une stratégie de coping centré émotion est le déni : on ignore l'événement et ses conséquences).

Gratch and Marsella (2004) proposent dans le modèle EMA une opérationnalisation de la théorie de l'évaluation cognitive de Smith et Lazarus (1990).

Le modèle EMA a notamment été utilisé dans le projet MRE – *Mission Rehearsal Exercise* qui vise à l'entraînement, en environnement virtuel, de leaders militaires à la prise de décision en situation critique (Hill Jr et al., 2003).

Avantages et limites. Cette approche ne vise pas l'intégration d'un grand nombre de facteurs. Au contraire, les auteurs plaident pour la simplification de certaines théories de l'évaluation cognitive (notamment celles de Smith et Kirby (2000) et Scherer (2001)) en soulignant qu'un système plus simple comme EMA permet de produire le même type de comportements complexes et semblables à ceux des humains tout en restant facilement utilisable (Marsella and Gratch, 2009). La validité du cadre théorique dans lequel s'inscrivent ces travaux est assurée par l'opérationnalisation d'une seule théorie issue d'une discipline unique. De plus, les auteurs présentent une évaluation du modèle EMA (Marsella et al., 2009) : des sujets humains participent à un jeu de bataille navale. À certains points clés du jeu (début, milieu, fin), les sujets remplissent un questionnaire dont les réponses rendent compte de leur appréciation de la situation (selon les dimensions d'*appraisal* utilisées dans EMA) et de leur stratégie de coping. Ces résultats sont comparés à l'*appraisal* et à la stratégie de coping sélectionnés par EMA dans la même situation. Cette évaluation permet de valider deux des trois dimensions d'*appraisal* évaluées (le désengagement et la résignation).

2.3.3 Conclusion

Les études en SHS identifient une multitude de facteurs qui influencent les comportements humains. Pour reproduire dans des agents virtuels des comportements semblables à ceux des humains, il est possible de s'appuyer sur des théories de SHS. Deux approches sont alors envisageables : de nombreux facteurs peuvent être intégrés ensemble dans l'espoir d'une part de rendre compte de la complexité du comportement humain et d'autre part que le modèle permette de gérer un très large éventail de situations. Adopter une approche modulaire pour l'intégration de ces différents facteurs permet alors de faciliter l'intégration ou la suppression de certains facteurs. Cependant cela rend la validité du modèle difficile à vérifier et à évaluer. La présence de multiples facteurs rend également le modèle difficile à paramétrer et à utiliser dans un contexte de formation. Travailler avec peu de facteurs permet de pallier à ces désavantages et les comportements générés ne sont pas nécessairement plus limités.

Nous présentons dans le chapitre suivant quelques travaux qui portent sur l'utilisation de la confiance pour la génération de comportements de personnages autonomes en environnement virtuel.

2.4 Bilan du chapitre

Nous nous sommes intéressés dans ce chapitre à la génération des comportements de personnages virtuels en environnement virtuel pour la formation. Nous avons dans un premier temps identifié différentes approches pour la génération des comportements de personnages virtuels : les comportements peuvent être scriptés, auquel cas les possibilités de rejou sont très limitées. Lorsque les comportements ne sont pas scriptés, ils peuvent être générés par un système centralisé, comme par exemple un module de scénarisation. Cela permet d'assurer une cohérence globale du scénario et/ou une organisation optimale d'un groupe de personnages, mais engendre un risque de perte de cohérence au niveau du comportement individuel de chaque agent, ce qui peut nuire à l'engagement du sujet. Pour éviter ces pertes de cohérence, les personnages peuvent être autonomes. Cela permet de plus de réduire les efforts de programmation nécessaires pour générer un grand nombre de scénarios. Nous adoptons donc une approche personnages autonomes dans le cadre de cette thèse.

Notre problématique étant la génération de comportements collaboratifs semblables à ceux des humains, nous nous sommes intéressés dans un second temps aux travaux, notamment dans le domaine des SMA qui permettent la génération de comportements collaboratifs. Nous avons vu qu'il existe des formalisations pour la génération de tels comportements : ces formalisations précisent les éléments nécessaires à la collaboration notamment en termes de croyances mutuelles. De nombreux SMA se sont appuyés sur ces formalisations (i.e. ont opérationnalisés les mécanismes qui permettent d'obtenir ces croyances mutuelles) pour générer des comportements d'équipe, comme par exemple STEAM. Nous avons vu avec l'exemple de $MOISE^+/\mathcal{S} - MOISE^+$ et DEPNET que d'autres approches sont possibles pour générer les comportements collaboratifs : $MOISE^+/\mathcal{S} - MOISE^+$ adopte une approche centrée organisation dans laquelle la collaboration est assurée par la spécification de l'activité à réaliser et des rôles que chaque agent devra jouer dans la réalisation de cette activité. À l'inverse, DEPNET adopte une approche centrée agents

dans laquelle la collaboration est émergente ; les agents décident en fonction de leurs buts s'ils souhaitent ou non collaborer avec les autres. Nous souhaitons dans le cadre de nos travaux voir émerger la collaboration : en effet, nous voulons avoir des agents qui puissent par exemple décider de ne pas participer à la réalisation de la tâche collaborative ou à l'inverse, un agent i qui pourrait aider l'agent j à réaliser l'un de ses buts personnels. Cependant nous nous intéressons à des activités complexes, et pour obtenir des comportements représentatifs de ceux observés sur le terrain, nous avons besoin, comme dans STEAM, d'une description de l'activité qui sera réalisée. Nous choisissons donc dans nos travaux d'adopter une approche hybride entre les approches centrées agents et celles centrées organisation.

Enfin, nous avons abordé la problématique de l'intégration des facteurs humains. Nous avons en effet constaté que dans la plupart des SMA pour la génération de comportements collaboratifs (e.g. STEAM, $\mathcal{M}OISE^+/\mathcal{S} - \mathcal{M}OISE^+$, et d'autres encore comme GRATE* ou CAST pour n'en citer que quelques-uns), la priorité est d'obtenir une équipe d'agents performante sur la tâche collective ; or ce n'est pas notre objectif. Nous avons donc étudié des SMA et des environnements virtuels dans lesquels les facteurs humains sont pris en compte pour la génération des comportements d'agents, et en particulier nous nous sommes intéressés aux systèmes qui s'appuient sur des théories issues de travaux en SHS. En effet, ces théories permettent de comprendre les comportements humains et d'identifier les facteurs qui les influencent. Nous avons pu distinguer deux grandes approches que nous avons étudiées au travers de deux exemples. La première consiste à intégrer de multiples facteurs dans une même architecture, et ce pour pouvoir gérer un maximum de situations et rendre compte de la complexité du comportement humain. Cependant nous avons constaté que cette approche ne permet pas d'assurer la validité du modèle produit (e.g. opérationnalisation de nombreuses théories issues de cadres théoriques différents) et le rend difficile à manipuler et à évaluer. La seconde approche consiste à s'intéresser à un petit nombre de facteurs en s'appuyant sur un cadre théorique fort, ce qui assure l'utilisabilité du système et l'explicabilité des comportements générés. Des travaux tels que ceux de Marsella and Gratch (2009) montrent de plus que les comportements générés n'en sont pas moins complexes.

Nous nous intéressons dans cette thèse à la simulation du comportement de petites équipes d'agents virtuels. Nous sommes particulièrement intéressés par l'influence des relations interpersonnelles sur le comportement de chacun : comment la prise en compte des autres influence-t-elle notre propre comportement ? Dans quelle situation va-t-on être enclin à aider quelqu'un d'autre dans sa tâche ? Dans quelle mesure va-t-on privilégier des buts liés à l'équipe plutôt que nos buts personnels ? Nous choisissons pour répondre à ces questions de nous appuyer sur les relations de confiance. La confiance est en effet capitale : elle est considérée par (Castelfranchi and Falcone, 2010) comme la glu de la société. Dans un contexte de travail en équipe, un climat de confiance permet notamment une répartition plus souple des tâches, un échange plus libre d'informations lors des communications et un meilleur investissement de chacun des membres de l'équipe (Jones and George, 1998). Mais surtout, la confiance influence toutes nos décisions dans la vie de tous les jours (Gambetta, 2000). Dans le cadre de nos travaux, les personnages virtuels doivent en effet tenir compte de questions comme *Ai-je confiance en la compétence de mon coéquipier ?* avant de prendre une décision. La confiance est donc capitale et c'est le facteur que nous choisissons d'étudier dans le cadre de nos travaux sur la génération des comportements

d'une petite équipe de personnages virtuels.

Nous abordons donc dans le chapitre 3 les différents environnements virtuels dans lesquels la confiance est utilisée ainsi que les SMA qui reposent sur un modèle de confiance. Nous consacrerons ensuite le chapitre 4 à l'étude de la confiance dans le domaine des SHS, ce qui sera l'occasion de développer les éléments liés à l'influence de la confiance sur le comportement ainsi que les différentes conceptualisations de la confiance.

Chapitre 3

Utilisation de la confiance en informatique pour la génération de comportements de personnages virtuels et dans le domaine des systèmes multi-agents

Nous pensons que la confiance est un facteur indispensable à prendre en compte pour la génération des comportements d'agents virtuels travaillant sur des activités collaboratives. Nous verrons notamment dans le chapitre 4 que la confiance influence les comportements individuels et donc la performance des groupes.

Puisque nous souhaitons prendre en compte ce facteur de la confiance pour la génération des comportements de nos agents, nous étudions dans la première partie de ce chapitre les différentes utilisations de la confiance pour la génération des comportements d'agents virtuels.

La confiance étant particulièrement étudiée dans le domaine des systèmes multi-agents (SMA), nous dépeignons dans la seconde partie de ce chapitre l'état de l'art sur les systèmes de confiance dans ce domaine.

3.1 Personnages virtuels et confiance

La notion de confiance a été utilisée dans certains environnements virtuels, et ce principalement de deux manières différentes, répondant à des objectifs différents. Elle a d'une part été utilisée comme une variable d'évaluation du comportement des personnages virtuels, dans le sens où le personnage doit créer, par ses comportements, un climat de confiance avec l'utilisateur. Elle a d'autre part été utilisée comme variable sur laquelle s'appuie la décision du personnage virtuel. Nous présentons ces deux types d'usage de la confiance dans les sections suivantes.

3.1.1 Climat de confiance

Dans les environnements virtuels, la confiance a été une variable d'intérêt en particulier au regard de la génération du comportement des ACA – Agents Conversationnels Animés. Les ACA peuvent être utilisés dans des environnements virtuels de formation, comme c'est le cas par exemple dans le projet TARDIS (Anderson et al., 2013) pour l'entraînement à l'entretien d'embauche, mais peuvent aussi être utilisés dans d'autres contextes. Par exemple, les ACA Ada et Grace sont agents d'accueil au Boston Museum of Science (Swartout et al., 2010).

L'un des enjeux de l'utilisation des ACA est de bâtir une relation de confiance entre le personnage virtuel et l'utilisateur humain avec lequel il dialogue. L'objectif est alors d'identifier les éléments qui favoriseront cette confiance. Ces éléments peuvent concerner l'apparence ou l'animation des personnages virtuels, mais aussi des stratégies comportementales qui permettraient d'établir un climat de confiance.

Apparence et animation

De nombreux travaux se sont intéressés aux éléments de l'apparence et/ou de l'animation des personnages virtuels qui peuvent les faire paraître, par exemple, plus sympathiques, plus crédibles, plus compétents ou plus dignes de confiance. Nunamaker et al. (2011) montrent notamment qu'un ACA de sexe masculin est perçu comme plus digne de confiance lors d'un dialogue avec l'utilisateur qu'un ACA de sexe féminin.

Dans leurs travaux sur la négociation humain – humain virtuel, Antos et al. (2011) et de Melo et al. (2013) montrent que l'on a plus tendance à faire confiance et à coopérer avec un personnage virtuel qui exprime des émotions qu'avec un personnage qui n'en n'exprime pas. De manière similaire, Cowell and Stanney (2005) s'intéressent à l'animation des ACA et étudient l'influence des comportements non-verbaux de l'ACA sur la confiance qu'il inspire au sujet avec lequel il interagit, au cours d'une séance de travail pendant laquelle le sujet et l'ACA doivent trier des photos ensemble. Ainsi, les auteurs montrent qu'un ACA qui sourit, qui évite de froncer les sourcils, et qui maintient le contact visuel avec le sujet est perçu comme étant plus crédible et plus digne de confiance qu'un agent qui ne le fait pas. Les auteurs montrent par ailleurs que les sujets se sentent plus compétents et font moins d'erreurs lorsqu'ils interagissent avec un agent ayant de tels comportements.

Des études similaires sont également faites sur des robots, notamment par Si and McDaniel (2016).

Stratégies comportementales

D'autres travaux se sont intéressés à des stratégies comportementales plus générales pour permettre à un ACA de construire une relation de confiance avec l'utilisateur. Bickmore and Cassell (2001) et Bickmore and Cassell (2005) par exemple proposent Rea – *Real Estate Agent*, un ACA agent immobilier qui interroge l'utilisateur sur le type de bien immobilier qu'il souhaite acquérir. Les auteurs proposent la stratégie du *small talk*¹ pour établir un climat de confiance

1. "small talk" signifie bavarder, parler de tout et de rien.

entre Rea et le sujet. L'hypothèse est la suivante : le small talk permet à Rea d'agir sur trois dimensions de sa relation avec l'utilisateur :

- Le small talk devrait augmenter l'appréciation de Rea par l'utilisateur ; c'est la dimension de l'*affect*.
- Le small talk devrait permettre à l'utilisateur de se sentir plus à l'aise pour partager des informations plus intimes avec Rea ; c'est la *familiarité*.
- Enfin le small talk doit permettre à l'utilisateur de percevoir que Rea a des buts, croyances, valeurs et comportements similaires aux siens ; c'est la *solidarité*.

De plus, chacune de ces dimensions influence la confiance de l'utilisateur en Rea. Le small talk devrait donc permettre d'établir une relation / un climat de confiance entre Rea et l'utilisateur. Les résultats montrent que cette stratégie n'est efficace que sur les sujets extravertis ; les sujets introvertis ayant par défaut une forte confiance en Rea.

Dans ces travaux tout comme dans ceux portant sur l'apparence et l'animation des personnages, aucune notion de confiance n'est opérationnalisée : Rea applique la stratégie du small talk mais elle n'a pas de croyances sur la confiance qu'elle inspire au sujet (ni sur celle que le sujet lui inspire). La confiance intervient uniquement pour la définition des stratégies et lors de l'évaluation de l'interaction par l'utilisateur.

3.1.2 La confiance comme support de décision

Marsella et al. (2004) modélisent la confiance qu'a l'agent A en l'agent B, et qui sert, combinée à d'autres variables, à l'agent A pour décider de croire ou ne pas croire l'information transmise par l'agent B. La confiance de A en B est représentée comme une simple valeur numérique comprise entre 0 et 1. La confiance est dynamique : lorsque A décide de croire l'information transmise par B, la confiance de A en B augmente, alors qu'elle diminue si A décide de ne pas croire B.

Les travaux de Traum et al. (2005) portent sur l'interaction en langage naturel avec un personnage virtuel. Dans le projet SASO-ST, qui vise l'entraînement à la négociation en environnement virtuel, le sujet humain fait face à un personnage virtuel avec lequel il négocie. Tout comme dans les travaux de Marsella et al. (2004), le degré de confiance qu'a l'agent envers le sujet humain avec lequel il interagit est utilisé pour décider d'accepter ou de refuser la proposition de ce dernier. Ici cependant, la confiance est multidimensionnelle : les auteurs se sont inspirés des travaux de Bickmore and Cassell (2005) pour définir les trois dimensions qui composent la confiance :

- L'agent évalue sa *solidarité* avec le sujet, qui correspond au degré de similarité entre ses buts et ceux du sujet.
- L'agent évalue la *crédibilité* du sujet, qui concerne la cohérence des demandes du sujet par rapport aux croyances de l'agent.
- Enfin l'agent évalue sa *familiarité* avec le sujet selon le respect du sujet des normes sociales et de politesse de l'agent.

La confiance globale est ensuite calculée par combinaison de ces trois variables. Ces trois variables sont dynamiques : elles sont mises à jour à chaque interaction. La confiance est donc elle aussi dynamique. La confiance est ensuite utilisée comme l'une des variables qui influencent le choix de la stratégie de négociation à adopter par le personnage virtuel.

Dans des travaux ultérieurs Traum et al. (2008) ont étendu leur modèle de manière à permettre la négociation multipartite : la confiance n'est plus générale (i.e. les notions de relations de confiance et de climat de confiance ne sont plus similaires – contrairement aux travaux de Bickmore and Cassell (2005)) mais est dirigée spécifiquement d'un agent vers un autre. Le personnage virtuel A peut donc avoir confiance en le personnage B et pas en l'utilisateur humain. L'asymétrie de la relation de confiance est ici évidente : la confiance qu'a l'agent A en l'agent B ne correspond pas nécessairement à la confiance qu'a l'agent B en l'agent A.

Dans ces travaux, la confiance est opérationnalisée : le personnage virtuel a des croyances sur la relation de confiance qu'il entretient avec un autre personnage virtuel ou avec l'utilisateur. Cette confiance est mise à jour au cours de l'interaction et est prise en compte dans la décision de l'agent, ce que lui permet de faire des choix informés au regard de ses relations de confiance. La confiance est multidimensionnelle ; cependant le choix de ces dimensions est discutable car il ne s'appuie pas sur des travaux en sciences humaines et sociales (SHS).

3.1.3 Conclusion

Nous nous intéressons à la prise en compte de la confiance pour la génération de comportements crédibles, cohérents et représentatifs de ceux observés en situation réelle pour des agents virtuels formant une petite équipe. Dans cette optique, nous nous sommes intéressés aux travaux portant sur les personnages virtuels qui intègrent la notion de confiance.

La confiance est souvent envisagée sous l'angle du climat de confiance à établir entre l'agent virtuel et l'humain avec lequel il interagit. Plusieurs éléments peuvent favoriser ce climat de confiance : d'une part l'apparence et l'animation du personnage virtuel (e.g. sourire), et d'autre part certaines stratégies comportementales comme le small talk peuvent être utilisées par le personnage virtuel. Dans ce type d'application, la confiance n'est pas opérationnalisée (i.e. l'agent n'a pas de représentation de sa confiance en l'humain, ni de représentation de la confiance de l'humain en lui).

Dans d'autres travaux, la confiance est opérationnalisée. Par exemple Traum et al. (2008) la décrivent comme une relation tridimensionnelle et non symétrique. La confiance sert alors de support à la prise de décision du personnage autonome et est mise à jour au fil de ses interactions. Décrire la confiance selon trois dimensions permet une modélisation fine des relations de confiance entre les personnages et la génération de comportements complexes. De plus ces comportements sont explicables de par les valeurs de confiance selon chacune des dimensions. Cependant, le choix des dimensions qui permettent de décrire la confiance est critiquable car il ne s'appuie pas sur des travaux en SHS ; et les comportements produits peuvent donc difficilement être justifiés.

Dans le domaine des SMA, de nombreux travaux se sont intéressés à la confiance et l'ont opérationnalisée. Nous étudions ces systèmes dans la section suivante.

3.2 Systèmes de confiance numériques

La confiance est une notion à laquelle les SMA s'intéressent particulièrement. En effet, elle permet l'émergence de la coopération et la régularisation des échanges. Elle peut être utile par

exemple dans des systèmes de e-commerce, où un agent assistant un utilisateur doit évaluer le caractère digne de confiance des agents avec lesquels il traite.

Confiance et réputation. Dans le domaine des SMA, confiance et réputation sont souvent associées : généralement, on considère la confiance comme une évaluation d'un agent j par un agent i , et la réputation comme l'évaluation de ce même agent j par un groupe d'agents I . L'agent i du groupe I peut par exemple avoir complètement confiance en j pour lui donner une information juste, même si j a la réputation dans le groupe I de donner des informations incorrectes. Ces concepts sont donc semblables et peuvent s'influencer l'un l'autre : [Castelfranchi and Falcone \(2010\)](#) définissent la décision de faire confiance comme la décision de compter sur quelqu'un pour faire quelque chose. Des informations telles que la réputation de la personne peuvent alors être utilisées pour prendre cette décision.

On peut également souligner le fait qu'il y a deux éléments distincts qui peuvent être intégrés à un SMA utilisant les notions de confiance ou de réputation :

- (i) *Un processus de calcul des valeurs de confiance et/ou de réputation* permet de mettre à jour ces valeurs.
- (ii) *Un processus de prise de décision* à partir de ces valeurs de confiance et/ou de réputation permet à l'agent de choisir une action.

A partir de ce constat, et pour clarifier notre position par rapport aux divergences d'opinion sur les définitions des systèmes de confiance et de réputation, nous adoptons dans nos travaux une définition particulière pour désigner les systèmes de confiance qui répondent au critère (ii) :

Système décisionnel s'appuyant sur la confiance et/ou la réputation

On considère comme décisionnel s'appuyant sur la confiance un système qui identifie les processus décisionnels en jeu lors de la décision de faire confiance ; c'est-à-dire les processus décisionnels s'appuyant sur des valeurs de confiance et/ou de réputation liées à d'autres agents qui guident le choix d'une action.

Cette définition est proposée par [Pinyol and Sabater-Mir \(2013\)](#), mais les auteurs l'utilisent pour désigner ce qu'ils appellent un *système de confiance* et non un *système décisionnel s'appuyant sur la confiance*. Cependant, le terme de *système de confiance* n'étant habituellement pas employé de cette manière, nous lui préférons celui plus précis de *système décisionnel s'appuyant sur la confiance*. Cette définition répond en effet à notre problématique, puisque nous souhaitons proposer un moteur décisionnel qui permette la génération de comportements liés à la confiance pour des personnages autonomes. Selon cette définition, il est clair que certains systèmes présentés comme des systèmes de réputation peuvent être considérés comme des systèmes décisionnels s'appuyant sur la confiance (e.g. AFRAS, voir 3.2.3).

Systèmes numériques et cognitifs. La littérature sur les systèmes de confiance et/ou de réputation dans le domaine des SMA est très riche et plusieurs auteurs en proposent des revues ; notamment [Ramchurn et al. \(2004\)](#), [Sabater and Sierra \(2005\)](#), [Lu et al. \(2009\)](#), [Pinyol and Sabater-Mir \(2013\)](#), [Yu et al. \(2013\)](#) et [Granatyr et al. \(2015\)](#).

Ces différentes revues proposent des classements des systèmes de confiance et/ou réputation selon de nombreux critères (e.g. la généralité du système ([Pinyol and Sabater-Mir, 2013](#)), la

prise en compte des risques (Granatyr et al., 2015)). Dans le cadre de nos travaux, nous sommes focalisés sur un critère qui est utilisé dans la majorité des classifications existantes : *la dimension cognitive de la confiance* (Pinyol and Sabater-Mir, 2013) ou *le paradigme numérique ou cognitif* (Granatyr et al., 2015). En effet, nous verrons en 3.3 que dans les systèmes dits cognitifs, autant d'importance est donnée à la formalisation de la confiance et des processus qui mènent à la décision de faire confiance qu'à la décision elle-même, ce qui permet notamment d'expliquer la décision prise.

Nous abordons donc dans un premier temps la définition des systèmes de confiance et/ou de réputation dits numériques, et nous donnons quelques exemples de ces systèmes pour mieux pouvoir en identifier les avantages et les limites. Dans un second temps, nous donnerons une définition des systèmes cognitifs, puis nous nous attarderons sur les systèmes C&F, ForTrust et BDI+Repage, avant de conclure.

3.2.1 Définition des systèmes de confiance numériques

Ramchurn et al. (2004) ne parlent pas exactement de modèles de confiance numériques, mais de systèmes dans lesquels le niveau de confiance repose sur l'évaluation d'une interaction. La valeur de confiance d'un agent A en un agent B peut être calculée par A en fonction de l'historique de ses interactions directes avec B (i.e. combien de fois B a-t-il adopté le comportement souhaité par le passé?) ou grâce à des informations sur B reçues d'autres agents, c'est-à-dire à partir de la réputation de B.

Sabater and Sierra (2005) définissent simplement les modèles de confiance numériques comme des modèles qui utilisent des approches issues de la théorie des jeux et qui ne permettent pas une représentation explicite des éléments cognitifs qui constituent la confiance. La confiance de l'agent A en l'agent B correspond alors à la probabilité selon A que B adopte le comportement attendu. Cette définition est reprise par Granatyr et al. (2015). Pinyol and Sabater-Mir (2013) expliquent également que dans les approches numériques, le calcul de la confiance correspond à une sorte de boîte noire qui ne permet pas à l'agent d'expliquer clairement sa décision de faire confiance ou de ne pas faire confiance.

Il existe de très nombreux systèmes de confiance et/ou de réputation numériques (pour une énumération plus complète de ces systèmes, voir notamment les revues de Pinyol and Sabater-Mir (2013), Yu et al. (2013) et Granatyr et al. (2015)), et ceux-ci sont principalement utilisés dans des applications de e-commerce où chaque agent a besoin d'évaluer le niveau de confiance qu'il a en l'agent avec lequel il interagit. Par exemple, un acheteur cherche à savoir si un vendeur est fiable et digne de confiance. Parmi les systèmes numériques les plus connus, on peut citer par exemple les travaux de Marsh (1994), le système REGRET (Sabater and Sierra, 2001a) ou le système LIAR (Muller and Vercouter, 2005).

3.2.2 Système de confiance de Marsh (1994)

Marsh (1994) est l'un des premiers à avoir proposé une formalisation de la confiance. L'auteur distingue trois types de confiance :

- La confiance basique T_x de l'agent x correspond à la disposition de x à faire confiance.
- La confiance générale de l'agent x en l'agent y : $\widehat{T_x(y)}$ est une estimation de cette confiance

qui tient compte des expériences passées que x a eues avec y . L’auteur propose trois manières pour calculer $\widehat{T_x(y)}$, qui représentent des tendances optimistes, pessimistes et réalistes de x .

— La confiance situationnelle $T_x(y, \alpha)$ représente la confiance de x en y dans la situation α .

Dans le cadre d’une interaction directe entre les agents x et y , Marsh propose la formule 3.1 qui permet de calculer la confiance situationnelle :

$$T_x(y, \alpha) = U_x(\alpha) \times I_x(\alpha) \times \widehat{T_x(y)} \quad (3.1)$$

où $U_x(\alpha)$ représente l’utilité pour x de la situation (mesure rationnelle) et $I_x(\alpha)$ représente l’importance pour x de la situation (mesure irrationnelle).

L’agent x décide de faire confiance à l’agent y dans une situation α si et seulement si la confiance situationnelle $T_x(y, \alpha)$ est supérieure au seuil de coopération $\text{Cooperation_Threshold}_x(\alpha)$. La valeur du seuil de coopération est donnée par la formule 3.2 :

$$\text{Cooperation_Threshold}_x(\alpha) = \frac{\text{Perceived_Risk}_x(\alpha)}{\text{Perceived_Competence}_x(y, \alpha) + \widehat{T_x(y)}} \times I_x(\alpha) . \quad (3.2)$$

Dans cette formule, la compétence $\text{Perceived_Competence}_x(y, \alpha)$ de y pour gérer la situation α telle que perçue par x est calculée à partir des expériences passées de x avec y , ou à partir de la confiance basique de x si x ne connaît pas y . Le risque $\text{Perceived_Risk}_x(\alpha)$ lié à la situation α tel que perçu par x est défini de manière *ad hoc* à partir du scénario étudié. Les mécanismes proposés permettent à l’agent x de décider de faire confiance ou de ne pas faire confiance à y : ce système répond donc à la définition d’un système décisionnel s’appuyant sur la confiance que nous avons adoptée.

L’auteur propose également des mécanismes de mise à jour de la confiance basique et de la confiance générale ; cependant seules les interactions directes sont prises en compte pour la mise à jour de ces valeurs et les mécanismes proposés sont spécifiques aux situations type dilemme du prisonnier.

3.2.3 AFRAS

AFRAS – *A Fuzzy Reputation Agent System* (Carbo et al., 2003) est un système de réputation numérique : les agents utilisent des informations relatives à la réputation des autres pour décider avec qui interagir et éviter les déconvenues dans une application de e-commerce. Pour calculer la réputation d’un agent B, l’agent A prend en compte ses observations directes (i.e. ses propres évaluations de B suite aux interactions passées) et des informations fournies par des tiers. Dans le cas d’une information fournie par un tiers, la réputation de ce tiers est prise en compte pour décider du poids à accorder à l’information fournie.

Cependant, dans le système AFRAS, la réputation n’est pas contextualisée, c’est-à-dire qu’un agent A a une seule valeur de réputation pour un autre agent B. De ce fait, l’agent A ne peut pas distinguer plusieurs aspects de sa relation avec l’agent B : l’agent B peut à la fois être un mauvais fournisseur de service, et bon fournisseur d’information sur les autres agents, mais A ne fait pas de différence. De plus les auteurs ne précisent pas les mécanismes de prise de décision en fonction de la valeur de réputation calculée ; ce système n’est donc pas un système décisionnel s’appuyant sur la confiance.

3.2.4 REGRET

Le système REGRET (Sabater and Sierra, 2001a,b) permet à chaque agent de faire un choix informé sur l'agent auquel acheter un service en s'appuyant sur un calcul complexe de réputation.

Après chaque interaction, les agents notent et mémorisent leur niveau de satisfaction en leur partenaire. Lorsqu'un agent A est confronté à la possibilité d'interagir avec un agent B, A calcule une valeur de réputation pour B. Contrairement au système de Marsh (1994), ce calcul prend en compte différentes sources d'information :

- La *dimension individuelle de la réputation* reflète l'opinion que A s'est faite de B au cours de ses interactions directes avec B, dont le niveau de satisfaction a été mémorisé.
- La *dimension sociale de la réputation* reflète l'opinion que A s'est faite de B au cours d'interactions indirectes. Trois types d'interactions indirectes sont à distinguer : (i) A a une opinion sur le groupe auquel B appartient, qui influence son opinion de B. (ii) A prend en compte l'opinion sur B du groupe auquel lui-même appartient. Enfin, (iii) A prend en compte l'opinion du groupe auquel il appartient sur le groupe auquel B appartient.

De plus une valeur de fiabilité est associée à chacune de ces informations, ce qui permet à l'agent d'exprimer un degré de certitude différent pour chaque information selon la source. Par exemple, une information dont on est sûr à 100% (e.g. on l'a observée directement) pourra être traitée différemment d'une information dont on n'est sûr qu'à 25%.

De manière similaire à Marsh (1994), Sabater and Sierra (2001a) contextualisent la réputation des agents (i.e. B peut avoir une bonne réputation en tant qu'agence de voyage et une mauvaise réputation comme cuisinier). Par exemple, lorsque l'on évalue une agence de voyage, on prend en compte la qualité des hôtels proposés, leur localisation, l'intérêt du parcours, la compétence du guide, etc. Les auteurs proposent donc d'utiliser une ontologie de réputation qui décrit la manière dont sont combinés les différents éléments et le poids de chacun.

Bien que les auteurs utilisent le terme de réputation, le système proposé permet aux agents de calculer une valeur de confiance globale en les autres, associée à une valeur de fiabilité. Cependant, le mécanisme de prise de décision à partir de ces valeurs n'est pas défini : faut-il interagir en priorité avec les agents pour lesquels notre valeur de confiance est élevée, même si cette valeur n'est pas très fiable ? Ou faut-il privilégier les interactions avec un agent en qui on a un peu moins confiance, mais pour lequel on est sûr de notre évaluation ? Ce système n'est donc pas un système décisionnel s'appuyant sur la confiance.

Une approche similaire est adoptée dans le système FIRE (Huynh et al., 2006) : la confiance est également contextualisée ; et les auteurs précisent les mécanismes qui permettent de calculer une valeur de confiance globale associée à une valeur de fiabilité, mais ne donnent pas les mécanismes qui permettent de prendre une décision par rapport à ces valeurs. Tout comme REGRET, les agents FIRE utilisent différentes sources d'information. Les auteurs proposent notamment d'utiliser l'agent B lui-même comme source d'information pour l'agent A : l'agent B peut faire parvenir à A des références, à la manière de lettres de recommandations, obtenues lors d'interactions passées. D'après les auteurs, la prise en compte de ce type d'information permet de réduire grandement le nombre de cas où l'agent A est incapable de trouver une information sur B qui lui permette de calculer une valeur de confiance.

3.2.5 LIAR

Le système LIAR (Muller and Vercouter, 2005) se penche sur le problème des agents tricheurs qui mentent lorsqu'ils communiquent des informations à d'autres dans le cadre des réseaux peer-to-peer. Un protocole de détection de mensonge est défini, qui permet à un agent A de repérer les incohérences dans les informations transmises par d'autres agents. Lorsque A repère une incohérence entre des informations transmises par B et C, il peut se servir de ses valeurs de confiance en B et C afin de déterminer qui ment. Tout comme dans REGRET et FIRE, les agents LIAR considèrent différentes sources pour calculer la confiance, qui sont : (1) les observations directes, (2) les informations transmises par des tiers² et (3) une valeur de confiance par défaut. Le processus de calcul de ces valeurs n'est pas donné par les auteurs.

Les auteurs précisent en revanche le processus de prise de décision à partir de ces informations : A décide de faire confiance ou de ne pas faire confiance respectivement si la valeur de confiance calculée à partir de (1) est supérieure à θ_{trust} ou inférieure à $-\theta_{trust}$, θ_{trust} étant un seuil de confiance défini par le concepteur du système. Si cette valeur est comprise entre θ_{trust} et $-\theta_{trust}$, alors A applique le même raisonnement sur les valeurs de confiance issues de (2). Enfin si cela ne permet toujours pas à A de faire un choix, le même raisonnement est appliqué sur la valeur de confiance (3) et un seuil égal à 0.

Ce système est donc un système décisionnel s'appuyant sur la confiance selon la définition que nous avons choisie ; et le problème traité est original puisque la problématique des agents menteurs est souvent ignorée dans les systèmes de confiance et/ou réputation (c'est le cas par exemple dans REGRET ou FIRE).

3.2.6 Conclusion

Les systèmes de confiance et/ou réputation numériques sont extrêmement nombreux : dans leur revue, Granatyr et al. (2015) en comptent 91 dans les travaux publiés au cours de ces 20 dernières années, ce qui représente 85% des systèmes étudiés. Ces systèmes sont créés pour des applications très diverses, telles que les réseaux peer-to-peer, le e-commerce ou les systèmes de recommandation.

Nous cherchons à proposer un moteur décisionnel qui s'appuie sur des relations de confiance pour des personnages virtuels. Nous nous intéressons dans ce cadre aux systèmes décisionnels s'appuyant sur la confiance tels que LIAR ou le système de Marsh (1994). Toutefois une étude plus large de la littérature nous permet d'identifier des éléments d'intérêt.

Nous avons vu avec le contre-exemple des agents AFRAS qui ne tiennent compte que d'une seule valeur de réputation (i.e. la réputation est décontextualisée) qu'il est important d'avoir une confiance qui soit contextualisée, de manière à ce qu'un agent A puisse évaluer différemment sa confiance en un agent B selon la situation considérée. La confiance est par ailleurs contextualisée dans les autres systèmes présentés. Nous avons aussi pu constater notamment avec le système REGRET qu'il est important d'associer une valeur de fiabilité à chaque élément considéré pour calculer la confiance de manière à pouvoir donner plus de poids aux informations dont on est

2. Deux valeurs de réputation différentes sont calculées à partir d'informations indirectes, selon que ces informations portent sur (i) la simple répétition à A d'un message de B par un tiers ou (ii) la transmission à A d'une évaluation sur B par un tiers.

sûr. Le système de Marsh (1994) ne prend pas en compte ce paramètre. Enfin, pour le calcul des valeurs de confiance et/ou de réputation, nous avons remarqué qu'il est possible de prendre en compte différentes sources d'information, comme c'est le cas dans les systèmes AFRAS, REGRET et LIAR. Il faut alors se poser la question des agents menteurs : bien que ce problème soit ignoré dans la plupart des systèmes, des mécanismes comme ceux proposés dans LIAR peuvent permettre de démasquer les agents menteurs.

Nous ne nous intéresserons pas, dans le cadre de cette thèse, au processus de calcul et de mise à jour des valeurs de confiance, car nous nous concentrons dans un premier temps sur la proposition d'un moteur décisionnel. Cependant, une amélioration possible à nos travaux serait l'intégration de tels mécanismes, et il est alors intéressant de savoir que de tels mécanismes ont déjà été proposés. Ces mécanismes permettent notamment de prendre en compte différentes sources d'information ou de repérer les agents menteurs, et pourraient être intégrés à notre système (voir 8.2.2).

Enfin, les principales limites des systèmes de confiance et/ou réputation numériques sont les suivantes : ils sont souvent centrés sur un contexte particulier (i.e. manque de généralité) ; et les mécanismes liés à la décision de faire confiance sont rarement inspirés d'études en SHS. En effet, l'objectif dans ces systèmes est avant tout la performance, et les comportements générés ne sont pas nécessairement représentatifs des comportements humains.

3.3 Systèmes de confiance cognitifs

Les définitions des systèmes cognitifs varient selon les auteurs. Nous développons les points de vue de plusieurs auteurs de revues faisant intervenir ce critère, et nous clarifions notre position par rapport à la confusion qui règne dans la littérature.

Plusieurs définitions d'un système cognitif

Yu et al. (2013) ne parlent pas de systèmes cognitifs utilisant la confiance et/ou la réputation, mais de systèmes sociocognitifs ; et leur définition est la moins restrictive que l'on puisse trouver dans les différentes revues. Selon ces auteurs, dans un système de confiance sociocognitif, la confiance d'un agent A en un agent B s'appuie sur l'évaluation par A de propriétés de B et de facteurs extérieurs qui pourraient influencer le comportement de B. Par exemple, le système de confiance de Noorian et al. (2011) répond à la définition d'un système de confiance sociocognitif selon les critères de Yu et al. (2013) : la décision par A de croire en l'information transmise par B (i.e. faire confiance à B) repose d'une part sur une évaluation de l'expérience de B et d'autre part sur l'évaluation de la crédibilité de B. De plus, la confiance de A en B est modulée par une propriété propre à A qui vise à introduire un biais humain dans son jugement : A peut être optimiste, réaliste ou pessimiste.

Les autres définitions de systèmes cognitifs sont plus précises et plus restrictives. Ramchurn et al. (2004) considèrent un système de confiance comme cognitif si la confiance de l'agent A en l'agent B s'appuie sur une évaluation subjective de certaines caractéristiques de B par A. La confiance de A en B est alors représentée par un état mental complexe composé d'un ensemble de croyances de A sur des caractéristiques de B. Ces croyances peuvent porter par exemple sur les capacités de B ou sur le caractère fiable de B. Pinyol and Sabater-Mir (2013) éclaircissent

cette définition et établissent que, dans un système de confiance cognitif, les agents ont une représentation claire de la confiance en termes d'éléments cognitifs tels que des croyances et des désirs. Ces éléments leur permettent d'expliquer leur décision et de raisonner à propos de la confiance. Les auteurs mettent en avant le fait que dans ces modèles, la définition des éléments cognitifs sur lesquels s'appuie la décision de faire confiance est aussi importante que la décision elle-même. Cette définition est reprise par Granatyr et al. (2015).

Cette notion d'état mental complexe composé d'éléments cognitifs n'est pas présente dans la définition de Yu et al. (2013), et en l'introduisant dans la définition, on restreint le nombre de systèmes de confiance qui peuvent répondre à ces critères. Dans les travaux de Noorian et al. (2011) par exemple, la confiance n'est pas formalisée en termes d'éléments cognitifs.

Plusieurs interprétations de la définition d'un système de confiance cognitif

Par ailleurs, même lorsque les définitions de référence sont identiques, les classifications établies par les auteurs des revues diffèrent : un système classé comme cognitif par un auteur peut être classé comme non cognitif par un autre, tout en utilisant la même définition d'un système de confiance cognitif.

Par exemple, AFRAS (voir 3.2.3) est classé comme système cognitif par Granatyr et al. (2015) alors qu'il est classé comme modèle numérique par Sabater and Sierra (2005) et Pinyol and Sabater-Mir (2013). Cette confusion peut s'expliquer par le fait que les agents AFRAS possèdent des éléments qui sont généralement associés à la cognition : ils ont un *désir* de solitude et ils ont un composant appelé *modèle mental*. Cependant les éléments liés à la confiance ne sont pas exprimés en termes d'éléments cognitifs dans ce modèle. Selon l'interprétation stricte de la définition de Pinyol and Sabater-Mir (2013), ce système est un système de confiance numérique.

Nous adoptons dans nos travaux la définition suivante d'un système de confiance cognitif ainsi que son interprétation stricte :

Système de confiance cognitif

Un système de confiance cognitif est un système de confiance dans lequel les agents ont une représentation claire de la confiance qu'ils ont en les autres, en termes d'éléments cognitifs qui composent cette confiance, tels que les désirs et les croyances. Une telle représentation leur permet de raisonner à propos de la confiance et d'expliquer leur décision (Pinyol and Sabater-Mir, 2013).

Nous présentons dans les sections suivantes les principaux travaux sur les systèmes de confiance cognitifs dans le domaine des SMA : BDI+Repage, le modèle de Castelfranchi and Falcone (2010), noté dans la suite de ce manuscrit *modèle C&F* et ForTrust.

3.3.1 Repage et BDI+Repage

Sabater et al. (2006) proposent le système Repage – *REPutation and imAGE among limited autonomous partners* dans lequel est définie une formalisation des concepts de réputation et d'image. Repage a ensuite été étendu à BDI+Repage dans lequel les mécanismes décisionnels sont formalisés en BDI (Pinyol et al., 2012). Ces travaux s'appuient sur les recherches en sciences

sociales et cognitives de (Conte and Paolucci, 2002, cité par Sabater et al. (2006)) qui proposent que la réputation d'un agent soit évaluée par rapport à une attitude sociale idéale.

Repage

Dans Repage, la décision de l'agent i de faire confiance à l'agent j s'appuie sur la réputation de j et sur l'image qu'a i de j . L'image de j correspond à une évaluation de j partagée par tous les agents i d'un groupe I : i pense que j est un bon cuisinier. La réputation de j dans le groupe d'agent I est une méta-croyance : i pense que le groupe I pense que j est un cuisinier moyen. Dans le système Repage, l'image et la réputation sont contextualisées par rapport à un rôle joué par l'agent j : j peut donc avoir une image de mauvais cuisinier et une image de bon fournisseur de vin. Pour un même rôle, plusieurs critères peuvent être évalués. Par exemple, pour le rôle de fournisseur, on peut évaluer le temps de livraison, la qualité du produit, le prix, etc. L'image et la réputation d'un agent *cible* jouant un rôle *role* et évalué sur le critère *critere* sont alors définies respectivement par $Img(cible, role(critere), contenu)$ et par $Rep(cible, role(critere), contenu)$.

Pour pouvoir exprimer un degré d'incertitude sur la réputation ou l'image d'un agent, Repage utilise des ensembles flous pour décrire le *contenu* de l'évaluation. On a alors

$$contenu = [w_1, w_2, w_3, w_4, w_5] \quad \text{avec } w_i \in [0; 1] \quad \text{et} \quad \sum w_i = 1$$

où w_i représente le poids attribué au fait que j soit respectivement *très mauvais*, *mauvais*, *moyen*, *bon*, *très bon* sur le *critere* du *role*. Par exemple,

$$img_1 = Img(j, seller(quality), [0; 0; 0.1; 0.2; 0.7])$$

exprime le fait que j a l'image de fournir (*role* = *seller*) un vin plutôt de très bonne qualité (*critere* = *quality*), alors que

$$img_2 = Img(j, seller(dTime), [0.2; 0.2; 0.2; 0.2; 0.2])$$

exprime le fait que l'on n'a aucune information sur le temps de livraison du fournisseur j (ici *critere* = *dTime* pour *delivery time*, temps de livraison). L'image est mise à jour à la fois lors des interactions directes entre i et j et lors de la communication à i d'informations sur l'image de j par un tiers parti, alors que la réputation de j n'est mise à jour que lors de la communication d'informations sur la réputation de j par un tiers parti. La distinction entre les deux concepts est donc toujours maintenue.

Repage utilise ainsi des notions liées à la cognition (i.e. croyances), mais ne propose pas de formalisation en termes de croyances, désirs et intentions. Ce système n'est donc pas considéré comme un système cognitif par Pinyol and Sabater-Mir (2013). Le système BDI+Repage propose une telle formalisation.

BDI+Repage

BDI+Repage (Pinyol et al., 2012) est une formalisation BDI du système de confiance s'appuyant sur les concepts de réputation et d'image définis dans Repage. Dans BDI+Repage, les mécanismes décisionnels qui permettent à un agent de choisir avec quel autre agent traiter sont formalisés ; le système est donc classé comme un système de confiance par Pinyol and Sabater-Mir (2013).

Système multi-contextes. BDI+Repage est un système multi-contextes. Chaque agent possède des croyances, désirs et intentions qui sont exprimés dans des contextes spécifiques dans différentes logiques. Les quatre contextes principaux sont les suivants (nous expliquons dans les paragraphes suivants comment sont faites les liaisons entre les différents contextes) :

- Le **Repage Context** – **RC** contient les informations de Repage relatives à l'image (prédicat *Img*) et à la réputation (prédicat *Rep*).
- Les croyances de l'agent sont contenues dans le **Belief Context** – **BC**. Une croyance personnelle à l'agent s'exprime par le prédicat *B*. Par exemple,

$$B(\text{buy}(j), \text{OKProduct}, 0.1, \text{seller}(\text{quality}))$$

exprime la croyance que l'action $\text{buy}(j) = \text{acheter un produit à } j$ aura pour résultat, si l'on évalue j sur le rôle *seller* et selon le critère *quality* ($\text{seller}(\text{quality})$) d'obtenir un produit de qualité moyenne (*OKProduct*) avec une probabilité de 10%. BC contient également des prédicats *E* et *S* pour exprimer respectivement les croyances de l'agent sur l'image et sur la réputation des autres. Ces prédicats ont la même forme :

$$E/S(\text{action}, \text{resultat}, \text{proba}, \text{role}(\text{critere})) \quad .$$

(Un exemple de prédicat *E* est donné plus loin.)

- Les désirs sont exprimés dans le **Desire Context** – **DC**. Un agent ayant le désir

$$D^+(\text{hasProduct} \wedge \text{veryGoodQuality}, 0.7)$$

aurait un niveau de satisfaction de 0.7 en obtenant un produit de très bonne qualité, alors que

$$D^-(\text{hasProduct} \wedge \text{dTime} > 10, 0.9)$$

veut dire que l'agent aurait un niveau d'insatisfaction de 0.9 si le produit était livré en plus de 10 jours.

- Enfin, les intentions contenues dans le **Intention Context** – **IC**. Par exemple

$$I_{\text{buy}(j)}(\text{hasProduct} \wedge \text{veryGoodQuality}, 0.4)$$

exprime le fait que l'agent souhaite acheter un produit à j et ainsi obtenir un produit de très bonne qualité avec une force de 0.4. L'agent choisira alors d'exécuter l'action correspondant à l'intention la plus forte.

Règles de liaison. Un ensemble de règles de liaison (*bridge rules*) permet d'inférer de nouvelles propositions dans un contexte C_i à partir d'autres contextes C_k . Par exemple, les auteurs définissent une règle de liaison qui permet de transformer un prédicat *Img* de RC en cinq prédicats *E* de BC. Un exemple de traduction serait le suivant :

$$\frac{\text{RC} : \text{Img}(j, \text{seller}(\text{quality}), [0; 0; 0.1; 0.2; 0.7])}{\begin{array}{l} \text{BC} : E(\text{buy}(j), \text{veryBadQuality}, 0, \{\text{seller}(\text{quality})\}) \\ \text{BC} : E(\text{buy}(j), \text{badQuality}, 0, \{\text{seller}(\text{quality})\}) \\ \text{BC} : E(\text{buy}(j), \text{OKQuality}, 0.1, \{\text{seller}(\text{quality})\}) \\ \text{BC} : E(\text{buy}(j), \text{goodQuality}, 0.2, \{\text{seller}(\text{quality})\}) \\ \text{BC} : E(\text{buy}(j), \text{veryGoodQuality}, 0.7, \{\text{seller}(\text{quality})\}) \end{array}}$$

Les prédicats E et S sont ensuite combinés pour former des croyances B comme dans la formule suivante :

$$E(act, res, p_E, r(c)) \wedge S(act, res, p_S, r(c)) = B(act, res, h_{r(c)}(p_E, p_S), r(c))$$

où $h_{r(c)}(p_E, p_S) = \frac{\delta_E \cdot p_E + \delta_S \cdot p_S}{\delta_E + \delta_S}$ est une moyenne pondérée. Les poids δ_E et δ_S sont définis par le concepteur du système, et il est possible de définir différentes fonctions $h_{r(c)}$ (i.e. différents poids) pour chaque critère de chaque rôle. Cela permet, d'après les auteurs, d'exprimer le fait qu'un agent peut, selon le rôle et le critère évalués, donner plus d'importance à ses propres croyances (i.e. image) qu'aux croyances du groupe (i.e. réputation), ce qui a du sens notamment dans le cas où l'agent sait qu'il n'est pas très bon pour évaluer la qualité d'un produit reçu.

Génération des intentions et décision. Les désirs de l'agent sont également renseignés par le concepteur du système, et l'agent raisonne alors sur ses désirs et ses croyances pour générer ses intentions. Les règles de liaison pour la génération des intentions des agents permettent de prendre en compte tous les effets attendus négatifs et positifs associés à une même action. Autrement dit, si l'agent a les désirs

$$D^+(hasProduct \wedge veryGoodQuality, 0.8) \text{ et } D^-(hasProduct \wedge dTime > 10, 0.9)$$

et pense que ces deux résultats peuvent être réalisés par l'action $buy(j)$ avec une probabilité de 0.9, c'est-à-dire que

$$B(buy(j), veryGoodQuality \wedge dTime > 10, 0.9, \{seller(quality), seller(dTime)\}) \quad ;$$

alors l'insatisfaction engendrée par le fait de recevoir le produit dans plus de 10 jours est plus grande que la satisfaction de recevoir un produit de très bonne qualité. L'agent n'aura donc pas l'intention d'acheter le produit à j .

À l'inverse, si l'agent accorde plus d'importance à la qualité du produit qu'au temps de livraison

$$(D^+(hasProduct \wedge veryGoodQuality, 0.9) \text{ et } D^-(hasProduct \wedge dTime > 10, 0.8))$$

et a les mêmes croyances, l'intention générée sera

$$I_{buy(j)}(hasProduct \wedge veryGoodQuality \wedge dTime > 10, 0.9) \quad .$$

L'agent sélectionnera ensuite l'action qui correspond à l'intention ayant la plus forte valeur, c'est-à-dire que l'agent exécutera l'action qui aura le plus d'effets positifs par rapport aux effets négatifs.

Avantages et limites

Dans Repage, Sabater et al. (2006) formalisent les concepts d'image et de réputation. Leur proposition d'utiliser des ensembles flous pour représenter l'image et la réputation des agents est intéressante puisqu'elle permet d'intégrer de manière transparente la notion d'incertitude des agents : on peut comparer l'utilisation de ces ensembles flous dans Repage à celle des croyances

graduées dans ForTrust (voir 3.3.3). De plus, les mécanismes de combinaison de ces ensembles, notamment utilisés pour la mise à jour de l'image et de la réputation, sont également formalisés. Les mécanismes de prise de décision à partir des connaissances sur l'image et sur la réputation ne sont cependant pas décrits dans ces travaux.

Dans des travaux ultérieurs, Repage devient BDI+Repage (Pinyol and Sabater-Mir, 2009; Pinyol et al., 2012). Les auteurs formalisent les notions d'image et de réputation définies dans Repage en termes d'éléments BDI. Cette formalisation permet aux auteurs de spécifier les mécanismes de prise de décision à partir des désirs des agents et de leurs croyances sur l'image et la réputation des autres : c'est un système décisionnel s'appuyant sur la confiance et la réputation, comme défini page 31.

Néanmoins, deux critiques peuvent être faites à ces travaux : d'une part, le système proposé est uniquement centré sur la notion de délégation. Dans le cas où l'on ne peut pas faire soi-même l'action (e.g. produire du vin) qui permettra d'atteindre son but (e.g. avoir du vin), on cherche la personne la plus digne de confiance à qui déléguer cette action (e.g. on achète du vin que quelqu'un a produit). Aucun mécanisme n'est précisé qui permettrait à l'agent B d'accepter ou de refuser la demande d'un agent A qui souhaite lui déléguer une action. De même, il est impossible ici pour un agent j de proposer son aide à l'agent i : l'adoption altruiste de but (voir 4.3.1) n'est pas considérée. Or ces notions sont importantes à prendre en compte dans le cadre de nos travaux : en effet, dans les activités collaboratives, il est souhaitable que les agents soient capables d'anticiper les besoins des autres pour proposer une aide³.

D'autre part, si les auteurs proposent de contextualiser l'image et la réputation qu'un agent A a d'un agent B en les jugeant par rapport à un rôle et un critère, tous ces éléments semblent être liés à la compétence de l'agent B. Pour un même rôle et un même critère, les auteurs ne distinguent pas différents éléments qui constituent le caractère digne de confiance de B. Par exemple, l'agent B pourrait avoir très envie de répondre au besoin de l'agent A, mais ne pas en avoir la compétence. Ou à l'inverse, l'agent B pourrait être très compétent pour ce que l'agent A attend de lui, mais ne pas du tout avoir envie d'aider l'agent A. Ces éléments sont pris en compte dans le modèle de confiance sociocognitive de Castelfranchi and Falcone (2010) présenté dans la section 3.3.2 et dans la formalisation ForTrust que nous abordons section 3.3.3.

3.3.2 Modèle C&F de la confiance sociocognitive

Castelfranchi and Falcone (2010) proposent un modèle computationnel très riche de la confiance sociocognitive issu d'études à la fois en SHS (sciences cognitives et sciences du comportement) et en intelligence artificielle. Les auteurs discutent en effet de très nombreux aspects de la confiance, et nous aborderons notamment leurs différentes contributions sur la conceptualisation et la clarification de ce concept au cours du chapitre 4. Nous présentons ici uniquement les points principaux du modèle C&F.

3. Nous souhaitons former un apprenant à être attentif à ses coéquipiers dans des activités collectives. Dans de "bonnes" équipes, les agents doivent pouvoir anticiper les besoins des autres et de l'apprenant et éventuellement proposer une aide. Nous devons également pouvoir, à l'inverse, confronter l'apprenant à des équipes dans lesquelles les agents n'anticipent pas les besoins des autres.

Ingrédients de la confiance

Les auteurs proposent que la confiance est toujours située par rapport à un but : l'agent i a confiance en l'agent j pour l'aider à atteindre son but φ . L'agent i doit alors évaluer la *capacité* de j de faire une action α qui permettra de réaliser son but φ , ainsi que la *volonté* de j à réaliser l'action α . Pour chacun de ces deux éléments, de nombreux facteurs sont à prendre en compte : la capacité de j à réaliser une action α dépend par exemple de ses compétences et savoir-faire, ainsi que de sa confiance en lui ; la volonté de j de réaliser α dépend entre autres de ses motivations (données par ses buts), de sa moralité et de la pression d'une autorité extérieure. Les dimensions de capacité et de volonté composent le noyau de la confiance (*core trust*).

L'agent i a également d'autres éléments à prendre en compte, tels que l'*opportunité* que j aura de réaliser l'action α , et les obstacles qui peuvent se présenter (dangers). Enfin, pour que i ait une raison de faire confiance à j , i doit penser qu'il est *dépendant* de j ; c'est-à-dire que i doit penser que l'aide de j est nécessaire pour réaliser son but φ , ou à minima que l'aide de j serait appréciable, i.e. le but serait plus facilement atteint ou le résultat serait meilleur.

Ces "ingrédients" de la confiance sont combinés pour calculer une valeur de confiance globale : selon la situation considérée, le poids accordé à chacun de ces facteurs peut varier. Par exemple dans certaines situations, i accordera beaucoup d'importance à la volonté de j alors que dans d'autres il accordera plus d'importance à la capacité de j .

Formalisation

Tous ces ingrédients de la confiance sont formalisés en termes d'éléments cognitifs : la confiance de i en j pour la réalisation de son but (*désir*) φ est donnée par les *croyances* de i sur la capacité, la motivation et l'opportunité de j de faire α pour réaliser φ , et sur sa dépendance à j pour faire α .

Chacune de ces croyances est alors décrite par :

- une valeur numérique correspondant à l'évaluation par i de l'élément en question ;
- une seconde valeur numérique décrivant le degré de certitude ou de force associé à la croyance en question (les auteurs parlent de méta-croyance).

La valeur de confiance globale calculée à partir de ces croyances est notée $DoT_{ij\alpha}$, et cette valeur est comprise entre 0 et 1.

Décision de faire confiance

D'après [Castelfranchi and Falcone \(2010\)](#), la décision de faire confiance ou de ne pas faire confiance correspond à la décision de déléguer ou de ne pas déléguer l'action α . Pour prendre sa décision, l'agent i évalue l'utilité de chaque situation-résultat possible :

- (1) L'agent i évalue l'utilité $U(i)_{p+}$ de faire α lui-même, et que α réussisse (φ est atteint). L'utilité $U(i)_{p+}$ est donnée par la valeur du but φ et par le coût pour i de faire l'action α .
- (2) L'agent i évalue l'utilité $U(i)_{p-}$ de faire α lui-même, et que α échoue (φ n'est pas atteint). L'utilité $U(i)_{p-}$ est donnée par le coût pour i de faire l'action α et un coût additionnel lié à l'échec de α .

(3) L'agent i évalue l'utilité $U(i)_{d+}$ de déléguer α à j , et que α réussisse. L'utilité $U(i)_{d+}$ est donnée par la valeur du but φ et par le coût pour j de faire l'action α .

(4) Enfin l'agent i évalue l'utilité $U(i)_{d-}$ de déléguer α à j , et que α échoue. L'utilité $U(i)_{d-}$ est donnée par le coût pour j de faire l'action α et un coût additionnel lié à l'échec de α .

Chacune de ces situations-résultat est associée à une valeur de confiance : par exemple la situation (1) est associée à la valeur de confiance $DoT_{ii\alpha}$ (confiance en soi de i pour faire α) et la situation (2) est associée à $1 - DoT_{ii\alpha}$. À partir de ces valeurs, les auteurs définissent la règle de décision qui permet à l'agent de choisir de déléguer ou de ne pas déléguer :

$$DoT_{ij\alpha} \times U(i)_{d+} + (1 - DoT_{ij\alpha})U(i)_{d-} > DoT_{ii\alpha} \times U(i)_{p+} + (1 - DoT_{ii\alpha})U(i)_{p-} . \quad (3.3)$$

Si la formule 3.3 est vérifiée, alors i délèguera à j l'action α . Cette règle de décision est basée sur un calcul complexe reflétant les pertes et les gains de chaque solution relativement aux autres.

Opérationnalisation

Castelfranchi and Falcone (2010) proposent deux opérationnalisations de leur modèle : la première vise à illustrer le processus de calcul d'une valeur de confiance à partir de différentes sources, et la seconde porte sur la décision de faire confiance (i.e. de déléguer).

Calcul de la confiance. Les auteurs proposent une opérationnalisation de leur modèle à l'aide de Fuzzy Cognitive Map (FCM). Les FCM combinent les avantages de la logique floue et des Cognitive Maps. Les FCM sont des graphes dont les nœuds représentent les concepts du modèle. Les arcs orientés et pondérés (avec des valeurs floues) indiquent l'effet des concepts sources sur les concepts cibles et leur force. Certaines valeurs peuvent être renseignées par l'utilisateur et les autres sont calculées dynamiquement par la FCM lors du lancement de la simulation. Les valeurs finales sont obtenues lorsque l'état des nœuds de la FCM sont identiques au pas s et au pas $s - 1$. Dans l'opérationnalisation proposée, les concepts sont des valeurs qui représentent les croyances de l'agent i sur les différents éléments qui lui servent à calculer une valeur de confiance, et les arcs indiquent l'impact de chaque élément sur la valeur de confiance finale.

Le scénario suivant est proposé par les auteurs : un agent *malade* cherche à savoir s'il doit aller consulter l'agent *médecin* (i.e. s'il a confiance en l'agent *médecin* pour le soigner : l'action déléguée est ici *soigner*) (Castelfranchi and Falcone, 2010; Falcone et al., 2002). L'agent *malade* considère plusieurs éléments pour calculer sa confiance en l'agent *médecin* (**Trusfulness** de l'agent *médecin*) :

- Les facteurs internes (**Internal Factors**) sont liés au *médecin* lui-même, et sont la capacité (**Ability**) du *médecin* à *soigner*, la disponibilité (**Availability**) du *médecin* et le caractère non dangereux (**Unharmfulness**) de ce dernier.
- Les facteurs externes (**External Factors**) sont liés à la situation considérée : l'opportunité (**Opportunity**) pour le *malade* d'aller voir le *médecin* et les dangers (**Danger**) liés à la visite du *médecin* (e.g. le *médecin* peut critiquer le mode de vie du *malade*).

La confiance de l'agent *malade* est calculée à partir de différentes sources d'informations : son expérience directe (**DE**), le rôle que joue l'agent *médecin* (**C** pour Catégorisation; e.g. de par son rôle, l'agent *médecin* est supposé être capable de soigner l'agent *malade*), le raisonnement (**R**) et les autres sources d'informations incluant la réputation (**O/R** pour *Other/Reputation*).

La FCM produite pour ce scénario est donnée figure 3.1. Ici, la décision de déléguer ou non l'action *soigner* au *médecin* n'est pas détaillée : les auteurs précisent que d'autres paramètres peuvent entrer en compte, par exemple le coût de la consultation.

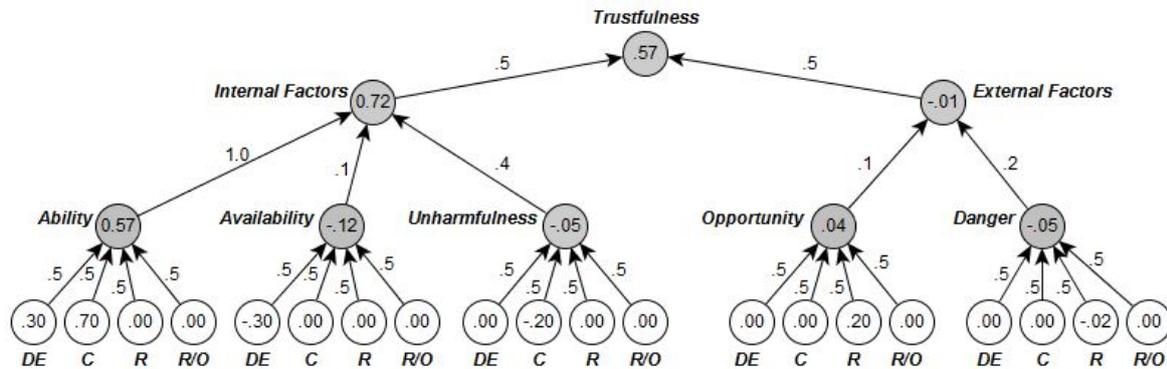


FIGURE 3.1 – Exemple d'une FCM décrivant la confiance de l'agent *malade* en l'agent *médecin*.

Les valeurs grisées sont les valeurs calculées par le système (Castelfranchi and Falcone, 2010, p. 318).

Décision de faire confiance. Les auteurs proposent un second scénario de Contract Net dans lequel les agents gagnent des points pour chaque tâche accomplie, qu'ils accomplissent la tâche eux-mêmes ou qu'ils la délèguent. Chaque agent a donc un ensemble de tâches à accomplir ; une valeur de compétence pour chaque tâche possible (choisie au hasard) ; une valeur de volonté (choisie au hasard) qui déterminera la quantité de ressources que l'agent investit dans chaque tâche qu'il essaie d'accomplir ; et une stratégie de délégation, parmi les quatre suivantes :

- *hasard* : l'agent choisit au hasard à qui déléguer la tâche.
- *statistique* : pendant une phase d'entraînement, l'agent est confronté à tous les autres et évalue de manière statistique leur caractère digne de confiance par rapport à leur nombre de succès et d'échecs.
- *confiance cognitive* : des renseignements sur les compétences et la volonté des autres agents sont donnés aux agents qui emploient cette stratégie, et l'agent choisit celui qui a les meilleures valeurs de compétence et volonté. La confiance est calculée grâce à des FCM ici également.

Cette expérience montre que les agents qui utilisent la stratégie de confiance cognitive sont plus performants que les autres.

Avantages et limites

Contrairement à Repage et BDI+Repage, différents aspects de la confiance sont considérés dans ce modèle, notamment la volonté et la compétence de l'agent auquel on souhaite déléguer. De nombreux autres éléments comme l'opportunité ou le caractère non dangereux sont également considérés ; ce qui rend le modèle proposé difficile à appréhender. C&F prennent également en compte de nombreuses sources d'information, comme cela a été proposé dans certains modèles numériques de la confiance. Par ailleurs les règles de décision à partir des valeurs de confiance

calculées, bien que complexes, sont précisées : c'est un modèle décisionnel s'appuyant sur la confiance.

3.3.3 ForTrust

ForTrust⁴ – *Social trust analysis and formalization* est un système de confiance cognitif né d'un projet ANR du même nom (Herzig et al., 2008, 2010; Lorini and Demolombe, 2008a,b).

ForTrust s'inspire des travaux de Castelfranchi and Falcone (1998) et définit une formalisation en logique multimodale \mathcal{L} des concepts de confiance et de réputation, le but étant de proposer une spécification formelle des conditions dans lesquelles on peut dire qu'un agent fait confiance à un autre. Castelfranchi and Falcone (1998) proposent la définition suivante de la confiance : i fait confiance à j pour faire l'action α dans l'optique de réaliser le but φ si et seulement si i a le but φ et i pense que :

- j est capable de faire α ,
- j , en faisant α , réalise le but φ , et
- j a l'intention de faire α .

Exprimée en logique \mathcal{L} , on obtient la définition 3.4 :

$$\text{Trust}(i, j, \alpha, \varphi) \stackrel{\text{def}}{=} \text{Choice}_i \text{F}\varphi \wedge \text{Bel}_i(\text{Capable}_j(\alpha) \wedge \text{After}_{j:\alpha}\varphi \wedge \text{Intends}_j(\alpha)) \quad (3.4)$$

où :

- $\text{Choice}_i \text{F}\varphi$ signifie " i veut que φ soit finalement vrai"; autrement dit i a le but φ .
- $\text{Bel}_i(X)$ signifie " i pense que X ".
- $\text{Capable}_j(\alpha)$ signifie " j est capable de faire α ".
- $\text{After}_{j:\alpha}\varphi$ signifie " j a l'opportunité de réaliser le but φ en faisant α ".
- et $\text{Intends}_j(\alpha)$ signifie " j a l'intention de faire α ".

Herzig et al. (2010) mettent en évidence le manque de précision de cette définition : qu'entend-on par j a l'intention de faire α ? j va-t-il faire α maintenant, ou plus tard? Lorsque j procrastine et ne fait pas α alors qu'il en a l'occasion, peut-on toujours considérer que i a confiance en j ?

Les auteurs proposent de distinguer deux concepts : la confiance *occurrente* (*occurrent trust*) et la confiance *dispositionnelle* (*dispositional trust*).

Confiance occurrente

La confiance *occurrente* caractérise le fait que i fait confiance à j pour faire α dans l'immédiat. Autrement dit, si j est capable de faire α et en a l'intention, alors il va faire α . En logique \mathcal{L} , cette proposition est traduite par : $\text{Does}_{i:\alpha}\top \leftrightarrow \text{Capable}_j(\alpha) \wedge \text{Intends}_j(\alpha)$, où $\text{Does}_{i:\alpha}\top$ se lit " j va faire α ". La définition de la confiance *occurrente* est donc donnée par la formule 3.5 (Herzig et al., 2010) :

$$\text{OccTrust}(i, j, \alpha, \varphi) \stackrel{\text{def}}{=} \text{Choice}_i \text{F}\varphi \wedge \text{Bel}_i(\text{Does}_{j:\alpha}\top \wedge \text{After}_{j:\alpha}\varphi) \quad . \quad (3.5)$$

4. Lien vers la page Internet du projet ForTrust [ici](#).

Confiance dispositionnelle

Les auteurs définissent le fait pour i d'avoir une confiance dispositionnelle en j comme suit : i n'a pas encore le but φ , mais pourrait l'avoir (φ est un but potentiel) si la circonstance κ se présente; et i pense que dès lors qu'il aura le but φ dans la circonstance κ , j fera l'action α qui permettra de réaliser φ . La définition de la confiance dispositionnelle est donc donnée par la formule 3.6 (Herzig et al., 2010) :

$$\text{DispTrust}(i, j, \alpha, \varphi, \kappa) \stackrel{\text{def}}{=} \text{PotGoal}_i(\varphi, \kappa) \wedge \text{Bel}_i \mathbf{G}*((\kappa \wedge \text{Choice}_i \mathbf{F}\varphi) \rightarrow (\text{Does}_{j:\alpha} \wedge \text{After}_{j:\alpha} \varphi)) \quad (3.6)$$

où :

- $\text{PotGoal}_i(\varphi, \kappa)$ signifie " i a le but potentiel φ dans la circonstance κ "; c'est-à-dire que i pense que dans le futur, la circonstance κ se produira et i aura alors le but φ .
- $\text{Bel}_i \mathbf{G}*(X)$ signifie que i pense que X est vrai maintenant et sera vrai dans le futur.
- et $(\kappa \wedge \text{Choice}_i \mathbf{F}\varphi) \rightarrow (\text{Does}_{j:\alpha} \wedge \text{After}_{j:\alpha} \varphi)$ exprime le fait que si κ se produit et i a le but φ , alors j fera α et réalisera ainsi le but φ .

Confiance graduée

Dans la formalisation précédemment décrite, la confiance n'est pas associée à une valeur numérique mais est binaire. Autrement dit, la proposition " i a confiance en j pour faire α " est vraie ou fausse; on ne peut pas exprimer le fait que i a *faiblement* ou *fortement* confiance en j pour faire α . Pour remédier à cette limitation, Lorini and Demolombe (2008a) proposent une formalisation de la confiance graduée, dans laquelle les croyances graduées sont utilisées. La définition 3.7 de la confiance occurrente graduée est très semblable à celle de la confiance occurrente (voir 3.5) et est la suivante :

$$\text{OccTrust}^x(i, j, \alpha, \varphi) \stackrel{\text{def}}{=} \text{Choice}_i \mathbf{F}\varphi \wedge \text{Bel}_i^x(\text{Does}_{j:\alpha} \top \wedge \text{After}_{j:\alpha} \varphi) \quad (3.7)$$

Cette définition se lit de la façon suivante : i a confiance au degré x en j pour faire α et ainsi réaliser φ si et seulement si i a le but φ et i pense au degré x que j va faire α et en faisant α j réalisera φ . Ici, la règle appliquée pour le calcul du degré de croyance dans une combinaison de croyances est la suivante : $\text{Bel}_i^x(X) \wedge \text{Bel}_i^y(Y) = \text{Bel}_i^{\min(x,y)}(X \wedge Y)$ ⁵.

Opérationnalisation

Hübner et al. (2009) proposent un exemple d'opérationnalisation et d'évaluation à partir de la formalisation ForTrust ainsi qu'un scénario d'évaluation du modèle. Le scénario est le suivant : l'agent i est en compétition avec d'autres agents et a un ensemble d'œuvres d'art à évaluer qui appartiennent à différents domaines d'expertise. L'agent i ayant un niveau de compétence différent pour chaque domaine d'expertise, il ne peut pas évaluer toutes ses œuvres et doit demander l'aide des autres. Chaque autre agent j a alors le choix de ne pas répondre à i , de répondre à i en disant la vérité, ou de mentir à i . Avant de demander à j son opinion, i doit donc calculer s'il peut faire confiance à j ou non.

5. Il n'y a pas de consensus dans la littérature sur la formule à utiliser pour la combinaison des croyances : parfois, on utilise la formule $\text{Bel}_i^x(X) \wedge \text{Bel}_i^y(Y) = \text{Bel}_i^{\min(x,y)}(X \wedge Y)$, parfois on utilise la formule $\text{Bel}_i^x(X) \wedge \text{Bel}_i^y(Y) = \text{Bel}_i^{x \times y}(X \wedge Y)$, comme par exemple dans BDI+Regpage.

Calcul de la confiance. La définition de la confiance utilisée est celle donnée dans la formule 3.7. Le but de i étant $\varphi = \textit{obtenir une évaluation juste pour la peinture } p$, i prend en compte différents éléments relatifs à j :

- $\text{Bel}_i^{x_1}(\text{Does}_{j:\alpha}\top)$: la capacité et l'intention de j de réaliser $\alpha = \textit{évaluer la peinture } p$ est évaluée par i à partir de l'historique de ses interactions avec j . Si le taux τ de réponse de j aux demandes précédentes d'évaluation de i dépasse un seuil $\epsilon = 0.5$, alors le degré de croyance x_1 est égal à τ , sinon x_1 est égal à 0.
- $\text{Bel}_i^{x_2}(\text{After}_{j:\alpha}\varphi)$ correspond ici à ce que i pense du pouvoir de j de produire une évaluation juste pour la peinture p ; c'est-à-dire le pouvoir de j de réaliser φ en faisant α . Ce pouvoir est évalué par i en considérant d'une part la sincérité de j (i.e. est-ce que j ment? – j est considéré comme menteur si son taux d'erreur est élevé) et d'autre part avec l'image que i a de j . L'image que i a de j est une valeur $image \in [0; 1]$ et correspond à une estimation de la qualité des évaluations de j par i ; cette image étant calculée de manière à donner plus de poids aux évaluations récentes qu'aux évaluations anciennes. Si $image$ dépasse un seuil $\delta = 0.5$, alors $x_2 = image$, sinon $x_2 = 0$.

Évaluation. L'évaluation proposée par les auteurs montre une très bonne performance (jugée en fonction du nombre de peintures correctement évaluées) de l'agent ForTrust comparé à des agents qui utilisent d'autres systèmes de confiance. Par ailleurs, les auteurs ont comparé des agents qui utilisent simplement $\text{Bel}_i^{x_1}(\text{Does}_{j:\alpha}\top)$ ou simplement $\text{Bel}_i^{x_2}(\text{After}_{j:\alpha}\varphi)$ pour décider de faire confiance à des agents qui utilisent les deux (i.e. agents ForTrust) : les résultats montrent que la combinaison des deux éléments permet une meilleure performance des agents que lorsqu'ils n'utilisent que la capacité et l'intention ou que le pouvoir.

Avantages et limites

La formalisation proposée dans ForTrust est extrêmement claire et précise et permet de clarifier certains points de la proposition de Castelfranchi and Falcone (1998) qui étaient laissés libres d'interprétation au lecteur. Chaque concept de la définition d'origine de Castelfranchi and Falcone (1998) est exprimé à l'aide d'un opérateur de la logique multimodale \mathcal{L} qui a été définie par les auteurs spécifiquement pour combiner l'expressivité des logiques dynamique (Harel et al., 2000, cité par Herzig et al. (2010)) et BDI (Chellas, 1980, cité par Herzig et al. (2010)). Le sens de chaque opérateur est défini de façon non ambiguë, ce qui permet de clarifier chaque concept et de proposer des définitions non ambiguës adaptées au raisonnement agent. Par ailleurs, la formalisation de Lorini and Demolombe (2008a) utilisant les croyances graduées permet non plus d'exprimer simplement des propositions vraies ou fausses, mais des degrés de confiance. Cette formalisation précise dans quels cas un agent doit faire confiance à un autre (i.e. déléguer) : ForTrust est donc un système décisionnel s'appuyant sur la confiance.

Cependant, bien qu'il soit possible d'exprimer des degrés de croyance, il est impossible d'exprimer le fait que i est sûr (pense au degré 100%) que j n'est capable de faire α qu'à 25% (i.e. j n'est pas très compétent), et i n'a donc pas beaucoup confiance en j (i.e. à 25%) pour faire α . Cela paraît problématique, puisque l'on peut être plus ou moins compétent sur une tâche donnée. Si cette notion n'est pas présente dans les travaux de Castelfranchi and Falcone (1998), elle a été ajoutée dans des travaux ultérieurs (Castelfranchi and Falcone, 2010).

Enfin, ces travaux sont centrés sur la notion de délégation : dans le cas où l'on ne peut pas faire soi-même l'action qui permettra d'atteindre son but, on cherche la personne la plus digne de confiance à qui déléguer cette action. À l'opposé, la notion d'adoption de but n'est pas abordée : dans le cas où l'agent i délègue à l'agent j l'action α , dans quel cas l'agent j doit-il accepter de faire cette action ? De même, il est impossible ici pour un agent j de proposer son aide à l'agent i : l'adoption altruiste de but (voir 4.3.1) n'est pas considérée. Or ces notions sont importantes à prendre en compte dans le cadre de nos travaux : en effet, dans les activités collaboratives, il est souhaitable que les agents soient capables d'anticiper les besoins des autres pour proposer une aide.

3.3.4 Conclusion

Nous nous sommes donc intéressés dans cette section aux systèmes de confiance cognitifs ; nous avons remarqué qu'il n'existe pas de consensus autour de la définition de ces systèmes et que, selon les auteurs, certains systèmes peuvent être classés à la fois comme cognitifs et comme non cognitifs. Pour clarifier notre analyse, nous avons adopté la définition des systèmes de confiance cognitifs de Pinyol and Sabater-Mir (2013) (voir p. 36) : un système de confiance cognitif propose une formalisation de la confiance en termes d'éléments tels que les désirs et les croyances. Nous avons ensuite étudié les trois systèmes de confiance cognitifs majeurs : BDI+Repage, C&F et ForTrust. Nous avons pu constater que tous ces systèmes sont des systèmes décisionnels s'appuyant sur la confiance. De plus, tous s'inspirent de travaux en SHS : les décisions prises par les agents sont donc explicables au regard des théories sous-jacentes.

Dans tous ces travaux, le point de vue considéré est celui de l'agent qui cherche à déléguer une tâche ; on ne s'intéresse pas à l'agent à qui la tâche est déléguée (doit-il accepter ou refuser ?). Pourtant la problématique de l'adoption de but est elle aussi liée à la confiance. Par exemple Castelfranchi and Falcone (2010) discutent des différents types d'adoption de but (voir 4.3.1). Dans le cadre de nos travaux, nous nous intéressons à ces aspects puisque nous souhaitons qu'un agent puisse anticiper les besoins des autres. De tels systèmes ne suffisent donc pas pour répondre à notre problématique.

3.4 Bilan du chapitre

Nous avons identifié la confiance comme facteur clé pour la génération des comportements d'agents en équipe. Nous avons donc abordé dans ce chapitre les différents environnements virtuels et les différents SMA qui utilisent la notion de confiance.

Nous avons tout d'abord passé en revue les travaux sur les personnages virtuels qui utilisent la notion de confiance. Nous avons distingué deux types de démarches : certains travaux visent à identifier les éléments d'apparence, d'animation et/ou des stratégies comportementales qui permettent à l'agent de créer un climat de confiance avec l'utilisateur. Dans ces travaux, la confiance est utilisée comme variable d'évaluation mais n'est pas opérationnalisée. Une autre démarche consiste à opérationnaliser la confiance pour permettre aux agents de prendre des décisions informées au regard de leurs relations avec les autres. Cependant, il n'existe pas à notre connaissance de travaux qui s'appuient sur un modèle de confiance issu de la littérature

en SHS, ce qui peut être gênant par rapport à la validité et à l'explicabilité des comportements produits.

Dans un second temps, nous avons abordé les SMA utilisant des notions de confiance et/ou de réputation. Nous avons vu que ces deux notions sont souvent associées dans la littérature et qu'il n'existe pas de consensus sur leur définition. C'est pourquoi nous avons considéré les deux types de systèmes. Nous avons, en nous appuyant sur les travaux de Pinyol and Sabater-Mir (2013), proposé une définition d'un système décisionnel reposant sur la confiance et/ou la réputation : dans un tel système, les processus de prise de décision à partir des éléments de confiance et/ou de réputation sont spécifiés. Nous avons également vu qu'il existe deux grandes catégories de systèmes de confiance et/ou de réputation : les systèmes numériques et les systèmes cognitifs. Les systèmes numériques ne proposent pas toujours de formalisation claire de la confiance et s'intéressent avant tout à la performance du système, ce qui ne correspond pas à nos besoins.

Nous avons ensuite abordé les systèmes de confiance cognitifs, dans lesquels la confiance est formalisée au travers d'éléments tels que les croyances et les désirs. Les trois systèmes de confiance cognitifs que nous avons étudiés sont tous des systèmes décisionnels reposant sur la confiance et s'appuient tous sur des théories de SHS, ce qui permet de générer des comportements qui sont explicables. Le premier de ces systèmes, BDI+Repage, propose une formalisation des concepts d'image et de réputation. La principale limite de ce système est que, pour chaque valeur de confiance considérée, un seul aspect est pris en compte. À l'inverse, dans le modèle de C&F comme dans ForTrust, plusieurs aspects sont pris en compte, ce qui permet par exemple de distinguer la *volonté* (dans le modèle de C&F) ou l'*intention* (dans ForTrust) de l'agent B et la *compétence* de l'agent B pour faire l'action α que l'agent A envisage de déléguer. Pouvoir distinguer plusieurs aspects tel que proposé dans les modèles de C&F ou dans ForTrust est important dans le cadre de nos travaux : nous souhaitons en effet pouvoir confronter l'apprenant à des situations dans lesquelles ses coéquipiers mettraient de la mauvaise volonté à participer à l'activité collective bien qu'ils soient compétents.

Enfin, et en opposition aux "mauvais" coéquipiers, nous souhaitons également pouvoir générer des comportements idéaux (e.g. on peut avoir une équipe composée de quatre agents dont trois ont des comportements idéaux – participent à la tâche collective et prêtent attention aux autres – et le dernier est un "mauvais" coéquipier). Dans ce cas, considérer uniquement les situations de délégation, comme le font les systèmes de confiance et/ou de réputation dans la littérature SMA, ne suffit pas. En effet, dans ces systèmes, la problématique de l'adoption de but n'est pas abordée.

Pour proposer un système décisionnel qui tienne compte de tous ces éléments, nous abordons dans le chapitre 4 la littérature en SHS sur la confiance. Bien comprendre ce concept nous permettra d'identifier les éléments nécessaires à notre système décisionnel.

Chapitre 4

Étude de la confiance en sciences humaines et sociales

Nous avons abordé dans le chapitre 2 les problématiques liées à la génération de comportements collaboratifs pour des personnages virtuels en environnement virtuel. Nous avons constaté que ces systèmes laissent souvent de côté la problématique des facteurs humains, or nous pensons que ces facteurs sont indispensables à prendre en compte pour générer des comportements crédibles. En particulier, nous pensons que la confiance est un facteur clé qui influence toutes nos décisions et notamment celles en rapport avec les activités collaboratives. Nous avons donc, dans le chapitre 3, étudié les environnements virtuels et les systèmes multi-agents (SMA) qui intègrent ce facteur. Cependant, cette étude ne nous a pas permis d'identifier un modèle qui soit satisfaisant par rapport à nos besoins.

Nous abordons ainsi la littérature sur la confiance dans le domaine des sciences humaines et sociales (SHS), de manière à, dans un premier temps, mieux comprendre les influences de la confiance sur le comportement. Nous tâcherons ensuite de définir la confiance et les principales notions qui y sont liées. Enfin, nous aborderons la problématique de la composition de la confiance (i.e. où la confiance trouve-t-elle ses sources?) et de sa représentation.

Avant de développer ces éléments et pour faciliter la lecture de ce chapitre, nous introduisons quelques termes de vocabulaire liés à la confiance. Une relation de confiance implique deux entités : le sujet de la relation de confiance A et l'objet de la relation de confiance B. En anglais, les termes utilisés pour désigner A sont *trustor* ou *truster*, et le terme utilisé pour désigner B est *trustee*. La langue française ne possédant pas de traduction de ces termes, pour plus de simplicité et de lisibilité nous utiliserons par la suite les mots anglais *trustor* et *trustee*. Ainsi, on dira que "le *trustor* fait confiance au *trustee*".

4.1 Confiance et comportements

Nous avons vu que dans le domaine des systèmes multi-agents (SMA), la confiance est utilisée comme mécanisme de régulation des échanges : par exemple dans un système de e-commerce, un agent utilise des valeurs de confiance pour décider avec qui traiter. Par ailleurs, la confiance est associée notamment par [Castelfranchi and Falcone \(2010\)](#) à la notion de délégation. Nous avons également établi que nous souhaitons adopter une approche qui permettra l'émergence

de la collaboration (voir 2.2.2). Étudier les influences de la confiance sur le comportement sous l'angle des SHS nous permettra d'en comprendre les impacts et les enjeux ainsi que d'identifier les comportements que nous souhaitons pouvoir reproduire pour nos agents virtuels.

4.1.1 Confiance et groupes

De nombreuses études ont montré l'influence de la confiance sur les comportements de groupe. Nous abordons ici les principaux phénomènes liés à la confiance observés au niveau du groupe.

La confiance favorise la performance d'un groupe

Palanski et al. (2011) ont conduit deux expériences pour étudier la relation entre la confiance et la performance. La première expérience a été conduite en laboratoire : des équipes de trois à cinq étudiants devaient travailler sur un projet pendant huit semaines, qui était ensuite évalué par des professeurs (mesure de performance des groupes). La confiance des membres du groupe était évaluée par un questionnaire et les auteurs trouvent une relation significative entre le niveau de confiance et la performance du groupe. Pour confirmer ces résultats, les auteurs ont conduit une seconde expérience hors laboratoire dans le milieu hospitalier, sur des équipes d'infirmiers. La confiance était évaluée par le même questionnaire que dans la première expérience, alors que la performance de chaque équipe était estimée par le responsable de l'ensemble des équipes. Dans cette deuxième étude également, les auteurs trouvent une relation entre confiance et performance.

À plus grande échelle, Mayer and Gavin (2005) ont étudié l'influence de la confiance qu'ont les employés en leur leader sur la performance de l'équipe. Leurs travaux consistent en une étude de terrain sur plus de 200 employés répartis dans huit usines de la même société de production industrielle. Les résultats de cette étude montrent une influence de la confiance des employés en leur leader sur leur performance. Aucun autre facteur n'est pris en compte dans cette étude ; néanmoins les auteurs font l'hypothèse qu'un haut niveau de confiance de la part de ses employés permet au leader de mieux diriger leur attention sur les tâches importantes pour l'équipe et pour la société, et donc d'améliorer leur performance.

Des résultats similaires ont été rapportés par d'autres auteurs qui ont étudié l'influence de la confiance sur la performance (Chua et al., 2012; Davis et al., 2000; Dirks, 1999; Fink and Kessler, 2010).

La confiance nuit à la performance d'un groupe

À l'inverse, la confiance peut avoir une influence négative sur la performance d'un groupe : le phénomène de la pensée de groupe (*groupthink*) est notamment lié, entre autres choses, à la confiance d'après McKnight and Chervany (2001) ou Erdem (2003). La pensée de groupe caractérise la tendance d'un groupe à produire une performance médiocre ou mauvaise, voire catastrophique, suite à un processus de prise de décision dysfonctionnel car ses membres privilégient avant tout le fait d'être d'accord et d'éviter le conflit. Les membres d'un groupe victime de ce phénomène sont très liés : chacun a confiance en les autres et personne ne remettra en cause la parole ou la compétence de l'un des membres du groupe. Certaines personnes peuvent alors faire des erreurs qu'aucun des membres du groupe ne va corriger. Même si l'un des membres

identifie un problème de compétence ou un dysfonctionnement dans le groupe, il n'en fera pas part aux autres pour éviter le conflit.

Puisque l'influence de la confiance sur un groupe semble pouvoir être aussi bien positive que négative, nous proposons de nous intéresser aux conditions qui peuvent conduire à ces phénomènes. La question est donc la suivante : que se passe-t-il au niveau individuel qui conduise à l'émergence de tels phénomènes au niveau du groupe ?

4.1.2 Confiance et individus

Pour comprendre l'émergence de tels phénomènes liés à la confiance au niveau du groupe, nous nous intéressons à l'influence de la confiance sur les comportements individuels.

Confiance et coopération

Pour expliquer comment la confiance influence la performance d'un groupe, l'hypothèse suivante a été faite par de nombreux auteurs : la confiance favorise la coopération entre les individus, ce qui améliore la performance du groupe. Pour vérifier cette hypothèse, de nombreuses expériences ont été faites avec des jeux tels que ceux du dilemme du prisonnier ou du bien public.

Dans le jeu du dilemme du prisonnier qui se joue à deux, chaque personne peut choisir de *trahir* ou *coopérer*. Le binôme gagne globalement plus si les deux individus coopèrent, mais le gain individuel de chaque personne est plus élevé si elle trahit, indépendamment de ce que fait l'autre. De plus, la perte maximale est atteinte pour un joueur quand il coopère alors que l'autre trahit : la tentation de trahir est donc forte. Chaudhuri et al. (2002) montrent notamment que les personnes faisant le plus confiance sont aussi celles qui coopèrent le plus dans le jeu du dilemme du prisonnier. Ce jeu a également été utilisé notamment par Kee and Knox (1970), Parks and Hulbert (1995) ou Wong et al. (2005) pour étudier la confiance. Pour certains auteurs, le lien entre confiance et coopération est tellement évident que ces deux notions sont parfois confondues. C'est le cas par exemple dans les travaux de Deutsch (1958), qui étudie la confiance grâce au jeu du dilemme du prisonnier et qui parle de "confiance mutuelle" lorsque les deux sujets coopèrent et de "méfiance mutuelle" lorsque les deux sujets font le choix de trahir.

Le jeu du bien public est également utilisé pour étudier la confiance (Carpenter et al., 2004; Parks and Hulbert, 1995; Stouten et al., 2009). Dans le jeu du bien public, la même somme de départ est allouée à tous les participants. Chacun peut investir dans le pot commun une part de son capital personnel. La somme totale investie par l'ensemble des joueurs dans le pot commun est multipliée (par un facteur plus grand que 1 mais plus petit que le nombre de joueurs) et redistribuée à parts égales entre tous les joueurs. De même que le dilemme du prisonnier, pour l'intérêt commun les joueurs devraient coopérer, mais pour maximiser son gain individuel, chacun doit investir le moins possible dans le pot commun. Par exemple Gächter et al. (2004) montrent que les personnes qui ont tendance à faire confiance coopèrent plus facilement dans le jeu du bien public.

Confiance et délégation

Comme les auteurs cités précédemment, Castelfranchi (1998) argumente que la confiance est fondamentale pour la coopération. Néanmoins pour Castelfranchi, ce n'est pas l'acte de coopérer

qui est immédiatement lié à la confiance mais celui, pour le trustor, de déléguer, de compter sur le trustee. C'est cet acte de délégation qui permet d'établir une coopération entre le trustor et le trustee. Ce point de vue est notamment largement accepté dans la littérature sur la confiance dans le domaine des SMA, comme nous l'avons vu dans le chapitre 3.

Cependant, et contrairement à ce qui est fait dans le domaine des SMA, on peut distinguer plusieurs formes de délégations, selon que le trustee connaît les intentions du trustor ou non (Castelfranchi and Falcone, 2010; Castelfranchi, 1998). Dans la **délégation faible**¹, il n'y a pas d'accord explicite entre le trustee et le trustor; le trustee n'a, en fait, pas conscience que le trustor exploite son action. Par exemple, si dans une équipe de pompiers, Anna compte sur la maniaquerie de Béa pour ranger l'ambulance et le matériel après une intervention sans le lui demander directement, c'est une forme de délégation faible. Lorsque Béa rangera l'ambulance et le matériel, elle participera involontairement au plan d'Anna, dont elle fait partie sans le savoir. Dans la délégation faible, le trustee n'a pas de vision globale des plans du trustor. Lorsque le trustee a pleinement conscience des plans du trustor, on parle de **délégation forte**² (Castelfranchi and Falcone, 2010; Castelfranchi, 1998). Dans cette forme de délégation, le trustee adopte volontairement le but du trustor (éventuellement après négociation).

Dans les systèmes de confiance dans le domaine des SMA, seuls les comportements de délégation forte sont considérés : les agents cherchent le meilleur partenaire possible qu'ils pourraient contacter pour leur déléguer une tâche. Toutefois dans le cadre de nos travaux, la notion de délégation faible est particulièrement intéressante : en effet, la délégation faible suppose de prêter attention aux autres et d'anticiper leurs comportements, et c'est exactement le type de comportements que nous souhaitons pouvoir reproduire. Nous pensons également que la délégation faible contribue fortement à l'émergence de la collaboration puisqu'elle permet aux agents de s'organiser implicitement en tenant compte des comportements futurs des autres. Nous souhaitons donc proposer un moteur décisionnel qui permettra à nos agents d'adopter des comportements liés à la délégation faible.

Confiance et mauvais choix

Nous avons établi que la confiance favorise les comportements de coopération et de délégation au niveau individuel, ce qui permet à la collaboration d'émerger au niveau du groupe et favorise ainsi la performance du groupe. Cependant, nous avons également vu que la confiance mène parfois un groupe à prendre de mauvaises décisions : que se passe-t-il au niveau individuel qui explique cela ?

Erdem (2003) propose une explication à l'émergence du phénomène de la pensée de groupe : selon cet auteur, il faut distinguer plusieurs niveaux de confiance. Pour qu'un groupe de travail atteigne la meilleure performance et évite le phénomène de la pensée collective, le niveau de confiance entre ses membres ne doit pas être à son maximum mais atteindre un niveau *optimal*, en dessous du maximum. À ce niveau, les membres du groupe auront d'une part assez confiance pour s'exprimer, partager les informations et adopter des comportements d'aide les uns envers les autres, et auront d'autre part assez de méfiance (*distrust*) pour s'ouvrir à l'extérieur du groupe

1. Les auteurs emploient également les termes de délégation unilatérale ou délégation passive pour désigner cette forme de délégation.

2. Les auteurs utilisent aussi le terme délégation par acceptation comme synonyme de délégation forte.

et pour critiquer et mettre en doute les idées des autres dans le souci de trouver la meilleure solution. Dans le même ordre d'idées, Wicks et al. (1999) définissent la *confiance optimale* comme le fait d'être prudent dans ses relations tout en gardant la volonté de faire confiance. Le trustor devrait alors choisir prudemment à qui faire confiance, pourquoi, jusqu'à quel point et dans quel contexte.

Nous partageons le point de vue de ces auteurs : un trop haut niveau de confiance peut nuire à la performance du groupe. Cependant, nous pensons que ce qui conduit au phénomène de la pensée de groupe n'est pas le haut niveau de confiance en lui-même mais la surestimation par le trustor des qualités du trustee et donc les mauvaises prédictions du trustor sur les comportements du trustee.

Marsh and Briggs (2009) proposent la notion de **confiance mal placée**³. La confiance d'Anna en Béa est mal placée si Anna pense que Béa va ranger l'ambulance et le matériel (i.e. Béa est digne de confiance pour ranger l'ambulance et le matériel) alors qu'en réalité Béa n'adoptera pas le comportement attendu. Selon Marsh and Briggs, la confiance mal placée fait donc référence à un décalage entre l'évaluation par le trustor des caractéristiques du trustee et les caractéristiques réelles du trustee. Ce décalage peut aussi être dans le sens inverse : Anna n'a pas confiance en Béa pour ranger l'ambulance et le matériel alors qu'en réalité Béa accomplit parfaitement cette action. Ce mauvais jugement des autres et cette mauvaise estimation du comportement futur des autres peut conduire à un phénomène tel que la pensée de groupe.

Nous sommes particulièrement intéressés par la reproduction de ce genre de comportements : bien que les agents prennent en compte les autres, si leurs prédictions ne sont pas correctes, les comportements adoptés ne seront pas optimaux et il sera plus difficile pour l'équipe de s'organiser et de voir la collaboration émerger.

4.1.3 Conclusion

Nous avons vu dans le chapitre précédent lors de notre étude des systèmes de confiance dans le domaine des SMA que la confiance est liée à la délégation. Cependant notre étude de la littérature en SHS nous a permis de voir que la confiance influence bien d'autres aspects du comportement : nous nous sommes dans un premier temps intéressés à la confiance et à son influence sur les groupes. Nous avons vu que la confiance permet généralement d'améliorer la performance d'un groupe mais ce n'est pas systématiquement le cas. En effet, certains auteurs lient par exemple le phénomène de la pensée de groupe à la confiance.

Pour comprendre comment de tels phénomènes peuvent émerger au niveau du groupe, nous avons dans un second temps abordé l'influence de la confiance sur les comportements individuels. Si la confiance est très souvent liée à la coopération et étudiée dans des jeux tels que le dilemme du prisonnier ou le jeu du bien public, Castelfranchi and Falcone (2010) mettent en avant que c'est avant tout au travers de l'acte de délégation que l'on peut étudier la confiance. Les auteurs

3. Les auteurs emploient le terme de "mistrust"; cependant il n'y a pas de consensus autour du sens de ce terme. Ce terme est aussi employé par Castelfranchi and Falcone (2010) pour désigner l'évaluation du trustee par le trustor comme quelqu'un de non digne de confiance car incompetent ou non fiable, mais n'ayant pas de mauvaises intentions à l'égard du trustor. Les caractéristiques réelles du trustee n'entrent pas en jeu dans cette définition. Enfin, McKnight and Chervany (2001) emploient le terme de *mistrust* comme synonyme de *distrust*, et donc pour désigner un état de méfiance.

distinguent la délégation forte de la délégation faible : alors que dans le domaine des SMA c'est le problème de la délégation forte qui est traité, nous sommes particulièrement intéressés par la délégation faible qui permet notamment une répartition implicite des tâches dans un groupe grâce à la prise en compte des autres par chacun des membres.

Enfin, si la délégation faible doit permettre à la collaboration d'émerger, nous avons vu que cela n'est pas nécessairement le cas. En effet, si la confiance d'un agent en un autre est mal placée, ses prédictions sur le comportement des autres seront fausses, ce qui peut mener à des situations où la performance de l'équipe sera médiocre.

Puisque nous avons établi le type de comportements que nous souhaitons pouvoir reproduire pour nos agents virtuels et que nous avons montré comment la notion de confiance est liée à ces comportements, nous abordons dans la section suivante les différentes notions liées au concept de la confiance dans la littérature en SHS. En effet, la confiance est un concept complexe et comprendre les différentes notions qui y sont attachées nous permettra de mieux définir le cadre théorique de nos travaux.

4.2 Mieux définir la confiance

Il n'y a pas de consensus sur une définition de la confiance. Dans sa revue sur les définitions de la confiance proposées en psychologie, psychologie sociale et sociologie, [Castaldo \(2002\)](#) compte 72 définitions différentes. [Shapiro \(1987\)](#) parle d'un "pot-pourri" de définitions. Aborder les différentes définitions et les principales notions qui sont attachées à ce concept de la confiance nous permettra de mieux le comprendre et donc d'identifier les éléments essentiels pour construire notre moteur décisionnel.

4.2.1 Les différentes formes de confiance

Nous avons jusqu'ici discuté de la confiance entre deux personnes ou deux agents. Cependant, il existe bien d'autres formes de confiance qui sont étudiées dans la littérature de SHS.

Dans l'ensemble des définitions de la confiance que nous avons étudiées, le trustor est un agent capable d'avoir des croyances ou des attentes vis-à-vis d'un autre. Cet agent peut alors être un individu seul, un groupe d'individus ou une organisation. La nature du trustee ne fait en revanche pas consensus. Certains considèrent que l'on peut faire confiance à un objet ou à un événement. De manière similaire, [Castelfranchi and Falcone \(2010\)](#) précisent que l'objet de la confiance n'est pas nécessairement doué de capacités cognitives⁴ : on peut avoir confiance en le fait que le beau temps viendra après la pluie ou en la technologie.

Cependant cette acception large de la notion de confiance n'est pas partagée par tous. Dans de nombreux autres travaux, l'objet de la confiance est un agent : un individu seul, un groupe d'individus ou une organisation ([Jones, 1996](#); [Mayer et al., 1995](#)). Dans le cas où le trustee et le trustor sont tous les deux des agents (individus, groupe, organisation ou institution), on parle de **confiance sociale** ([Castelfranchi and Falcone, 2010](#); [Deutsch, 1958](#)).

En fonction de la nature exacte du trustee et du trustor et en fonction du contexte dans lequel

4. Dans d'autres travaux cependant, [Falcone and Castelfranchi \(2001\)](#) ou [Miceli and Castelfranchi \(2014\)](#) définissent la confiance comme étant forcément dirigée vers un agent cognitif.

la relation de confiance est étudiée, la confiance est qualifiée et conceptualisée de différentes manières :

- *Confiance et individus.* La forme de confiance la plus étudiée dans la littérature est celle qui existe entre deux individus, et on parle souvent de **confiance interpersonnelle**⁵ (Castaldo, 2002; Mayer et al., 1995; McKnight and Chervany, 2001). Une relation de confiance interpersonnelle peut désigner une relation de confiance entre deux individus qui ne se connaissent pas, auquel cas on peut parler de *confiance impersonnelle* (Mouzas et al., 2007) ou entre deux individus qui sont familiers, auquel cas on peut parler de *confiance relationnelle* (Rousseau et al., 1998) ou de *confiance personnelle* (Mouzas et al., 2007).
Mayer et al. (1995) proposent un modèle de la *confiance organisationnelle* ; l'utilisation de ce terme prête cependant à confusion car les auteurs s'intéressent avant tout à une relation de confiance dyadique dans laquelle trustor et trustee sont situés dans des organisations. Il s'agit donc en fait de confiance interpersonnelle.
C'est à cette définition de la confiance que nous avons fait référence jusqu'à maintenant ; et c'est à cette confiance interpersonnelle et relationnelle que nous nous intéressons dans le cadre de nos travaux. En effet, ce type de confiance permet à une personne de prendre en compte individuellement chaque membre de son groupe et influence toutes les décisions de cette personne. En effet, la réponse à une question telle que *Ai-je confiance en mon coéquipier pour faire cette tâche ?* sera déterminante pour la décision du trustor.
- *Confiance et organisation.* On parle de *confiance au niveau organisationnel* (Mishra, 1996) ou de *confiance inter-organisationnelle* (Castaldo, 2002; Mouzas et al., 2007) lorsque trustee et trustor sont des organisations, ou des ensembles d'individus composant les organisations. Les chercheurs qui s'intéressent à la confiance au niveau organisationnel ont souvent pour objet d'étude les transactions économiques inter-organisationnelles. La confiance est alors vue comme un mécanisme de contrôle alternatif pour ces transactions. Nous nous intéressons aux situations professionnelles et on pourrait donc faire le lien avec la confiance organisationnelle ; cependant, nous nous intéressons à des petites équipes de travail dans lesquelles il sera possible pour chacun de tenir compte des autres avant de prendre une décision. Nous souhaitons voir émerger la collaboration dans un groupe d'agents à partir des comportements individuels de chacun : nous ne sommes pas intéressés par la confiance au niveau organisationnel.
- *Confiance, institutions et société.* La *confiance institutionnelle* désigne la confiance en des institutions sociales (Castaldo, 2002) ou politiques (Mishra, 1996) : par exemple la confiance des français dans le système politique. La confiance institutionnelle est différente de la *confiance basée sur les institutions* (Hardin, 2002; McKnight and Chervany, 2001; Rousseau et al., 1998) : celle-ci est une relation de confiance entre deux partis (individus, groupes ou organisations) qui profite d'un support institutionnel assurant au trustor une certaine tranquillité de par les contrats et/ou sanctions prévus contre le trustee si celui-ci ne respectait pas ses engagements. Cependant le terme de confiance institutionnelle est

5. Il n'y a cependant pas de consensus sur la définition du terme confiance interpersonnelle, qui peut aussi être employé lorsque le trustee et/ou le trustor sont des groupes de personnes et non des individus isolés (Rotter, 1967).

parfois employé pour désigner la confiance basée sur les institutions (Rousseau et al., 1998).

Pour les mêmes raisons que précédemment, nous délaissions dans cette thèse la confiance institutionnelle.

4.2.2 Confiance contextualisée ou confiance décontextualisée ?

Une relation de confiance peut être considérée de deux manières : comme non située ou comme située. Autrement dit, on peut considérer que :

- "A fait confiance à B" (Lewicki et al., 1998; McKnight and Chervany, 2001; Rousseau et al., 1998). Dans ce cas, la confiance est considérée comme une relation bipartite. Nous désignons dans ce manuscrit ce type de confiance par les termes *confiance non-située* ou *confiance décontextualisée*, mais les termes *confiance ouverte* (Hardin, 2002) ou *confiance généralisée*⁶ (Castelfranchi and Falcone, 2010) peuvent aussi être employés.

Dans le chapitre précédent, nous avons vu que les agents du système de confiance AFRAS (voir 3.2.3) utilisent ce type de confiance et cela avait été l'occasion de discuter de la nécessité de pouvoir distinguer plusieurs confiances différentes pour une même paire trustee-trustor.

- "A fait confiance à B *pour faire C*" (Baier, 1986; Castelfranchi and Falcone, 2010; Deutsch, 1958; Gambetta, 2000; Hardin, 2002; Mayer et al., 1995). Dans ce cas, la confiance est comprise comme une relation tripartite. Nous emploierons le terme de *confiance située* ou *confiance contextualisée* pour désigner ce type de confiance.

De nombreux auteurs pensent que l'on ne peut pas décontextualiser la confiance. La confiance est forcément située autour d'une tâche, d'une action particulière pour laquelle le trustor a confiance dans le trustee. D'après Hardin (2002), même lorsque l'on déclare à quelqu'un qu'on lui fait confiance sans préciser *pourquoi*, la tâche par rapport à laquelle on fait confiance est sous-entendue et peut être comprise grâce au contexte. Par exemple, si Anna part en disant à Béa "je te fais confiance" après lui avoir expliqué comment vérifier les bouteilles à oxygène et comment les remplacer si besoin, la fin de la phrase *pour t'occuper des bouteilles à oxygène* est sous-entendue.

Comme nous l'avons déjà établi, c'est à ce type de confiance que nous nous intéressons dans cette thèse. En effet, nous voulons que l'agent Anna puisse distinguer sa confiance en l'agent Béa pour ranger l'ambulance de sa confiance en l'agent Béa pour prendre correctement la tension d'un blessé par exemple.

Cette distinction entre confiance située et non-située peut paraître simple et sans conséquence ; seulement il est important de comprendre que le débat sur le caractère situé ou non-situé de la confiance est loin d'être clos dans le domaine des SHS. Le fait qu'il n'y ait pas de consensus sur la nature située ou non-située de la confiance a des conséquences, et entraîne notamment une autre absence de consensus sur la représentation de la confiance. Nous aborderons ce point dans la partie 4.3.2.

6. La confiance généralisée de Castelfranchi and Falcone (2010) peut être généralisée pour un même trustee à d'autres tâches (A fait confiance à B) ou pour une même tâche à différents trustees (A fait confiance – à n'importe qui – pour faire C).

4.2.3 Croyances sur le futur

Bien qu'il n'y ait pas de consensus sur une définition de la confiance, la notion d'attente du trustor par rapport au trustee est présente dans une majorité des définitions. Par exemple pour Mayer et al. (1995), le trustor attend une action particulière de la part du trustee : "la confiance est la volonté d'être vulnérable aux actions d'un autre parti, cette volonté étant basée sur l'attente d'une action particulière de la part de cet autre parti"⁷.

Dans sa revue des définitions de la confiance, Castaldo (2002) trouve qu'environ 60% des définitions de la confiance comportent la notion d'attente du trustor par rapport au trustee. Castaldo note que les attentes du trustor peuvent faire référence à différents éléments : elles peuvent concerner un résultat favorable suite aux actions du trustee ou elles peuvent être fondées principalement sur des caractéristiques du trustee. Cependant, pour les définitions dans lesquelles les attentes du trustor sont fondées sur les caractéristiques du trustee, les auteurs cherchent principalement à prédire le comportement du trustee à partir de certaines de ses caractéristiques, et donc à justifier ou expliquer le fait que le trustor considère le trustee comme digne de confiance (voir section 4.3.1). Par exemple, McKnight and Chervany (2001) s'intéressent aux caractéristiques de prédictibilité, bienveillance, intégrité et compétence du trustee. Ces auteurs affirment alors qu'*un trustee qui se montre toujours (prédictibilité) volontaire (bienveillance) et capable (compétence) de servir les intérêts du trustor de manière honnête et éthique (intégrité) peut être considéré digne de confiance*. En effet, une personne possédant ces caractéristiques se comportera de manière favorable pour le trustor qui peut donc former des attentes positives vis-à-vis de cette personne.

Il est donc nécessaire pour nos agents de pouvoir former des attentes sur les comportements futurs des autres. Une manière de faire est de se projeter les uns à la place des autres et imaginer ce que feraient les autres. Cette capacité à raisonner sur l'état d'esprit des autres pour comprendre et prédire leurs comportements est appelée **théorie de l'esprit** (Carruthers and Smith, 1996). Un individu est capable de théorie de l'esprit s'il peut attribuer des états mentaux (croyances, désirs, etc.) à lui-même et à d'autres personnes, et raisonner sur ces états mentaux. Autrement dit, on est capable de théorie de l'esprit quand on simule, consciemment ou non, le raisonnement de l'autre.

L'être humain est naturellement doué de théorie de l'esprit et cette capacité se développe aux alentours de cinq ans (Carruthers and Smith, 1996). Une expérience classique pour vérifier si un agent a cette capacité est la suivante : *Maxi laisse la moitié de son goûter sur la table de la cuisine et sort jouer. Sa mère entre dans la cuisine, range le reste du goûter au frigo et part. Quand Maxi revient, où va-t-il chercher son goûter ?* La réponse est *sur la table*, mais par exemple les très jeunes enfants qui observent la scène ne comprennent pas que Maxi n'a pas la même information qu'eux. La théorie de l'esprit a notamment suscité de l'intérêt dans le cadre de l'étude de certaines pathologies telles que l'autisme (Baron-Cohen et al., 1985) ou la schizophrénie (Frith and Corcoran, 1996). D'autres espèces que l'Homme semblent être douées de théorie de l'esprit, comme par exemple le chimpanzé (Premack and Woodruff, 1978).

Pour proposer un moteur décisionnel reposant sur des relations de confiance, il est donc

7. "Trust [...] is the willingness of a party to be vulnerable to the actions of another party based on the expectations that the other party will perform a particular action" (Mayer et al., 1995, p. 712)

nécessaire de doter nos agents de théorie de l'esprit ; c'est-à-dire que nos agents devront avoir un modèle des autres (e.g. Quels sont les buts des autres ? Quelles sont leurs croyances ?) et devront pouvoir raisonner sur ces modèles.

Enfin, nous pouvons mentionner que pour une majorité des auteurs en SHS, avoir des attentes par rapport aux comportements futurs d'une autre personne ne suffit pas pour parler de confiance. Notamment, il est souvent mentionné que :

- Ces attentes doivent être en rapport avec un but du trustor (Castelfranchi and Falcone, 2010; Mayer et al., 1995).

Exemple 1 : Si Anna n'a pas le but que l'ambulance soit rangée, on peut dire qu'elle s'attend à ce que Béa range l'ambulance mais on ne peut pas dire qu'elle fait confiance à Béa pour ranger l'ambulance.

- Ces attentes doivent être positives (Jones, 1996; Rousseau et al., 1998).

Exemple 2 : Si Anna a le but que l'ambulance soit rangée, on ne peut pas dire qu'elle fait confiance à Béa pour ne pas ranger l'ambulance.

Toutefois, nous souhaitons proposer un moteur décisionnel s'appuyant sur les relations de confiance qui permette aux agents de tenir compte des autres avant de choisir une action. Dans ce contexte, bien que l'on ne puisse pas parler de confiance pour les situations présentées dans les exemples 1 et 2, il est nécessaire qu'Anna, pour choisir ses propres actions, puisse prédire ce type de comportements de la part de Béa (i.e. dans l'exemple 2, Anna choisira de ranger l'ambulance elle-même).

4.2.4 Conclusion

La littérature sur la confiance dans le domaine des SHS est très riche et il n'y a pas de consensus sur une définition de la confiance. Étudier cette littérature nous a cependant permis dans cette section de présenter les différentes formes de confiance qui existent : on peut distinguer principalement trois formes qui sont la confiance interpersonnelle, la confiance inter-organisationnelle et la confiance institutionnelle. Étant donné que nous travaillons sur les relations de confiance entre les individus constituant une petite équipe, nous nous intéressons ici à la confiance interpersonnelle.

Nous avons ensuite différencié la confiance contextualisée de la confiance décontextualisée ; c'est le premier de ces deux types de confiance qui nous intéresse, puisque nous souhaitons pouvoir différencier la confiance d'Anna en Béa pour ranger l'ambulance de la confiance d'Anna en Béa pour prendre la tension du blessé.

Enfin, nous avons abordé la notion de croyances sur le futur, qui est étroitement liée à celle de la confiance dans la grande majorité des définitions de ce concept. Nous avons établi que nos agents doivent être dotés de théorie de l'esprit, c'est-à-dire posséder un modèle mental des autres agents et être capables de raisonner sur ces autres agents pour prédire leurs comportements et ainsi décider d'une action à faire.

À partir de ces éléments, la question à laquelle nous allons chercher à répondre dans la section 4.3 est la suivante : sur quels éléments les prédictions d'un agent sur les comportements d'un autre doivent-elles s'appuyer ?

4.3 Modéliser la confiance

Nous cherchons dans cette section une réponse à la question *Comment modéliser la confiance ?* En effet, cela nous permettra de savoir sur quels éléments les prédictions d'un agent sur les comportements d'un autres sont fondées. Nous pourrions ainsi proposer un moteur décisionnel qui s'appuie sur ces éléments. Nous étudions donc dans un premiers temps les principales théories sur les fondements de la confiance puis nous chercherons une manière de représenter la confiance de manière à éclairer nos choix pour la proposition d'un modèle d'agent.

4.3.1 Les fondements de la confiance

De nombreux auteurs ont essayé de déterminer pourquoi l'on fait confiance. On fait confiance à quelqu'un lorsqu'on juge cette personne digne de confiance. Seulement, quels éléments prend-on en compte pour juger quelqu'un digne de confiance? Afin de définir sur quels éléments les croyances de nos agents doivent porter, nous étudions ici les principales théories sur les fondements de la confiance.

La motivation du trustee

Pour certains, les fondements de la confiance du trustor se trouvent en grande partie dans sa perception et son jugement des motivations du trustee⁸ (Castelfranchi and Falcone, 2010; Hardin, 2002; Jones, 1996). La question est alors la suivante : qu'est-ce qui motive le trustee à faire ce pourquoi on lui fait confiance? Si c'est Anna qui a le but de ranger l'ambulance, pourquoi Béa rangerait-elle l'ambulance?

Hardin (2002) propose la théorie de l'**encapsulation des intérêts** : on pense que c'est dans l'intérêt du trustee de faire telle action car c'est dans notre intérêt, et on pense que le trustee accorde de la valeur à nos intérêts. Lorsque le trustee encapsule les intérêts du trustor, cela ne signifie pas forcément qu'il va faire ce pourquoi le trustor lui fait confiance, car il peut avoir d'autres priorités. Par exemple Béa peut se sentir concernée par le but d'Anna de ranger l'ambulance, mais avoir pour priorité de vérifier les bouteilles d'oxygène.

Castelfranchi and Falcone (2010) précisent davantage cette notion d'encapsulation des intérêts, qu'ils appellent par ailleurs **adoption sociale de but** : le trustee adopte un but du trustor, ce qui permet au trustor de lui faire confiance. Si le trustee adopte le but du trustor car il y cherche un avantage purement personnel, égoïste, on parle d'**adoption instrumentale**. Si Béa range l'ambulance pour qu'ensuite Anna l'aide à vérifier les bouteilles d'oxygène, c'est une adoption instrumentale. L'**adoption coopérative** se produit quand le trustor et le trustee sont dépendants l'un de l'autre pour la réalisation d'un but commun. Enfin, Castelfranchi and Falcone parlent d'**adoption altruiste** lorsque le trustee n'a rien à gagner, personnellement, à aider le trustor. Nous avons notamment étudié dans le chapitre 2 le simulateur DEPNET (voir 2.2.2) : en raisonnant sur des réseaux de dépendances, deux agents peuvent identifier des situations où ils dépendent mutuellement l'un de l'autre : ces agents pourront alors adopter

8. Il faut noter que les auteurs qui s'intéressent aux motivations du trustee ne nient pas l'importance de la prise en compte des compétences du trustee. Seulement l'évaluation des compétences du trustee ne peut pas à elle seule servir au trustor pour faire une prédiction sur le comportement du trustee.

instrumentalement ou coopérativement les buts de l'autre. L'adoption altruiste de but n'est en revanche pas traitée.

Dans le cadre de nos travaux, nous souhaitons voir la collaboration émerger dans un petit groupe d'agents, et nous souhaitons que dans les situations idéales, les agents puissent prêter attention aux autres et même apporter une aide : nous sommes particulièrement intéressés par le cas de l'adoption altruiste de but. Une question se pose alors : comment Anna peut-elle savoir si Béa va ou ne va pas adopter son but ? C'est la question à laquelle tentent de répondre les auteurs qui s'intéressent aux caractéristiques du trustee qui justifient la confiance du trustor.

Modèles dimensionnels de la confiance

Pour certains auteurs, la confiance du trustor ne repose pas exactement sur les motivations du trustee mais sur des caractéristiques de ce dernier (Castelfranchi and Falcone, 2010; Deutsch, 1958; Mayer et al., 1995; McKnight and Chervany, 2001; Sitkin and Roth, 1993). Il peut s'agir par exemple de sa compétence, son intégrité, son honnêteté, sa gentillesse, etc. La liste des caractéristiques ou dimensions citées dans la littérature est longue. Les chercheurs ont alors essayé de déterminer lesquelles de ces dimensions sont essentielles pour en réduire le nombre et proposer un modèle de confiance. Lorsque ces modèles prennent en compte plusieurs caractéristiques différentes, on parle souvent de **modèles multidimensionnels de la confiance** (Mishra, 1996). De très nombreux modèles dimensionnels de la confiance ont été proposés et nous présentons trois de ces modèles qui sont très différents mais correspondent chacun à des travaux majeurs dans la littérature en SHS sur la confiance.

Les deux premiers modèles présentés nous permettrons de mettre en avant des éléments qui ne sont pas intégrés à ces modèles mais qui sont selon nous nécessaires pour proposer un modèle d'agent et un moteur décisionnel associé. Nous pourrions ainsi mettre en évidence la manière dont le troisième modèle présenté répond à nos besoins.

Deutsch (1958) : intégrité. Dans son étude sur la confiance, le chercheur en psychologie sociale Deutsch (1958) caractérise une personne digne de confiance comme une personne responsable, et définit la responsabilité de la manière suivante : une personne responsable produira le comportement X attendu par le trustor, même si le comportement Y lui serait plus avantageux dans l'immédiat. Deutsch explique que les raisons qui poussent quelqu'un à se montrer responsable sont multiples (eg. lien personnel avec le trustor, peur des sanctions, adhésion à certaines valeurs, etc.), mais que l'on peut lier le caractère responsable d'une personne à un trait de personnalité : l'**intégrité**. L'auteur définit en effet l'intégrité comme le fait d'accorder de la valeur à se comporter de manière responsable et le fait d'être capable d'éviter ou de résoudre les situations de conflit de responsabilité (i.e. deux trustors qui font confiance au même trustee, pour produire deux comportements différents et en conflit). Une personne intègre est également, d'après Deutsch, capable de faire siens les buts des autres.

Ici aucune autre caractéristique du trustee que son intégrité n'est considérée, pourtant Deutsch reconnaît que d'autres éléments peuvent influencer la décision de faire confiance, comme le lien personnel du trustee avec le trustor. Ce type d'élément lié non pas simplement à des principes et valeurs mais à des relations humaines nous paraît important à prendre en compte dans nos travaux. En effet, on veut pouvoir nuancer le comportement de l'agent x en fonction de

l'autre agent considéré : x devrait pouvoir adopter le but de l'agent y et pas celui de l'agent z . Dans des groupes de travail où les individus se connaissent personnellement, il est normal que des affinités se développent, et ces affinités influencent forcément les comportements des individus.

Mishra (1996) : compétence, ouverture d'esprit, souci, fiabilité. Mishra (1996) propose un modèle multidimensionnel de la confiance organisationnelle issu d'une double analyse : d'une part, la littérature sur la confiance est passée en revue, et d'autre part, une étude de terrain est réalisée dans de grandes entreprises automobiles. Les quatre dimensions proposées sont les suivantes :

- la **compétence** du trustee, qui est évaluée par le trustor de manière générale et ne fait pas référence à une tâche précise ;
- l'**ouverture d'esprit** (*openness*), qui est liée à la perception du trustee par le trustor comme quelqu'un d'honnête et de transparent ;
- le **souci** (*concern*), qui correspond à la perception du trustee par le trustor comme quelqu'un de non dangereux et concerné, intéressé par les intérêts des autres en général ;
- la **fiabilité** (*reliability*), que Mishra définit en termes de cohérence entre les paroles du trustee et ses actes.

Le modèle de Mishra (1996) met en avant l'importance pour le trustor de prendre en compte certaines caractéristiques du trustee liées à des facteurs humains comme le souci, avant de décider de faire confiance. Cependant ces caractéristiques sont considérées à un niveau global, et il manque dans ce modèle la prise en compte de la relation personnelle qui peut lier le trustee au trustor. Cette relation semble pourtant capitale : l'agent Béa peut ne pas vouloir aider l'agent Anna mais bien vouloir aider l'agent Cara. Il faut donc qu'Anna prenne en compte non pas une tendance générale ou moyenne de Béa à bien vouloir aider les autres, mais la volonté de Béa l'aider personnellement. De même, nous nous plaçons dans le cadre de la confiance située et considérer un niveau de compétence général sans prendre en compte la spécificité de la tâche particulière pour laquelle on fait confiance n'est pas une solution satisfaisante pour notre problématique.

Mayer et al. (1995) : compétence, intégrité et bienveillance. Dans le domaine de l'étude du comportement organisationnel, Mayer et al. (1995) ont établi un modèle de confiance interpersonnelle à partir d'une revue des modèles déjà proposés dans la littérature en psychologie sociale. Des validations expérimentales de ce modèle ont par la suite été conduites, notamment par Jarvenpaa et al. (1998) ou Gill et al. (2005). Le modèle de confiance interpersonnelle de Mayer et al. (1995) a en outre été repris⁹ dans le domaine de l'ergonomie cognitive par Karsenty (2010). D'après les auteurs, l'intention de faire confiance du trustor s'appuie en partie sur une évaluation des caractéristiques suivantes du trustee :

- la **compétence**. L'évaluation de la compétence du trustee par le trustor se fait par rapport à un domaine spécifique : on ne fait pas référence à une compétence générale mais à une compétence pour une catégorie de tâches, qui correspondent à un domaine

9. Karsenty (2010) a repris et complété le modèle de Mayer et al. (1995) en spécifiant l'influence de la confiance en soi du trustor sur sa décision de faire confiance au trustee.

d'expertise spécifique.

- l'**intégrité**. La perception de l'intégrité du trustee par le trustor est décrite par Mayer et al. (1995) comme le sentiment du trustor que le trustee adhère à des valeurs qu'il juge acceptables. L'intégrité est également liée à la régularité du comportement du trustee et à la cohérence entre ses paroles et ses actes. Cette dimension semble donc regrouper les dimensions de fiabilité et d'ouverture d'esprit de Mishra (1996).
- la **bienveillance**. Le trustor percevra le trustee comme bienveillant s'il a l'impression que le trustee veut son bien. Cette dimension fait référence à un lien spécifique et personnel entre le trustee et le trustor, et a donc plus de poids lorsque le trustor et le trustee se connaissent bien, lorsque leur relation a eu le temps de se développer.

Les auteurs ajoutent également à leur modèle une dimension spécifique au trustor (et non à la relation trustee-trustor) : la **disposition du trustor à faire confiance** au trustee. Cette dimension est relative à une tendance générale à faire confiance et est particulièrement importante dans le cas où le trustor ne connaît pas le trustee. D'autres auteurs incluent également cette dimension dans leur modèle de confiance (Karsenty, 2010; McKnight and Chervany, 2001; Yamagishi and Yamagishi, 1994).

Dans ce modèle, la confiance est située : pour un même trustee, elle varie selon la situation, de par la prise en compte de la compétence du trustee. Le trustor prend par ailleurs en compte le lien personnel qui le lie au trustee grâce à la dimension de la bienveillance, ce qui n'est pas le cas par exemple dans le modèle de Mishra (1996). En d'autres mots, avec un tel modèle de confiance, l'agent Anna fait confiance à l'agent Béa pour ranger les bouteilles d'oxygène avec elle si elle pense que Béa est compétente pour cette tâche et si elle pense que Béa est bienveillante envers elle. La dimension d'intégrité permet aux agents de faire des prédictions sur les membres de l'équipe qui tiendront parole et respecteront leurs engagements en rapport au but collectif.

Ce modèle de confiance interpersonnelle répond donc à notre problématique : la dimension de la compétence permet d'associer un niveau de confiance à une tâche particulière. Les agents pourront tenir compte dans leur décision de l'intégrité des autres, qui est l'un des facteurs de confiance les plus souvent cités (Deutsch, 1958; Mishra, 1996). Les liens personnels des agents influenceront également leur décision grâce à la dimension de la bienveillance. Dans la suite de ce manuscrit, nous désignerons ce modèle par le terme "modèle MDS" (du nom de ses auteurs, Mayer, Davis, and Schoorman).

4.3.2 Représentation de la confiance

Ayant choisi le modèle de confiance que nous souhaitons utiliser pour construire notre moteur décisionnel, la question à se poser est maintenant la suivante : comment définir les croyances des agents sur les dimensions d'intégrité, de bienveillance et de compétence des autres ? Autrement dit, comment représenter le fait qu'Anna a confiance en Béa pour ranger l'ambulance ? Ou qu'Anna n'a pas confiance en Béa pour ranger l'ambulance ? Nous avons également vu dans la section 4.1.2 que Deutsch (1958) notamment emploie le terme de méfiance en opposition à la confiance. La méfiance est-elle alors équivalente à l'absence de confiance ? Doit-on avoir des concepts différents pour représenter ces trois éléments ? Nous abordons ces questions ici.

Confiance et méfiance, deux notions distinctes

Certains auteurs considèrent la confiance et la méfiance comme deux notions distinctes et non liées (Luhmann, 2000; McKnight and Chervany, 2001; Rousseau et al., 1998). Lewicki et al. (1998) donnent l'exemple suivant pour appuyer leur point de vue : deux chercheurs A et B collaborent pour produire un papier. Le chercheur A doit effectuer l'analyse des résultats, alors que B a la responsabilité de l'écriture. Si A ne produit pas l'analyse des résultats, alors B ne sera pas *complètement* méfiant vis-à-vis de A, par exemple B ne soupçonnera pas A de chercher à l'empoisonner s'il lui propose un café. Selon ces auteurs, avoir peu confiance est différent de se méfier, et ne pas se méfier est différent d'avoir confiance. Suivant cette conception des choses, on peut à la fois avoir confiance et à la fois se méfier de la même personne, et un équilibre entre confiance et méfiance serait l'idéal à atteindre.

Confiance et méfiance, deux opposés d'un même continuum

Comme nous l'avons déjà vu section 4.1.2, Deutsch (1958) met en opposition la méfiance (*suspicion*) à la confiance : il définit la confiance comme l'attente d'un événement qui aura des conséquences positives, et la méfiance comme l'attente d'un événement qui aura des conséquences négatives. De manière similaire, Marsh and Briggs (2009) définissent la méfiance comme liée à des mauvaises intentions de la part du trustee envers le trustor, en opposition aux bonnes intentions lorsque l'on parle de confiance. Gambetta (2000) définit également la méfiance (*distrust*) comme l'opposé symétrique de la confiance. Selon Gambetta, la confiance du trustor en le trustee peut être représentée par une valeur numérique comprise entre 0 (méfiance totale) et 1 (confiance totale). Le point central 0,5 correspond alors à une absence de confiance et une absence de méfiance. D'autres auteurs partagent cette même définition de la méfiance comme un opposé de la confiance (Castelfranchi and Falcone, 2010; Mayer et al., 1995; Schoorman et al., 2007).

Vers une réconciliation

Les deux visions de la méfiance comme un opposé de la confiance ou comme une notion distincte semblent incompatibles, mais cette différence d'opinion est en fait plus due à des désaccords sur la définition même de la confiance que sur celle de la méfiance. En effet, comme nous l'avons observé section 4.2, il n'y a pas de consensus sur le caractère situé ou non situé de la confiance. Or les auteurs qui argumentent pour considérer confiance et méfiance comme deux notions distinctes adoptent une vision de la confiance comme étant non située, c'est-à-dire que le trustor fait confiance au trustee de manière générale sans considérer une tâche particulière (Lewicki et al., 1998; McKnight and Chervany, 2001). Cela explique le besoin de différencier les notions de confiance et méfiance : il est nécessaire que le chercheur B, qui faisait confiance à A pour analyser les résultats, puisse être à la fois méfiant envers A (e.g. pour le prochain papier qu'ils envisageront d'écrire ensemble) et confiant envers A (e.g. pour ne pas empoisonner son café).

A l'inverse, la plupart des auteurs qui adoptent une vision de la confiance et de la méfiance comme des opposés d'un même continuum considèrent la confiance comme située, c'est-à-dire que le trustor fait confiance au trustee pour une tâche précise (Castelfranchi and Falcone, 2010; Deutsch, 1958; Mayer et al., 1995). Selon cette définition, il est donc possible pour B de ne

pas faire confiance à A pour analyser des résultats de recherche, tout en continuant à lui faire confiance pour ne pas l’empoisonner. Si l’on accepte cette définition de la confiance comme nécessairement située, il y a peu d’intérêt à avoir deux éléments différents pour désigner la confiance et la méfiance (Mayer et al., 1995), d’autant plus que même lorsque confiance et méfiance sont considérées comme deux notions séparées, les deux concepts sont définis par les mêmes termes et sont mis en opposition : la confiance/méfiance est une attente positive/négative du trustor vis-à-vis du trustee (Lewicki et al., 1998).

Nous avons choisi pour répondre à notre problématique de nous appuyer sur le modèle de confiance interpersonnelle MDS, et nous adoptons une définition de la confiance comme nécessairement située. Dans ce cadre, et en accord avec le point de vue des auteurs de ce modèle, nous considérons la méfiance comme l’opposé de la confiance sur un même continuum (Schoorman et al., 2007). En outre, cette conception de la confiance et de la méfiance correspond à celle classiquement utilisée dans le domaine des SMA (i.e. les agents calculent une seule valeur de confiance et non deux valeurs pour la confiance et la méfiance respectivement).

Par ailleurs, en nous appuyant sur la proposition précise de Gambetta (2000), nous suggérons de représenter par trois valeurs comprises entre 0 et 1 la confiance de l’agent x respectivement en l’intégrité de l’agent y , la bienveillance à son égard de l’agent y et la compétence de l’agent y sur la tâche τ . Ainsi, "Anna se méfie des intentions de Béa vis-à-vis d’elle" correspondrait alors à "Anna pense que la valeur de bienveillance de Béa vis-à-vis d’elle est inférieure à 0.5".

4.3.3 Conclusion

Nous nous sommes intéressés ici aux fondements de la confiance, c’est-à-dire aux raisons que peut avoir le trustor de faire confiance au trustee. Nous avons d’abord abordé la motivation du trustee à réaliser une tâche. Si le trustee est motivé pour accomplir une tâche alors le trustor peut lui faire confiance pour cette tâche (à condition que le trustee soit compétent).

Les auteurs qui proposent des modèles dimensionnels de la confiance cherchent principalement à répondre à la question suivante : comment le trustor peut-il évaluer la motivation du trustee ? Quelles-sont les caractéristiques du trustee importantes à prendre en compte ? De nombreux modèles de confiance ont été produits dans l’objectif de répondre à cette question. En particulier, le modèle de Mayer et al. (1995) identifie les dimensions d’intégrité, de bienveillance et de compétence. Ce modèle multidimensionnel de la confiance interpersonnelle offre un cadre théorique approprié à nos travaux, en particulier parce qu’il considère la confiance comme située. De plus, chaque agent pourra raisonner à la fois sur ses croyances sur les buts des autres et sur leurs dimensions d’intégrité, de bienveillance et de compétence pour déterminer leurs motivations. En particulier, la dimension de bienveillance offre une réponse appropriée à la problématique de l’adoption altruiste de but.

Enfin, nous avons abordé, au travers de l’étude de la méfiance, la question de la représentation de la confiance. Confiance et méfiance sont parfois considérées comme deux notions distinctes ; cependant cette conception des choses ne semble pas être appropriée pour nos travaux. En effet, étant donné que nous adoptons une vision de la confiance comme nécessairement située, on peut distinguer plusieurs niveaux de confiance pour une même paire trustee-trustor. Conformément à ce qui est fait dans la littérature sur les SMA, nous adoptons une vision de la confiance et de la méfiance comme deux opposés d’un même continuum.

4.4 Bilan du chapitre

Nous avons étudié dans ce chapitre les travaux sur la confiance dans le domaine des SHS. Cette étude nous a permis de mieux appréhender cette notion complexe de la confiance et d’aborder certains concepts qui y sont liés et qui sont intéressants par rapport à notre problématique de génération de comportements de personnages virtuels.

Nous avons dans un premier temps abordé l’influence de la confiance sur les comportements de groupe. Nous avons vu que si la confiance est souvent associée à une amélioration de la performance des groupes, cela n’est pas toujours le cas. Pour comprendre pourquoi, nous nous sommes intéressés à l’influence de la confiance sur les comportements individuels. Nous avons notamment constaté que la confiance est liée à la **délégation faible** : le trustor n’a pas d’accord explicite avec le trustee à propos de la tâche qu’il souhaite déléguer. Ce type de délégation nous intéresse particulièrement, puisqu’elle repose sur la prise en compte du trustee dans les plans du trustor et permet notamment aux agents d’une équipe de s’organiser rapidement et implicitement. Ce type de délégation n’est, à notre connaissance, jamais traité par les systèmes de confiance et/ou de réputation dans le domaine des SMA. L’étude de l’influence de la confiance sur les comportements nous a également permis d’aborder la notion de **confiance mal placée** : la confiance mal placée du trustor en le trustee conduit le premier à faire de mauvais choix, ce qui nuit à la performance du groupe.

Nous nous sommes ensuite intéressés aux différentes définitions de la confiance que l’on peut trouver dans la littérature en SHS. Plusieurs formes de confiance peuvent être distinguées et parmi ces différentes formes de confiance, nous avons identifié celle qui nous intéresse : la **confiance sociale interpersonnelle**, qui caractérise la relation de confiance entretenue par deux individus. Cette confiance peut être définie comme située ou non située : afin de pouvoir distinguer plusieurs confiances différentes pour une même paire trustee-trustor selon la tâche considérée, nous adoptons une définition de la confiance comme nécessairement **située**. D’autre part, s’il n’y a pas de consensus sur une définition de la confiance, nous avons remarqué que la notion de confiance est très souvent associée à la notion de croyance sur le futur. Nous avons alors établi que pour pouvoir former des croyances sur le futur, nos agents doivent être dotés de **théorie de l’esprit**, c’est-à-dire être capables de raisonner sur un modèle mental des autres.

Enfin, nous nous sommes intéressés à l’origine et à la justification de la confiance du trustor en le trustee. Nous avons notamment vu que le trustor a confiance en le trustee car il pense que ce dernier va adopter l’un de ses buts. L’**adoption altruiste** de buts en particulier nous permettra de générer des comportements d’aide spontanée. Nous avons alors étudié les modèles de confiance qui s’intéressent aux caractéristiques du trustee. Nous choisissons d’appuyer notre proposition sur le modèle de confiance proposé par Mayer et al. (1995), qui adopte une définition de la confiance comme située, et qui contient notamment la dimension de la bienveillance : si le trustor pense que le trustee est bienveillant envers lui, alors il pense que ce dernier pourrait adopter ses buts de manière altruiste.

Le modèle d’agents et le moteur décisionnel que nous proposons seront décrits dans le chapitre 6. Avant cela, nous abordons dans le chapitre suivant les problématiques liées au modèle d’activité collective sur lequel vont raisonner les agents.

Chapitre 5

Activité collective

Les environnements virtuels sont très utilisés pour la formation, et nous souhaitons pouvoir les utiliser pour entraîner les capacités d'adaptation de l'apprenant à différentes équipes. L'apprenant sera alors entouré de personnages virtuels autonomes. Cependant, les activités sur lesquelles l'apprenant et les personnages virtuels vont intervenir sont complexes : par exemple, une équipe de secours doit intervenir pour soigner des victimes. Il y a un grand nombre d'actions à faire, qui doivent se faire dans un certain ordre ; e.g. avant d'intervenir sur une victime, il faut sécuriser la zone d'intervention. Il y a donc une forme de procédure à respecter, même s'il existe souvent plusieurs alternatives. C'est pourquoi nous avons besoin de pouvoir décrire l'activité des personnages autonomes. Cette description correspond à une spécification fonctionnelle (voir 2.2.2) : les agents sont contraints dans la manière d'atteindre leur but.

Il existe de très nombreux formalismes de description de l'activité, et nous choisissons d'utiliser le langage ACTIVITY-DL. Nous exposons dans la section 5.1 les raisons de ce choix. ACTIVITY-DL ne permettant pas la description d'activités *collectives*, nous nous intéressons dans la section 5.2 aux éléments à ajouter à ACTIVITY-DL pour remédier à ce problème.

Ayant défini le langage et les éléments de ce langage qui serviront à la description des activités collectives, nous devons ensuite traiter notre principal problème : comment les agents vont-ils raisonner sur la description de l'activité pour sélectionner une action à faire ? De quelles informations auront-ils besoin ? Nous présentons dans la section 5.3 un aperçu du problème lié au raisonnement des agents ainsi qu'un aperçu de la solution proposée en termes de traitement effectué sur la description de l'activité. Enfin nous détaillons nos contributions à ce sujet dans les sections 5.4 et 5.5.

5.1 Formalisme de description de l'activité

Proposer un nouveau langage de description de l'activité n'est pas notre objectif ici, mais nous avons besoin d'en choisir un qui nous permette de supporter le raisonnement de nos agents.

Il existe de très nombreux formalismes de description de l'activité que nous passons très rapidement en revue de manière à justifier le choix du formalisme que nous utilisons. Nous pourrions ainsi, dans la section suivante, proposer un mécanisme de propagation de contraintes et d'informations qui permettra de faciliter le raisonnement des agents.

5.1.1 Besoins

Nous détaillons ici les besoins que nous avons par rapport au langage de description de l'activité de manière à mieux comprendre le choix du langage de description.

Structure arborescente

Les agents ont des buts, et pour réaliser leurs buts ils devront raisonner sur le modèle d'activité : quelle action ou quel enchaînement d'actions leur permet de réaliser leurs buts ? Par exemple, si une équipe de pompiers a pour but de porter secours à une personne dont la maison s'est effondrée, comment doivent-ils procéder ? La manière la plus naturelle de répondre à cette question est de décomposer le but en sous-buts : les pompiers doivent dans un premier temps sécuriser la zone avant de porter secours à la victime. Pour porter secours à la victime, il faudra la libérer des décombres puis évaluer son état en faisant la première phase de soins. La suite de l'intervention dépendra de l'état de la victime.

On obtient donc un arbre de buts / sous-buts ou de tâches / sous-tâches¹, dont les feuilles sont des actions à réaliser pour réaliser le but parent ou la tâche parente.

Une telle structure arborescente de l'activité permet de plus aux agents de raisonner sur des tâches abstraites, ce qui a deux avantages principaux : (i) il est pratique pour un agent de pouvoir raisonner sur des tâches de haut niveau (e.g. savoir que la zone est sécurisée suffit, il n'est pas nécessaire de savoir comment elle a été sécurisée – couper le compteur électrique, s'assurer que la structure ne va pas plus s'effondrer, etc.) et (ii) il est également plus facile de se répartir les tâches lorsque les actions peuvent être regroupées dans une hiérarchie de tâches (e.g. quelqu'un peut avoir la charge de la tâche de haut niveau *sécuriser la zone*, il n'y a alors pas besoin d'énumérer toutes les actions que cette personne devra faire).

Enfin, les structures arborescentes sont classiquement utilisées pour la génération de comportements d'équipe dans le domaine des systèmes multi-agents (SMA). Les mécanismes que nous proposons pour manipuler la description de l'activité pourront donc être adaptés à d'autres langages, comme par exemple à *MOISE*⁺.

Activité collective

Bien que nous travaillions sur des activités collectives, nous ne souhaitons pas attribuer a priori un rôle à chaque agent : les agents devront s'organiser dynamiquement pour choisir quoi faire. Cela offre une plus grande souplesse à l'équipe de personnages virtuels et cette souplesse est nécessaire au vu de nos objectifs : pour entraîner l'apprenant à s'adapter à différents types d'équipe et à prêter attention aux personnes qui l'entourent, il est nécessaire de permettre une répartition flexible des tâches de l'activité collective.

En outre, décrire l'activité sous forme de rôles implique de définir au préalable combien de rôles vont intervenir (même si en théorie plusieurs agents et/ou apprenant peuvent avoir le même

1. Les deux termes "tâche" et "but" sont utilisés dans la littérature. Selon les domaines, on emploiera plutôt l'un ou plutôt l'autre : en ergonomie (et dans les langages inspirés de travaux en ergonomie type *ACTIVITY-DL*), on emploiera plutôt le terme "tâche", alors que dans le domaine des SMA parlera plutôt de "but". Pour éviter la confusion dans les termes, nous choisissons dans nos travaux d'utiliser le terme de "tâche". En effet, nous considérons que l'on peut décrire une activité qui ne correspondrait pas au but d'un agent, par exemple si l'on souhaite décrire l'activité que l'apprenant doit réaliser.

rôle, dans la pratique le plus souvent un rôle est attribué à un seul agent) et lors de l'exécution de l'activité, c'est toujours le même scénario qui est obtenu à quelques variations près². En autorisant une répartition plus souple des tâches, une multitude de scénarios peut être obtenue à partir de la même description de l'activité.

Autres besoins : expressivité, intelligibilité

Nous souhaitons utiliser le langage de description de l'activité pour la génération du comportement des personnages virtuels, et nous avons exposé nos besoins en rapport avec cette problématique. Toutefois, il ne faut pas perdre de vue le cadre plus large dans lequel s'inscrivent nos travaux : les environnements virtuels pour la formation. La description de l'activité ne sert pas uniquement à la génération des comportements des personnages virtuels autonomes mais peut aussi servir au suivi de l'apprenant. Dans ce contexte, il est intéressant de pouvoir représenter non pas seulement l'activité prescrite mais également l'activité observée sur le terrain (i.e. écarts par rapport à la procédure). De nombreux autres éléments sont à prendre en compte pour la description de l'activité : par exemple, on doit pouvoir décrire de manière précise l'ordonnancement temporel des tâches (i.e. quelles tâches doivent être réalisées simultanément, séquentiellement, de manière cyclique, etc.), ou les conditions dans lesquelles une tâche doit être préférée à une autre (i.e. pour prendre le pouls de la victime, prendre le pouls au poignet est préférable par rapport à prendre le pouls à la carotide si la victime n'est pas en état critique), etc. Bien que nous ne nous intéressions pas à tous ces éléments dans le cadre de cette thèse, ils ne sont pas à négliger pour autant : nous avons besoin d'un langage qui soit très expressif. En effet, ces informations pourraient être utilisées par d'autres modules du système de formation en environnement virtuel, comme un module de suivi de l'apprenant : on pourrait par exemple évaluer si l'apprenant a bien respecté la préférence de prendre le pouls au poignet si la victime n'est pas en état critique.

D'autre part, ce ne sont pas des informaticiens qui vont décrire l'activité, mais des spécialistes du domaine ou des ergonomes car ce sont eux qui sont les plus à même de le faire : le langage doit donc être intelligible pour des non-informaticiens. Cette intelligibilité peut être apportée par un éditeur graphique par exemple.

5.1.2 De nombreuses possibilités pour la description de l'activité

Il existe de très nombreuses manières de décrire l'activité qui doit être effectuée par les agents. Nous résumons ici les principaux formalismes qui sont utilisés en explicitant les avantages et inconvénients de chacun par rapport à nos besoins, ce qui nous permettra de motiver le choix du langage employé. Pour plus de détails sur les formalismes de description de l'activité, nous renvoyons le lecteur aux travaux de Barot (2014) qui propose un état de l'art détaillé des formalismes de description de l'activité en environnements virtuels (voir Chapitre 2 *Représentation des connaissances pour la scénarisation*) ainsi qu'aux travaux de Fan and Yen (2004) qui répertorient quelques formalismes utilisés dans les SMA.

2. Parfois des alternatives entre plusieurs actions sont précisées dans la description de l'activité et on obtiendra des scénarios différents selon l'alternative choisie. Cependant la liberté d'organisation des agents reste limitée.

Règles

On peut décrire l'activité avec des règles logiques de la forme *SI condition(s) ALORS conséquence(s)* comme cela est fait par Szilas (2007).

Ce type de description n'impose pas d'associer un rôle à chaque action et offre une grande souplesse. Rien ne l'impose, mais il serait possible de décrire une arborescence de tâches par ce type de règle, cependant cela serait un travail complexe car un tel formalisme de description est difficile à appréhender dans sa globalité (i.e. difficulté pour gérer plusieurs centaines de règles). Enfin sans éditeur spécifique, ce type de langage n'est pas intelligible pour des non-informaticiens (i.e. difficulté d'exprimer clairement les conditions, de faire le lien avec les objets du monde virtuel, etc.).

Diagrammes d'activité

Des langages de description de l'activité type UML peuvent être utilisés, comme par exemple le langage HAVE du framework MASCARET (Chevaillier et al., 2012) ou les Business Process Model Notation diagrams utilisés par Sanselone et al. (2014). Dans ces deux langages l'activité est découpée sous forme de rôles, chaque rôle étant décrit par une séquence d'actions à faire. Il est possible de préciser que des actions doivent être synchronisées, et il est également possible d'exprimer des alternatives.

Ce type de langage de description de l'activité a l'avantage d'être relativement simple à appréhender, cependant pour les deux langages étudiés il est nécessaire pour chaque agent d'avoir un rôle prédéfini. Par ailleurs, ces langages ne permettent pas de structurer la description de l'activité en une hiérarchie de tâches. Pour cela, il faudrait ajouter certaines spécificités au langage, ce qui rend le formalisme de description plus complexe à manipuler (Barange, 2015, partie 6.3).

Automates à états finis et réseaux de Pétri

Il est possible d'utiliser des automates à états finis ou des réseaux de Pétri pour décrire l'activité. Les actions correspondent alors aux états et les transitions représentent l'ordonnancement temporel des actions ou des liens de causalité entre les actions.

Ce type de description peut être adapté pour permettre une décomposition hiérarchique de l'activité : chaque état correspond alors soit à une action, soit à un sous automate. Le langage LORA – *Language for Object Relation Application* par exemple permet une telle description. LORA++ (Gerbaud, 2008) prend de plus en compte les aspects collectifs qui étaient ignorés dans LORA en offrant la possibilité de distinguer les actions individuelles des actions collectives. Contrairement aux langages type UML, l'activité n'est pas découpée sous forme de rôles ; cependant, des rôles peuvent être précisés sur les actions.

LORA/LORA++ est un langage assez expressif pour décrire l'activité de personnages virtuels. Cependant bien qu'il soit possible de décomposer hiérarchiquement une tâche, la structure obtenue n'est pas un arbre, et l'explicitation de toutes les transitions entre les états représente un travail difficile.

Opérateurs de planification

L'activité peut être décrite sous forme d'opérateurs de planification : chaque opérateur correspond alors à une action de base qui peut être réalisée par un agent. Les types d'opérateurs les plus souvent utilisés sont les opérateurs STRIPS (Pizzi and Cavazza, 2007) ou les opérateurs PDDL (Porteous et al., 2013). Ces opérateurs sont généralement de la forme $\langle \text{nom}, \text{pré-conditions}, \text{effets} \rangle$.

Les opérateurs peuvent être organisés hiérarchiquement pour former un arbre dont la racine est le but à atteindre, les nœuds sont des sous-buts et les feuilles sont des actions à accomplir. Les *effets* d'un opérateur composé sont alors ses sous-opérateurs. C'est par exemple le formalisme utilisé pour STEAM (voir 2.2.2). On peut alors parler de réseau hiérarchique de tâches (*hierarchical task network* – HTN). Les HTN sont utilisés par exemple par Rickel and Johnson (1999) ou par Cavazza et al. (2002) dans le domaine des environnements virtuels ou dans RETSINA-MAS (Giampapa and Sycara, 2002) dans le domaine des SMA pour la simulation de comportements d'équipes.

Une telle description de l'activité par des opérateurs a l'avantage d'être interprétable par un moteur de planification. Par ailleurs, l'utilisation d'une structure hiérarchique permet d'une part aux spécialistes du domaine d'organiser plus facilement la description de l'activité, et d'autre part aux agents de raisonner plus facilement sur des tâches de haut niveau. Cependant le langage est difficile d'accès à des non-informaticiens et son expressivité est limitée.

Langages ergonomiques

Il est également possible d'utiliser des langages ergonomiques (i.e. utilisés par des ergonomes et/ou nés de travaux en ergonomie) de description de l'activité. Les langages ergonomiques sont nombreux (Couix and Burkhardt, 2011) et ont l'avantage d'être accessibles à des non-informaticiens ; par ailleurs ils sont étudiés pour représenter au mieux l'activité humaine et sont donc très expressifs. À notre connaissance, un seul de ces langages a été utilisé en environnement virtuel pour la génération de comportements d'agents autonomes : ACTIVITY-DL.

ACTIVITY-DL permet une description précise de l'activité grâce aux éléments qui peuvent être associés à chaque tâche (e.g. différents types de pré-conditions, nombreuses possibilités d'ordonnement temporel, possibilité de spécifier les pratiques non réglementaires mais tolérées par l'usage, etc.). Ce langage peut être utilisé par des non-informaticiens grâce à un éditeur graphique. Enfin ACTIVITY-DL offre une représentation hiérarchique de l'activité sous forme d'un arbre de tâches.

5.1.3 Description de l'activité avec ACTIVITY-DL

Nous avons exposé notre préférence pour les langages de description de l'activité qui permettent d'obtenir une description de l'activité sous forme de structure arborescente. Ces représentations sont en effet classiquement utilisées dans les SMA et dans les environnements virtuels. Elles permettent d'appréhender facilement des activités de taille conséquente (i.e. comportant un grand nombre de tâches à effectuer) ; et la structure hiérarchique permet aux agents de raisonner sur des tâches abstraites. Cela limite la complexité du raisonnement en évitant d'avoir à raisonner sur plusieurs centaines d'actions. Par ailleurs nous souhaitons ne pas décrire l'activité

sous forme d'un ensemble de rôles, car cela limite grandement les possibilités de répartition des tâches dans le cas d'une activité collective : le langage ACTIVITY-DL répond à ces besoins.

ACTIVITY-DL est un langage qui permet de décrire l'activité en XML (voir annexe B.2) sous forme d'un arbre de tâches. Chaque tâche est alors (i) une tâche abstraite, auquel cas elle a des sous-tâches, ou (ii) une tâche concrète (on parlera d'action) et correspond à une tâche feuille. Comme nous l'avons déjà établi, ACTIVITY-DL est un langage très expressif. Nous allons détailler ici uniquement les éléments dont nous nous servons spécifiquement dans cette thèse. Barot (2014) propose une description complète du méta-modèle ACTIVITY-DL (voir Chapitre 5 : *Contenu scénaristique*).

Ordonnement temporel et logique des tâches

Il est possible de spécifier deux types de contraintes sur les sous-tâches d'une même tâche mère :

- Les contraintes logiques permettent d'exprimer la condition de validation de la tâche mère par rapport à ses tâches filles. La contrainte **AND** spécifie que toutes les tâches filles doivent être réalisées pour valider la tâche mère et **OR** spécifie qu'une seule des tâches filles a besoin d'être faite pour valider la tâche mère.
- Les contraintes temporelles permettent d'ordonner dans le temps la réalisation des sous-tâches. Si la tâche mère a la contrainte **SEQ-ORD**, alors les sous-tâches doivent être réalisées séquentiellement dans un ordre prédéfini ; avec la contrainte **PAR-SIM**, les sous-tâches doivent être réalisées simultanément (i.e. de manière synchronisée du début de l'exécution de la tâche jusqu'à la fin), c'est-à-dire qu'il faut être synchronisé pour réaliser ces sous-tâches. Enfin **IND** signifie qu'il n'y a pas de contrainte temporelle sur l'exécution des sous-tâches : elles peuvent être réalisées en séquence ou simultanément.

Les contraintes logiques et temporelles peuvent être combinées. Nous travaillons dans le cadre de cette thèse avec les combinaisons suivantes :

- **ET/SEQ-ORD** : les sous-tâches doivent toutes être effectuées en séquence. Cette combinaison correspond notamment à l'opérateur *séquence* de $MOISE^+$ (voir 2.2.2).
- **ET/PAR-SIM** : les sous-tâches doivent toutes être effectuées simultanément. Cette combinaison correspond notamment à l'opérateur *parallélisme* de $MOISE^+$ (voir 2.2.2).
- **ET/IND** : les sous-tâches doivent toutes être effectuées mais il n'y a pas de contrainte temporelle.
- **OU/SEQ-ORD** : une seule des sous-tâches doit être réalisée pour valider la tâche mère, cependant il est nécessaire de faire les sous-tâches dans l'ordre spécifié ; c'est-à-dire que l'on essaie d'abord de faire la première, dans le cas où on ne peut pas / on échoue, on essaie de faire la deuxième, etc. Cela permet d'exprimer un ordre de préférence sur les tâches. Par exemple, pour *prendre le pouls*, on essaie de *prendre le pouls au poignet*, si non ne peut pas, on essaie de *prendre le pouls à la carotide*.
- **OU/IND** : une seule des sous-tâches doit être réalisée pour valider la tâche mère, cependant il n'y a pas de contrainte sur le choix de la sous-tâche à faire. Cette combinaison correspond notamment à l'opérateur *choix* de $MOISE^+$ (voir 2.2.2).

ACTIVITY-DL est un langage très riche et nous ne traitons pas tous les éléments qui pourraient être exprimés dans ce langage :

- Nous ne traitons pas ici le cas OU/PAR-SIM : dans un tel cas, une seule des sous-tâches devrait être validée pour valider la tâche mère, mais on imposerait que toutes ces sous-tâches soient réalisées (ou que l'on essaie de les réaliser) de manière synchronisée, i.e. on ne peut pas essayer de faire l'une des sous-tâches tant que d'autres personnes n'essaient pas en même temps de faire les autres. Imposer une si forte contrainte n'a pas vraiment de sens ; de plus, du point de vue des agents, cela revient à traiter la tâche mère comme si elle avait la contrainte ET/PAR-SIM (i.e. répartition des sous-tâches et synchronisation).
- ACTIVITY-DL propose d'autres contraintes temporelles (il en existe actuellement sept et nous en traitons trois) et d'autres contraintes logiques (nous en traitons deux et il en existe quatre au total), et par souci de simplicité nous avons préféré dans nos travaux nous limiter à certaines contraintes. Néanmoins nous couvrons dans nos travaux les principaux cas. En effet, les possibilités de spécifier des contraintes logiques et temporelles dans la littérature sont généralement plus limitées que ce qui existe dans ACTIVITY-DL : classiquement, on traite le cas ET/SEQ-ORD, le cas ET/PAR-SIM et le cas OU/IND. C'est par exemple ce qui est fait dans MOISE⁺ (voir 2.2.2). Par ailleurs, les autres cas (à l'exception d'un seul) pourraient être traités avec les mécanismes proposés dans ce chapitre. Nous renvoyons le lecteur à l'annexe D pour plus de précisions à ce sujet.

Exemple

Un exemple d'arbre de tâches décrit en ACTIVITY-DL est donné dans la figure 5.1³, et la même figure est présentée en pleine page en annexe B.1. Cet arbre de tâches correspond à un scénario de sauvetage d'une personne (la victime) par une équipe de trois pompiers suite à l'effondrement de la maison de la victime. Dans cet arbre, les tâches sont numérotées (en haut à gauche). Pour désigner une tâche, on utilisera la notation $\tau_{numéro}$. Par exemple, la tâche *sauver la victime* est la tâche τ_1 .

Dans cet arbre, la tâche racine est $\tau_1 = \textit{sauver la victime}$. Cette tâche se décompose en trois sous-tâches qui doivent toutes être réalisées séquentiellement (ET/SEQ-ORD) : $\tau_{11} = \textit{partir en reconnaissance}$, $\tau_{12} = \textit{faire le dégagement d'urgence}$ et $\tau_{13} = \textit{faire la première phase de soins}$. Les tâches τ_{11} et τ_{13} sont elles-mêmes des tâches composées.

La tâche $\tau_{11} = \textit{partir en reconnaissance}$ est composée de deux sous-tâches qui doivent être toutes les deux réalisées mais n'ont pas de contrainte temporelle. Ces sous-tâches sont $\tau_{111} = \textit{localiser la victime}$ et $\tau_{112} = \textit{sécuriser la zone}$. *sécuriser la zone* peut être fait de deux manières (OU/IND) : il faut ou *éteindre le feu* (τ_{1121}) si l'incendie est déjà déclenché, ou *couper le compteur* (τ_{1122}) pour éviter qu'un incendie ne se déclenche⁴.

La tâche $\tau_{13} = \textit{faire la première phase de soins}$ est quant à elle composée de trois sous-tâches qui doivent être toutes les trois réalisées simultanément (ET/PAR-SIM). Ces sous-tâches sont $\tau_{131} = \textit{passer un message au poste de secours pour informer sur la situation}$, $\tau_{132} = \textit{maintenir}$

3. Cet exemple est inspiré du travail de l'étudiante et pompier volontaire Chloé Dejon qui, dans le cadre d'un projet tutoré en lien avec les travaux présentés dans ce manuscrit, a décrit un arbre de tâches en ACTIVITY-DL. Les travaux effectués au cours de ce projet ont également porté sur la représentation sous forme de workflow d'une activité décrite en ACTIVITY-DL. Le XML correspondant à cet exemple ainsi qu'une représentation de l'activité sous forme de workflow sont donnés en annexe B.2 et B.3.

4. En réalité il faudrait faire les deux (contrainte ET), mais pour les besoins de l'exemple on choisit une contrainte OU.

la tête dans l'alignement du tronc, car cela permet de soulager une éventuelle compression de la moelle épinière suite à un choc, et enfin $\tau_{133} = s'occuper\ de\ la\ victime$ pour établir un bilan.

Dans cet arbre, on a donc quatre tâches composées (τ_1 , τ_{11} , τ_{13} et τ_{112}) et sept tâches feuilles qui correspondent à des actions (τ_{111} , τ_{1121} , τ_{1122} , τ_{12} , τ_{131} , τ_{132} et τ_{133}).

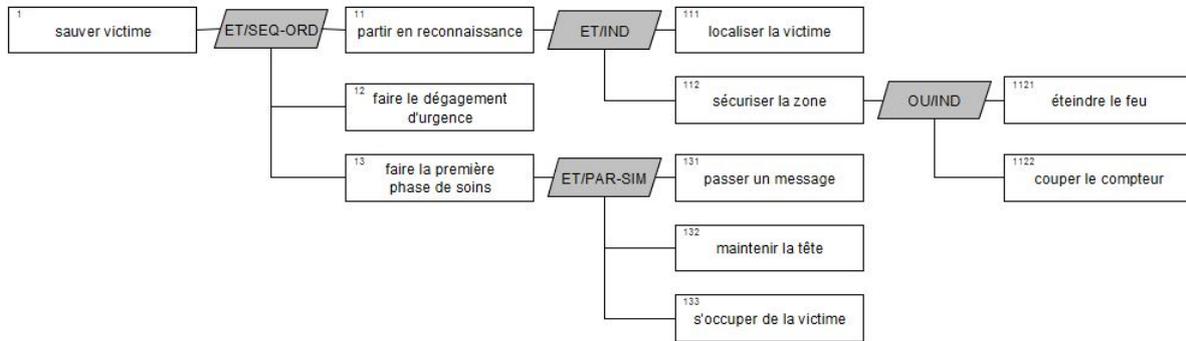


FIGURE 5.1 – Exemple d'un arbre de tâches décrit en ACTIVITY-DL.

5.2 Éléments liés aux aspects collectifs

Bien qu'ACTIVITY-DL soit très expressif, ce langage a, à l'origine, été pensé pour décrire l'activité d'une personne seule. HAWAII-DL, un langage de description de l'activité dont a été inspiré ACTIVITY-DL avait été utilisé pour décrire une activité collaborative entre deux personnes (Amokrane, 2010; Edward, 2011). Cependant, l'activité était alors découpée sous forme de rôles, et à chaque rôle était associé un arbre de tâches. Ici, nous souhaitons proposer un modèle qui permette une description où les rôles ne sont pas prédéfinis, et où on ne connaît pas nécessairement à l'avance le nombre de personnes exact qui vont participer à l'activité. En particulier, pour l'activité *sauver la victime*, on ne veut pas renseigner (*cas 1*) deux arbres de tâches pour le cas où il y aurait deux intervenants pompiers, et (*cas 2*) deux autres arbres de tâches pour le cas où il y aurait deux intervenants non pompiers, puis (*cas 3*) trois arbres de tâches pour le cas où il y aurait trois intervenants, etc. Nous souhaitons pouvoir décrire ces différents cas à l'aide d'un seul arbre de tâches. Nous devons donc ajouter à ACTIVITY-DL certains éléments pour que le langage supporte la description de telles activités.

D'après Gerbaud (2008), il existe plusieurs niveaux de collaboration dans les environnements virtuels de formation. La collaboration peut notamment se faire au niveau du scénario, c'est-à-dire que plusieurs personnes travaillent ensemble sur une activité collective mais en effectuant chacune des actions individuelles. La collaboration peut se faire également au niveau d'une action : certaines actions se font à plusieurs, par exemple *porter un brancard*.

Nous spécifions dans les paragraphes suivants les éléments dont nous aurons besoin pour décrire les activités collaboratives qui tiennent compte de ces deux niveaux de collaboration et sur lesquels peut s'appuyer le raisonnement des agents.

5.2.1 Collaboration au niveau du scénario

Pour mettre en place la collaboration au niveau du scénario et pour que les comportements des agents soient semblables à ceux observés sur le terrain, il peut être nécessaire de préciser certaines informations sur la description de l'activité. En particulier si l'on reprend l'exemple précédent, pour apporter des soins à la victime (action *s'occuper de la victime*), il faut avoir reçu une formation aux premiers secours. Cette information doit être précisée sur l'arbre de tâches, de manière à ce qu'un agent qui n'a pas cette compétence ne prenne pas cette tâche en charge.

Nous choisissons dans le cadre de nos travaux de parler de compétence : en effet, c'est le terme qui est utilisé dans le modèle de confiance de Mayer et al. (1995) sur lequel s'appuie notre moteur décisionnel. Nous faisons donc ici le lien entre ce modèle de confiance et la description de l'activité collective. Toutefois dans les environnements virtuels de formation ou dans le domaine des SMA, d'autres termes ou notions peuvent être utilisés ; notamment il est souvent question de *rôle*. La notion de rôle est cependant assez floue : parfois elle est associée à des obligations et/ou à des droits. Dans certains cas un rôle est associé à un but à accomplir (peu importe la manière dont le but est accompli), dans d'autres un rôle est associé à une séquence d'actions précise à exécuter. Parfois un personnage ne peut avoir qu'un seul rôle, parfois il peut en avoir plusieurs, et parfois il est possible de changer de rôle en cours de simulation, alors que dans d'autres cas cela n'est pas possible.

Nous faisons donc le choix de laisser de côté la notion de rôle. Nous utilisons une définition des compétences assez large (i.e. on peut parler de la compétence *savoir utiliser un tournevis* ou *avoir suivi une formation aux premiers secours*), et nous considérons que lorsqu'une compétence est précisée sur une action, il est nécessaire d'avoir cette compétence pour exécuter l'action. Nous avons vu dans le chapitre 2 qu'un système comme MOISE⁺ permet de spécifier des droits et des obligations (voir 2.2.2) : nous ne traitons ici que les obligations. Par ailleurs, nous imposons qu'une action demande au plus une compétence. Un exemple de description d'activité comportant des compétences nécessaires est donné plus loin (5.2.5).

Préciser les compétences nécessaires permettra donc aux agents de raisonner sur l'arbre de tâches et de sélectionner une action pour laquelle ils sont compétents. En prenant les autres en compte (i.e. *Mon coéquipier est-il compétent pour s'occuper de la victime ?*) les agents pourront s'organiser et la collaboration au niveau du scénario émergera.

5.2.2 Collaboration au niveau des actions

Le second niveau de collaboration se situe au niveau des actions. En effet, certaines actions se font à plusieurs, comme *porter un brancard*. Dans ce cas, les agents devront être synchronisés. Pour pouvoir faire l'action, les agents ont alors besoin de savoir combien de personnes doivent et peuvent faire l'action collective : il faut être au moins deux et il peut y avoir jusqu'à quatre personnes qui portent un brancard.

Actuellement, la collaboration au niveau des actions n'est pas prise en compte dans ACTIVITY-DL. Nous souhaitons ajouter à ACTIVITY-DL cette possibilité. Nous proposons donc, pour chaque action, de spécifier le nombre de personnes qui peuvent participer. Il faut alors distinguer le nombre minimum de personnes nécessaires pour faire l'action et le nombre maximum de personnes qui peuvent prendre part à l'action. Un exemple de description d'activité comportant

des nombres minimaux et maximaux d'intervenants est donné plus loin (5.2.5).

Note : nous considérons ici que si une action requiert une compétence particulière et plusieurs personnes pour l'exécuter, alors toutes les personnes doivent avoir cette compétence. Par exemple, si pour porter le brancard il fallait avoir reçu une formation aux premiers secours, alors les deux à quatre personnes qui portent le brancard doivent avoir cette compétence.

5.2.3 Spécification sur les tâches feuilles

Pour faciliter d'une part la description de l'activité par les spécialistes du domaine, et d'autre part pour garantir la cohérence des activités décrites, nous limitons la spécification de ces éléments (i.e. compétences et nombre de personnes) aux tâches *feuilles*, c'est-à-dire aux actions. En effet, spécifier ces éléments sur des tâches composées n'a parfois pas de sens ou est trop compliqué, e, particulier lorsque la tâche abstraite est de très haut niveau (i.e. lorsqu'un arbre de tâches comporte plusieurs dizaines voire centaines de tâches), il est très difficile de savoir quelles compétences exactement vont être nécessaires et quel est le nombre d'intervenants requis pour les tâches de haut niveau. On peut avoir un aperçu de la difficulté de la spécification de ces éléments sur des tâches abstraites avec une activité pourtant très simple comme *sauver la victime*.

De plus, on évite ainsi les incohérences éventuelles (e.g. spécifier sur une tâche composée qu'elle ne nécessite qu'une seule personne alors que l'une de ses sous-tâches requiert l'intervention de deux personnes) qui poseraient ensuite problème pour la génération des comportements des agents.

5.2.4 Notations utilisées

Nous expliquons dans les paragraphes suivants les notations que nous utilisons pour manipuler les informations de nombre de personnes et de compétences. Un récapitulatif des notations utilisées et de leur signification est proposé dans le tableau C.1 en annexe.

Nombre de personnes

Nous proposons de spécifier un nombre minimal et un nombre maximal de personnes qui peuvent faire la tâche τ , que nous notons respectivement $n_{min}(\tau)$ et $n_{max}(\tau)$. Nous ferons apparaître ces informations sur l'arbre de tâches sous la forme $n_{min}(\tau) \dots n_{max}(\tau)$.

Compétences

Nous désignons une compétence par le symbole κ ; on pourra alors dire que la tâche feuille τ_k requiert la compétence κ .

5.2.5 Exemple de description d'activité collective

Un exemple d'arbre de tâches sur lequel figurent des contraintes de nombre de personnes et de compétences est donné dans la figure 5.2. (Cette figure est également présentée en pleine page en annexe B.2.)

Pour ne pas alourdir inutilement l'arbre de tâches, nous proposons de considérer des valeurs par défaut de $1..INF$ où INF désigne $+\infty$ pour toutes les tâches feuilles (actions) sur lesquelles aucune information en termes de nombre de personnes n'est spécifiée (i.e. un nombre illimité de personnes peut participer). Nous ne faisons pas apparaître ces valeurs par défaut sur l'arbre de tâches. $\tau_{111} = \textit{localiser la victime}$ ne requiert donc aucune compétence et 1 à INF personnes peuvent participer.

En reprenant l'exemple précédent, nous expliquons les compétences et nombres de personnes nécessaires pour chaque action :

- $\tau_{1121} = \textit{éteindre le feu}$ nécessite d'avoir suivi la formation *Intervention Feu niveau 1* : on note **IF1** la compétence associée.
- $\tau_{1122} = \textit{couper le compteur}$ nécessite exactement une personne. Lors de l'intervention d'une équipe de secours, c'est le chef d'équipe (ayant suivi la formation *Gestion Opérationnelle et Commandement niveau 1*) qui est chargé de faire cette action : on note **GOC1** la compétence associée.
- $\tau_{12} = \textit{faire le dégagement d'urgence}$ nécessite exactement deux personnes qui doivent avoir suivi une formation aux premiers secours (*Prévention et Secours Civiques niveau 1*) : on note **PSC1** la compétence associée.
- Pour $\tau_{131} = \textit{passer un message}$ au poste de secours, il faut également avoir la compétence **PSC1**, car il est nécessaire de savoir quelles informations sont importantes à transmettre. Une seule personne peut faire cette tâche.
- Exactement une personne doit *maintenir la tête* de la victime (τ_{132}).
- Pour *s'occuper de la victime* et lui apporter les premiers soins (τ_{133}), il faut avoir la compétence **PSC1**. Exactement une seule personne peut faire cette tâche⁵.

Sur la figure 5.2, les éléments de nombre de personnes et compétences renseignés par les experts du domaine ou les ergonomes apparaissent dans les bulles bleues en bas à droite de chaque tâche feuille.

5.3 Raisonnement des agents sur l'arbre de tâches

Nous présentons dans les sections suivantes notre contribution portant sur la propagation de contraintes et d'informations des tâches filles à leur tâche mère. Nous expliquons ici dans un premier temps le besoin auquel cette proposition répond et nous apportons une vision globale du problème et de la solution.

5.3.1 Besoins

Nous illustrons dans un premier temps le problème par un exemple qui correspond à l'arbre de tâches que nous avons décrit figure 5.2.

5. La tâche *s'occuper de la victime* et faire le bilan pourrait en réalité être réalisée à deux voire trois personnes (Il est difficile d'être plus nombreux à intervenir sur la victime – pour prendre le pouls, vérifier la respiration, la tension, etc. – car l'espace est limité et de plus, il est préférable qu'une seule voire deux personnes récoltent les informations, car plus le nombre d'intervenants est grand, plus le risque de perte d'information augmente.); cependant pour les besoins de l'exemple, on impose qu'une seule personne puisse intervenir.

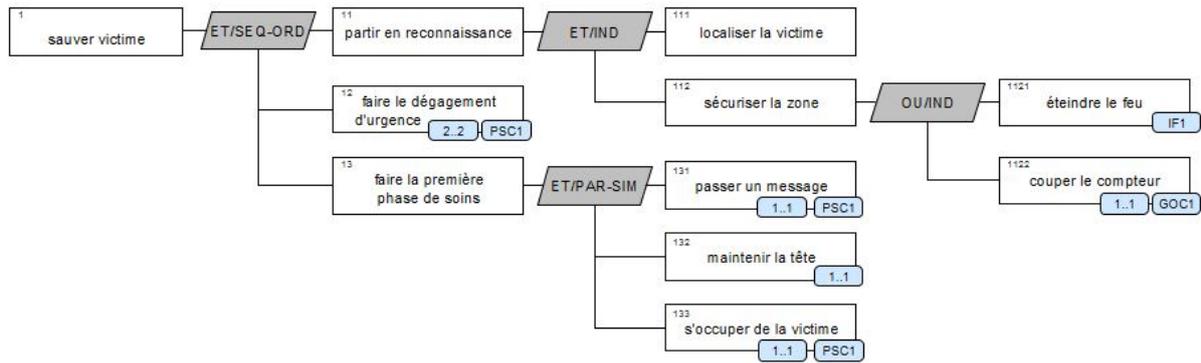


FIGURE 5.2 – Même exemple que dans la figure 5.1. Ici l’arbre de tâches décrit en ACTIVITY-DL comporte, dans les bulles bleues en bas à droite de chaque tâche feuille, les éléments de nombre de personnes et de compétences renseignés par les experts du domaine.

Activité réalisable

Prenons l’exemple suivant : deux agents jeunes pompiers débutants ayant seulement la compétence PSC1 doivent intervenir pour *sauver la victime*. Ils vont tout d’abord *partir en reconnaissance*, c’est-à-dire qu’ils vont devoir *localiser la victime* et *sécuriser la zone*. Pour sécuriser la zone, les agents doivent *éteindre le feu* (τ_{1121}) ou *couper le compteur* (τ_{1122}). Cependant, aucune de ces tâches ne pourra être accomplie car les agents n’ont ni la compétence GOC1 ni la compétence IF1 (il est trop risqué de faire une de ces deux tâches sans avoir été entraîné pour). Les agents ne pourront donc rien faire après avoir localisé la victime.

De la même manière, si l’on imagine deux agents pompiers expérimentés ayant les compétences PSC1, COG1 et IF1, ils pourront réaliser les tâches *partir en reconnaissance* et *faire le dégagement d’urgence*. Cependant, la tâche *faire la première phase de soins* a trois sous-tâches qui doivent être réalisées de manière synchronisée : les deux agents pompiers ne pourront encore une fois pas atteindre le but.

Participation d’un agent

Si aux deux agents pompiers expérimentés s’ajoute un témoin de la scène n’ayant aucune compétence particulière, les trois agents pourront sauver la victime. Cependant, la participation du témoin sera très limitée : une fois la victime localisée, le témoin ne peut (temporairement) plus participer à l’activité collective, puisqu’il est spécifié sur les tâches feuilles τ_{1121} et τ_{1122} que les compétences IF1 et GOC1 respectivement sont nécessaires. Le témoin pourra à nouveau participer à l’activité collective après le dégagement d’urgence, en maintenant la tête de la victime.

Enfin, si une seconde équipe de pompiers arrive en renfort pendant la première phase de soins, ces pompiers ne pourront pas participer à l’activité collective, puisque chaque sous-tâche de *faire la première phase de soins* admet au plus un participant.

Raisonnement des agents

Pour éviter ce genre de situation où, en raisonnant sur les tâches abstraites, les agents ne s'aperçoivent pas qu'ils ne pourront pas atteindre leur but, ou qu'ils ne peuvent pas participer à l'activité collective à l'instant t , il faudrait que les agents considèrent l'ensemble de l'arbre de tâches, i.e. il faudrait que les agents étudient l'ensemble du plan qui leur permettrait d'atteindre leur but. Cela conduirait les agents à raisonner parfois sur plusieurs centaines de tâches à la fois ; or, comme nous l'avons exposé en début de chapitre, le choix d'un langage qui permette de décrire l'activité comme une arborescence de tâches est principalement guidé par la possibilité qu'offre un tel langage de raisonner sur des tâches abstraites.

5.3.2 Aperçu de la solution proposée

Pour parer à ce problème, une solution est de faire remonter les informations pertinentes (e.g. avoir la compétence GOC1 ou la compétence IF1, être au moins trois, etc.) au niveau de la tâche racine *sauver victime*, de manière à ce que les agents sachent dès le départ s'ils sont assez nombreux et s'ils ont les compétences nécessaires pour atteindre leur but. Toutefois, faire remonter ces informations n'est pas trivial. En effet, il faut distinguer deux types d'information : dans notre exemple, une équipe dont aucun des membres n'a les compétences GOC1 et IF1 ne peut pas sauver la victime, mais un agent n'ayant pas ces compétences peut tout de même participer à l'activité collective.

Avant de sélectionner une tâche, un agent a donc besoin de pouvoir répondre à quatre questions : (1) *L'équipe a-t-elle les ressources nécessaires en termes de nombre de personnes pour faire la tâche ?* (2) *L'équipe a-t-elle les ressources nécessaires en termes de compétences requises pour faire la tâche ?* (3) *Puis-je encore participer à cette tâche ou bien le nombre maximal d'intervenants est-il déjà atteint ?* et (4) *Ai-je personnellement les compétences nécessaires pour participer à cette tâche ?*

Les questions 1 et 2 seront associées à ce que l'on appelle la ***contrainte de faisabilité collective*** en termes de nombre minimal de personnes et de compétences respectivement. Pour répondre à ces questions, l'agent devra raisonner sur l'ensemble de l'équipe, c'est-à-dire prendre en compte non seulement ses propres compétences mais aussi celles des autres. Si la réponse à l'une de ces deux questions est négative, l'activité telle qu'elle est décrite ne pourra pas être réalisée⁶.

Les questions 3 et 4 seront associées à ce que l'on appelle les ***informations dynamiques*** en termes de nombre maximal de participants et de compétences. Ces informations serviront, pour chaque agent qui envisage de participer à l'activité collective, à savoir s'il peut y participer,

6. Étant donnée l'urgence de la situation dans l'exemple présenté, il est irréaliste que les agents ne fassent rien si la contrainte de faisabilité collective n'est pas vérifiée ; la contrainte devrait être non-bloquante. Cependant, cette contrainte peut aider les agents à identifier les éléments manquants (e.g. un pompier ayant la compétence PSC1) de manière à réagir le plus tôt possible pour trouver une solution (e.g. appeler le poste de secours pour demander du renfort). Nous rappelons ici que bien que l'exemple proposé traite une situation d'urgence, nous ne nous limitons pas à ce type de situation dans nos travaux. Si l'on imagine une activité de chargement de matière dangereuse dans un camion, la contrainte de faisabilité collective serait bloquante, car la situation est dangereuse et non-urgente. Nous cherchons donc ici à proposer des mécanismes génériques qui permettront aux agents de raisonner sur la description de l'activité.

à un instant t .

Dans les deux cas, ces contraintes/informations seront propagées des tâches filles aux tâches mères de l'arbre de tâches, depuis les tâches feuilles jusqu'à la tâche racine.

Nous pouvons d'ores et déjà remarquer que la propagation des contraintes/informations d'un ensemble de sous-tâches à la tâche mère dépend des contraintes temporelles et logiques associées à la tâche mère. Par exemple, la tâche *faire la première phase de soins* est associée à une contrainte ET/PAR-SIM et ses trois sous-tâches doivent donc être réalisées de manière synchronisée, il faut alors un minimum de trois agents. Si la contrainte avait été par exemple ET/IND, un seul agent aurait pu réaliser les trois sous-tâches les unes à la suite des autres. Il aurait alors fallu un minimum d'un seul agent pour réaliser la tâche *faire la première phase de soins*.

Nous exposons dans la section 5.4 les mécanismes de génération et de propagation de contraintes de faisabilité collective, ces contraintes permettant de répondre aux questions 1 et 2. Nous exposons ensuite dans la section 5.5 les mécanismes de génération des informations dynamiques qui permettent de répondre aux questions 3 et 4.

5.4 Propagation des contraintes de faisabilité collective

Nous détaillons ici les mécanismes de génération de contraintes de faisabilité collective. Comme expliqué dans la section précédente, ces contraintes permettent aux agents de déterminer si l'équipe peut accomplir l'activité collective. Nous traiterons dans un premier temps les contraintes de faisabilité collective en termes de nombre de personnes, puis nous nous intéresserons aux contraintes de faisabilité collective en termes de compétences.

5.4.1 Contrainte de faisabilité collective en termes de nombre de personnes

La contrainte de faisabilité collective en termes de nombre de personnes permettra aux agents de répondre à la question 1, *L'équipe a-t-elle les ressources nécessaires en termes de nombre de personnes pour faire la tâche ?*

Pour exprimer la contrainte en termes de nombre minimal d'agents attachée à τ_k dont les sous-tâches sont celles de l'ensemble $T(\tau_k) = \{\tau_{k1}, \tau_{k2}, \dots, \tau_{kn}\}$, on note $n_{min}(\tau_k)$ le nombre minimal de personnes pouvant faire la tâche τ_k . Les contraintes sont propagées différemment selon les contraintes temporelles et logiques associées aux tâches abstraites. Nous développons dans les paragraphes suivants ces différents cas.

Cas OU/SEQ-ORD et OU/IND. La formule 5.1 donne le nombre de personnes minimal pour réaliser la tâche τ_k lorsqu'une contrainte OU/SEQ-ORD est associée à τ_k ou lorsqu'une contrainte OU/IND est associée à τ_k .

$$n_{min}(\tau_k) = \min_{i \in [1;n]} (n_{min}(\tau_{ki})) \quad (5.1)$$

Dans ces deux cas, une seule des sous-tâches de τ_k devra être réalisée pour valider τ_k . Le nombre minimum de personnes nécessaires pour faire τ_k correspond donc au plus petit des nombres minimaux de personnes requis pour faire les sous-tâches de τ_k .

Cas ET/PAR-SIM. La formule 5.2 donne le nombre de personnes minimal pour réaliser la tâche τ_k lorsqu'une contrainte ET/PAR-SIM est associée à τ_k .

$$n_{min}(\tau_k) = \sum_{i=1}^n n_{min}(\tau_{ki}) \quad (5.2)$$

Dans ce cas, toutes les sous-tâches de τ_k doivent être faites simultanément. Le nombre minimal de personnes nécessaires pour faire τ_k correspond donc à la somme des nombres minimaux de personnes nécessaires pour faire chacune des sous-tâches de τ_k .

Cas ET/SEQ-ORD et ET/IND. La formule 5.3 donne le nombre de personnes minimal pour réaliser la tâche τ_k lorsqu'une contrainte ET/SEQ-ORD est associée à τ_k ou lorsqu'une contrainte ET/IND est associée à τ_k .

$$n_{min}(\tau_k) = \max_{i \in [1;n]} (n_{min}(\tau_{ki})) \quad (5.3)$$

Dans le cas où une contrainte ET/SEQ-ORD est attachée à τ_k , les sous-tâches de τ_k *doivent* être réalisées séquentiellement. Dans le cas où une contrainte ET/IND est attachée à τ_k , les sous-tâches de τ_k *peuvent* être réalisées séquentiellement. Dans les deux cas, pour réaliser τ_k le nombre minimal de personnes est donc le plus grand des nombres minimaux de personnes nécessaires pour réaliser les sous-tâches de τ_k .

5.4.2 Contrainte de faisabilité collective en termes de compétences

La contrainte de faisabilité collective en termes de compétences doit permettre d'exprimer le fait qu'au moins deux agents doivent avoir la compétence PSC1 pour *sauver la victime*. La propagation des contraintes en termes de compétence est donc complexe, puisqu'il faut associer chaque compétence à un nombre d'agents.

Contrainte de faisabilité collective en termes de compétences sur les tâches feuilles : compétence associée à un nombre de personnes

Si une tâche feuille comporte à la fois une contrainte de nombre de personnes et de compétences, ces informations peuvent être combinées : pour la compétence attachée à τ_k , il faut au moins $n_{min}(\tau_k)$ personnes ayant la compétence κ . Si plus de personnes que $n_{min}(\tau_k)$ souhaitent participer à τ_k , *toutes* doivent avoir la compétence κ . Cependant, il sera possible de réaliser τ_k à partir du moment où $n_{min}(\tau_k)$ personnes ont la compétence κ . Cette information est automatiquement déduite à partir des éléments de nombre de personnes ($n_{min}(\tau_k) \cdot n_{max}(\tau_k)$) et de compétences renseignés par les experts du domaine.

Formalisation. On utilise la notation $n_i(\tau_k) \cdot \kappa_i$ pour exprimer le fait que $n_i(\tau_k)$ agents doivent avoir la compétence κ_i pour réaliser la tâche τ_k .

Si l'on désigne par $c_{col}(\tau_k)$ la *contrainte de faisabilité collective en termes de compétences* associée à τ_k , τ_k étant une tâche **feuille** nécessitant la compétence κ_i , alors $c_{col}(\tau_k)$ est donnée par la formule 5.4 :

$$c_{col}(\tau_k) = n_i(\tau_k) \cdot \kappa_i = n_{min}(\tau_k) \cdot \kappa_i \quad (5.4)$$

Pour exprimer le fait qu'une tâche feuille τ_k ne nécessite pas de compétences particulières, on utilisera l'expression 5.5 :

$$c_{col}(\tau_k) = TRUE. \quad (5.5)$$

Il est en effet nécessaire d'expliciter cette information puisqu'elle devra être transmise aux tâches mères (voir 5.4). Par ailleurs, les éléments *TRUE* ne sont jamais associés à un nombre de personnes.

Exemple. En reprenant l'exemple précédent, nous expliquons la génération des contraintes $c_{col}(\tau_k)$ sur les tâches feuilles :

- τ_{111} nécessite au minimum une personne et aucune compétence : $c_{col}(\tau_{111}) = TRUE$.
- τ_{1121} = *éteindre le feu* nécessite la compétence IF1 et $n_{min}(\tau_{1121}) = 1$. On a donc $c_{col}(\tau_{1121}) = 1 \cdot IF1$.
- τ_{1122} = *couper le compteur* nécessite exactement une personne qui doit avoir la compétence GOC1. On a donc $c_{col}(\tau_{1122}) = 1 \cdot GOC1$.
- τ_{12} = *faire le dégagement d'urgence* nécessite exactement deux personnes ayant la compétence PSC1. On a donc $c_{col}(\tau_{12}) = 2 \cdot PSC1$.
- Pour τ_{131} = *passer un message*, il faut également avoir la compétence PSC1 et une seule personne suffit pour faire cette tâche. On a donc $c_{col}(\tau_{131}) = 1 \cdot PSC1$.
- τ_{132} = *maintenir la tête* nécessite une personne et aucune compétence : $c_{col}(\tau_{132}) = TRUE$.
- Pour τ_{133} = *s'occuper de la victime* (τ_{133}), il faut la compétence PSC1 et exactement une seule personne peut faire cette tâche. On a donc $c_{col}(\tau_{133}) = 1 \cdot PSC1$.

Ces éléments sont représentés dans la figure 5.3. La contrainte en termes de compétence $c_{col}(\tau_k)$ générée automatiquement apparaît pour chaque tâche feuille τ_k dans les bulles orange en haut à droite de chaque tâche feuille. Cette figure peut également être trouvée en pleine page en annexe B.3.

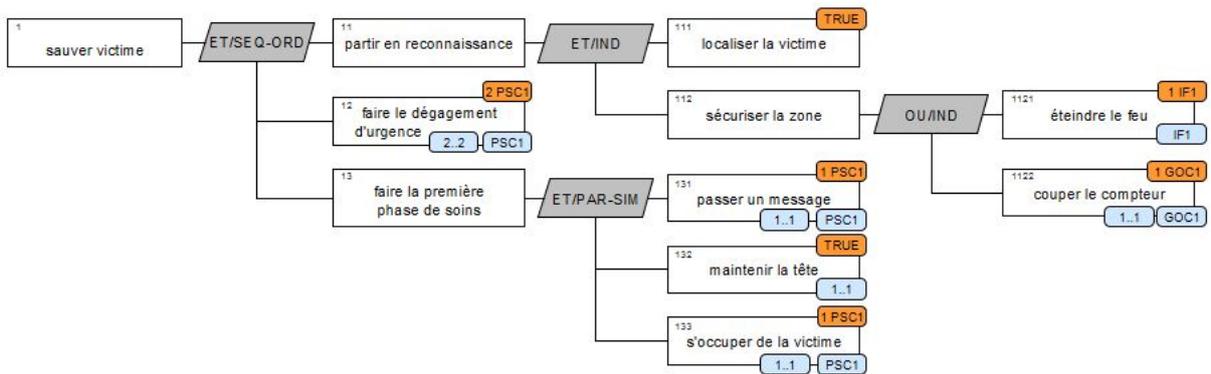


FIGURE 5.3 – Même arbre de tâches que celui présenté en figure 5.2 comportant (i) dans les bulles bleues en bas à droite de chaque tâche feuille, les éléments de nombre de personnes et de compétences renseignés par les experts du domaine ; et (ii) dans les bulles orange en haut à droite de chaque tâche feuille, les contraintes de faisabilité collective en termes de compétences associées aux tâches feuilles.

Forme générale des contraintes de faisabilité collective en termes de compétences

Soit $c_{col}(\tau_k)$ la contrainte de faisabilité collective en termes de compétences attachée à τ_k , dont les sous-tâches sont celles de l'ensemble $T(\tau_k) = \{\tau_{k1}, \tau_{k2}, \dots, \tau_{kn}\}$. Cette contrainte doit permettre de répondre à la question : *Quelles compétences faut-il pour faire la tâche τ_k et combien de personnes doivent avoir chaque compétence pour faire la tâche τ_k ?* Autrement dit, $c_{col}(\tau_k)$ doit permettre d'exprimer l'information suivante :

Exemple 1. Pour sauver la victime, il faut (*option1*) deux personnes ayant la compétence PSC1 et une personne ayant la compétence GOC1 OU (*option2*) deux personnes ayant la compétence PSC1 et une personne ayant la compétence IF1.

En effet, ACTIVITY-DL permet d'exprimer des alternatives pour la réalisation de certaines tâches. Par exemple, pour réaliser $\tau_{112} = \text{sécuriser la zone}$, il faut ou bien faire $\tau_{1121} = \text{éteindre le feu}$, ou bien faire $\tau_{1122} = \text{couper le compteur}$. Ces alternatives doivent être reflétées dans la génération de la contrainte $c_{col}(\tau_{112})$.

Formalisation. Étant donné que chaque tâche feuille peut nécessiter aucune ou une seule compétence, on pourra toujours exprimer $c_{col}(\tau_k)$ par une forme normale disjonctive (FND), comme dans l'*exemple 1*, et comme exprimé dans la formule 5.6 la contrainte $c_{col}(\tau_k)$. Dans cette formule, les $C_j(\tau_k)$ représentent des conjonctions de compétences associées à la tâche τ_k ($n(k)$ est le nombre de conjonctions de compétences associées à τ_k) :

$$c_{col}(\tau_k) = C_1(\tau_k) \vee \dots \vee C_{n(k)}(\tau_k) = \bigvee_{j=1}^{n(k)} C_j(\tau_k). \quad (5.6)$$

On peut alors écrire chaque conjonction $C_j(\tau_k)$ avec la formule 5.7, où $n_i(C_j(\tau_k))$ représente le nombre de personnes nécessaires devant avoir la compétence κ_i pour faire τ_k selon la conjonction $C_j(\tau_k)$. Dans cette formule, $m(j)$ représente le nombre de compétences différentes présentes dans $C_j(\tau_k)$.

$$C_j(\tau_k) = n_1(C_j(\tau_k)) \cdot \kappa_1 \wedge \dots \wedge n_{m(j)}(C_j(\tau_k)) \cdot \kappa_{m(j)} \quad (5.7)$$

À partir du moment où l'une des conjonctions de compétences $C_j(\tau_k)$ de $c_{col}(\tau_k)$ est vérifiée, alors $c_{col}(\tau_k)$ sera vérifiée et la tâche τ_k sera réalisable.

On rappelle que le tableau C.1 en annexe récapitule les notations utilisées et leur signification.

Exemple. Si l'on reprend l'*exemple 1* page 85, on a, suivant la formule 5.6 : $c_{col}(\tau_1) = C_1(\tau_1) \vee C_2(\tau_1)$, avec $C_1(\tau_1) = 2 \cdot \text{PSC1} \wedge 1 \cdot \text{GOC1}$ et $C_2(\tau_1) = 2 \cdot \text{PSC1} \wedge 1 \cdot \text{IF1}$ (formule 5.7). Ces deux conjonctions correspondent respectivement à *option1* et *option2*.

Propagation depuis les tâches filles

Pour une tâche composée τ_k dont les sous-tâches sont celles de l'ensemble $T(\tau_k) = \{\tau_{k1}, \tau_{k2}, \dots, \tau_{kn}\}$, la contrainte $c_{col}(\tau_k)$ va être générée à partir des contraintes précisées sur chacune de ses tâches filles $c_{col}(\tau_{k1}), c_{col}(\tau_{k2}), \dots, c_{col}(\tau_{kn})$. Deux grands cas sont alors à distinguer :

- (i) La contrainte logique associée à τ_k est ET : toutes les tâches filles devront être réalisées, donc toutes les contraintes $c_{col}(\tau_{ki})$ ont besoin d'être vérifiées.

- (ii) La contrainte logique associée à τ_k est OU : une seule des tâches filles devra être réalisée, donc une seule des contraintes $c_{col}(\tau_{ki})$ a besoin d'être vérifiée.

Cependant, on ne peut pas simplement exprimer $c_{col}(\tau_k)$ comme (i) la conjonction des $c_{col}(\tau_{ki})$ ou comme (ii) la disjonction des $c_{col}(\tau_{ki})$. En effet, certaines sous-tâches de τ_k peuvent nécessiter des compétences identiques. Dans notre exemple, c'est le cas pour la tâche $\tau_{13} = \text{faire la première phase de soins}$, dont les sous-tâches $\tau_{131} = \text{passer un message}$ et $\tau_{133} = \text{s'occuper de la victime}$ nécessitent toutes les deux la compétence PSC1. Au niveau de τ_{13} , il faut donc combiner ces informations et calculer combien d'agents doivent avoir la compétence PSC1. Pour cela, il faut prendre en compte la contrainte temporelle associée à τ_{13} .

Nous avons donc cinq cas différents à traiter pour propager la contrainte de faisabilité collective en termes de compétences depuis ses tâches filles à une tâche composée. Nous développons ces cinq cas dans les paragraphes suivants.

Cas OU/SEQ-ORD et OU/IND. De même que pour la génération de la contrainte de faisabilité collective en termes de nombre de personnes, nous traitons de la même manière les cas OU/SEQ-ORD et OU/IND pour la génération de la contrainte de faisabilité collective en termes de compétences.

Expression de $c_{col}(\tau_k)$. Dans les cas OU/SEQ-ORD et OU/IND, $c_{col}(\tau_k)$ est donnée par la formule 5.8 :

$$c_{col}(\tau_k) = \bigvee_{i=1}^n c_{col}(\tau_{ki}) = \bigvee_{i=1}^n \left(\bigvee_{j=1}^{n(ki)} C_j(\tau_{ki}) \right) = \bigvee_{j=1}^{\sum_{i=1}^n n(ki)} C_j(\tau_k). \quad (5.8)$$

La disjonction de plusieurs FND donne une FND. Cependant sous cette forme, $c_{col}(\tau_k)$ peut contenir plusieurs fois la même conjonction de compétences, au nombre de personnes nécessaires près. Autrement dit, on peut avoir $C_{j_1}(\tau_k)$ et $C_{j_2}(\tau_k)$ telles que définies dans la formule 5.9 :

$$\begin{aligned} C_{j_1}(\tau_k) &= n_1(C_{j_1}(\tau_k)) \cdot \kappa_1 \wedge \dots \wedge n_x(C_{j_1}(\tau_k)) \cdot \kappa_x \\ C_{j_2}(\tau_k) &= n_1(C_{j_2}(\tau_k)) \cdot \kappa_1 \wedge \dots \wedge n_x(C_{j_2}(\tau_k)) \cdot \kappa_x. \end{aligned} \quad (5.9)$$

Simplification par le minimum. Dans ce cas, on peut simplifier $c_{col}(\tau_k)$ si et seulement si, parmi les $C_j(\tau_k)$ qui sont identiques au nombre de personnes près, il existe une conjonction $C_{min}(\tau_k)$ qui vérifie :

$$\forall \ell \in [0; x] \quad min = \arg \min_j \{n_\ell(C_j(\tau_k))\}. \quad (5.10)$$

Si une telle conjonction existe, alors toutes les conjonctions qui sont identiques au nombre de personnes près sont éliminées.

La formule de simplification par le minimum 5.10 traduit le fait que dire "il faut au minimum deux X OU il faut au minimum trois X" revient à dire "il faut au minimum deux X". La formule exprime également le fait que la proposition "il faut au minimum deux X et trois Y OU il faut au minimum trois X et deux Y" ne peut pas être simplifiée.

Après cette étape de simplification, on obtient :

$$c_{col}(\tau_k) = C_1(\tau_k) \vee C_2(\tau_k) \vee \dots \vee C_{n(k)}(\tau_k)$$

comme dans la formule 5.6 où $n(k) \leq \sum_{i=1}^n n(i)$.

Exemple pour les cas OU/SEQ-ORD et OU/IND. Soit la tâche τ_{14} = mettre la victime sur le plan dur, composée avec la contrainte OU/IND des sous-tâches (actions – tâches feuilles) :

- τ_{141} = basculer la victime pour la mettre sur le plan dur. Pour basculer la victime, il faut être au moins deux et il faut avoir la compétence PSC1. On a donc $n_{min}(\tau_{141}) = 2$ et $c(\tau_{141}) = 2 \cdot \text{PSC1}$.
- τ_{142} = porter la victime pour la mettre sur le plan dur. Pour porter la victime, il faut être au moins trois et il faut également avoir la compétence PSC1. On a donc $n_{min}(\tau_{142}) = 3$ et $c(\tau_{142}) = 3 \cdot \text{PSC1}$.

D'après la formule 5.8, on a :

$$c_{col}(\tau_{14}) = c_{col}(\tau_{141}) \vee c_{col}(\tau_{142})$$

soit

$$c_{col}(\tau_{14}) = 2 \cdot \text{PSC1} \vee 3 \cdot \text{PSC1}$$

où $2 \cdot \text{PSC1}$ et $3 \cdot \text{PSC1}$ correspondent respectivement aux conjonctions $C_1(\tau_{14})$ et $C_2(\tau_{14})$ de la formule 5.8.

$C_1(\tau_{14})$ et $C_2(\tau_{14})$ étant identiques au nombre de personnes près comme défini dans la formule 5.9 et $C_1(\tau_{14})$ minorant $C_2(\tau_{14})$ d'après la formule 5.10, on peut simplifier $c_{col}(\tau_{14})$:

$$c_{col}(\tau_{14}) = 2 \cdot \text{PSC1} .$$

De la même manière, dans l'exemple que nous avons utilisé jusqu'ici, la tâche τ_{112} = sécuriser la zone est composée avec une contrainte OU/IND des sous-tâches τ_{1121} = éteindre le feu qui nécessite la compétence IF1 et τ_{1122} = éteindre le feu qui nécessite la compétence GOC1. On peut exprimer $c_{col}(\tau_{112})$:

$$c_{col}(\tau_{112}) = c_{col}(\tau_{1121}) \vee c_{col}(\tau_{1122})$$

soit

$$c_{col}(\tau_{112}) = 1 \cdot \text{IF1} \vee 1 \cdot \text{GOC1} .$$

Ces deux conjonctions n'étant pas identiques, on ne modifie pas $c_{col}(\tau_{112})$.

Cas ET/PAR-SIM. On utilise la formule 5.11 pour désigner la contrainte de faisabilité collective en termes de compétences $c_{col}(\tau_k)$ associée à la tâche composée τ_k lorsque la contrainte temporelle sur τ_k est PAR-SIM et la contrainte logique est ET :

$$c_{col}(\tau_k) = \bigwedge_{i=1}^n c_{col}(\tau_{ki}) = \bigwedge_{i=1}^n \left(\bigvee_{j=1}^{n(ki)} C_j(\tau_{ki}) \right) . \quad (5.11)$$

Il faut transformer $c_{col}(\tau_k)$ pour qu'elle corresponde à une FND comme dans la formule 5.6.

Distributivité. Pour transformer $c_{col}(\tau_k)$ de manière à ce qu'elle corresponde à une FND comme dans la formule 5.6, on applique dans un premier temps la règle de distributivité de l'opérateur \wedge sur les conjonctions $C_j(\tau_{ki})$.

Dans un second temps, il faut simplifier chaque conjonction de la FND obtenue.

Si une conjonction $C_j(\tau_k)$ contient des éléments de la forme $n_i(\tau_{kj}) \cdot \kappa_i$ **et** un ou plusieurs éléments *TRUE*, les éléments *TRUE* peuvent être éliminés. En effet, ces éléments n'apportent

ici pas plus d'informations. Dans le cas où la conjonction est composée d'un seul élément *TRUE*, celui-ci ne peut évidemment pas être éliminé. Une telle conjonction exprime le fait que τ_k serait réalisable par des agents n'ayant aucune compétence.

De plus, chacune des conjonctions $C_j(\tau_k)$ peut contenir plusieurs fois la même compétence κ_ℓ , associée à différents nombres de personnes, comme dans la formule 5.12 :

$$C_j(\tau_k) = n_1(\tau_{ka_1}) \cdot \kappa_1 \wedge \dots \wedge n_1(\tau_{ka_{n_1}}) \cdot \kappa_1 \\ \wedge n_x(\tau_{ka_1}) \cdot \kappa_x \wedge \dots \wedge n_x(\tau_{ka_{n_x}}) \cdot \kappa_x . \quad (5.12)$$

Pour chaque conjonction, il faut donc calculer le bon nombre de personnes devant avoir chaque compétence.

Calcul de $n_\ell(C_j(\tau_k))$. Il faut maintenant déterminer $n_\ell(C_j(\tau_k))$ le nombre d'agents nécessaires pour chaque compétence κ_ℓ de chaque conjonction de compétences $C_j(\tau_k)$.

Comme nous l'avons déjà remarqué, dans notre exemple, τ_{13} a deux sous-tâches τ_{131} et τ_{133} qui nécessitent la compétence *PSC1*. Or ces sous-tâches doivent être réalisées simultanément et nécessitent chacune au minimum un intervenant : il faut donc deux agents ayant la compétence *PSC1* pour réaliser la tâche τ_{113} .

Pour toute compétence κ_ℓ de $C_j(\tau_k)$, le nombre d'agents $n_\ell(C_j(\tau_k))$ est donné par la formule 5.13 :

$$n_\ell(C_j(\tau_k)) = \sum_{\alpha=a_1}^{a_{n_\ell}} n_\ell(\tau_{k\alpha}) . \quad (5.13)$$

Dans notre exemple, on a $\kappa_\ell = \text{PSC1}$ nécessaire à τ_{131} et à τ_{133} , avec $n_\ell(\tau_{131}) = 1$ et $n_\ell(\tau_{133}) = 1$, ce qui donne bien $n_\ell(\tau_{13}) = 2$.

Simplification par le minimum. Enfin, tout comme pour les cas *OU/SEQ-ORD* et *OU/IND*, $c_{col}(\tau_k)$ peut être simplifiée en utilisant la même règle de simplification par le minimum que décrite précédemment.

Exemple pour le cas *ET/PAR-SIM*. Prenons l'exemple de la tâche τ_{13} . On cherche à exprimer $c_{col}(\tau_{13})$. D'après la formule 5.11, on a

$$c_{col}(\tau_{13}) = c_{col}(\tau_{131}) \wedge c_{col}(\tau_{132}) \wedge c_{col}(\tau_{133})$$

soit

$$c_{col}(\tau_{13}) = C_1(\tau_{13}) = 1 \cdot \text{PSC1} \wedge \text{TRUE} \wedge 1 \cdot \text{PSC1}$$

et dont l'élément *TRUE* peut être éliminé :

$$c_{col}(\tau_{13}) = C_1(\tau_{13}) = 1 \cdot \text{PSC1} \wedge 1 \cdot \text{PSC1} .$$

La compétence *PSC1* étant présente plusieurs fois comme dans la formule 5.12, on calcule $n_{\text{PSC1}}(C_1(\tau_{13})) = 1 + 1 = 2$ avec la formule 5.13, ce qui permet d'obtenir :

$$c_{col}(\tau_{13}) = 2 \cdot \text{PSC1} .$$

Nous avons traité ici un exemple simple pour illustrer le mécanisme de génération de $c_{col}(\tau_k)$ dans le cas où τ_k est une tâche composée à laquelle une contrainte *ET/PAR-SIM* est associée, mais le mécanisme défini permet de traiter des combinaisons complexes de compétences.

Cas ET/SEQ-ORD et ET/IND. Tout comme que pour la génération de la contrainte de faisabilité collective en termes de nombre de personnes, nous traitons de la même manière les cas ET/SEQ-ORD et ET/IND pour la génération de la contrainte de faisabilité collective en termes de compétences.

Dans les cas ET/SEQ-ORD et ET/IND, $c_{col}(\tau_k)$ est exprimée de la même manière que pour le cas précédant dans la formule 5.11 :

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

et la même règle de distributivité de l'opérateur \vee doit être appliquée pour obtenir la même forme que dans la formule 5.6 :

$$c_{col}(\tau_k) = \bigvee_{j=1}^{n(k)} (C_j(\tau_k)) .$$

De même que pour le cas ET/PAR-SIM, chaque conjonction de compétences $C_j(\tau_k)$ peut être simplifiée en enlevant les éléments *TRUE*, et le nombre $n_\ell(C_j(\tau_k))$ de personnes nécessaires pour chaque compétence κ_ℓ de la conjonction doit être calculé. Ce nombre est ici calculé avec la formule 5.14 :

$$n_\ell(C_j(\tau_k)) = \max_{\alpha \in [1; a_{n_\ell}]} (n_\ell(\tau_{k\alpha})) . \quad (5.14)$$

Une fois ce nombre calculé pour chaque compétence de chaque conjonction, on peut appliquer à nouveau la règle de simplification par le minimum, et on trouve $c_{col}(\tau_k)$ sous la forme 5.6.

Exemple pour les cas ET/SEQ-ORD et ET/IND. Nous considérons tout d'abord la tâche $\tau_{11} = \textit{partir en reconnaissance}$, composée par une contrainte ET/IND des sous-tâches $\tau_{111} = \textit{localiser la victime}$ et $\tau_{112} = \textit{sécuriser la zone}$.

D'après la formule 5.11, on a :

$$c_{col}(\tau_{11}) = c_{col}(\tau_{111}) \wedge c_{col}(\tau_{112})$$

soit

$$c_{col}(\tau_{11}) = \textit{TRUE} \wedge (1 \cdot \textit{IF1} \vee 1 \cdot \textit{GOC1}) .$$

On applique ici la règle de distributivité de l'opérateur \vee , et on obtient c_{col} telle que définie dans la formule 5.6 :

$$c_{col}(\tau_{11}) = (\textit{TRUE} \wedge 1 \cdot \textit{IF1}) \vee (\textit{TRUE} \wedge 1 \cdot \textit{GOC1}) .$$

On peut ensuite éliminer les éléments *TRUE* qui sont inutiles ici :

$$c_{col}(\tau_{11}) = 1 \cdot \textit{IF1} \vee 1 \cdot \textit{GOC1} .$$

Aucune des conjonctions de $c_{col}(\tau_{11})$ ne comportant plus d'une fois la même compétence, on ne modifie pas $c_{col}(\tau_{11})$.

Considérons maintenant la tâche $\tau_1 = \textit{sauver la victime}$, composée par une contrainte ET/SEQ-ORD des sous-tâches $\tau_{11} = \textit{partir en reconnaissance}$, $\tau_{12} = \textit{faire le dégagement d'urgence}$ et $\tau_{13} = \textit{faire la première phase de soins}$.

D'après la formule 5.11, on a :

$$c_{col}(\tau_1) = c_{col}(\tau_{11}) \wedge c_{col}(\tau_{12}) \wedge c_{col}(\tau_{13})$$

soit

$$c_{col}(\tau_1) = (1 \cdot \text{IF1} \vee 1 \cdot \text{GOC1}) \wedge (2 \cdot \text{PSC1}) \wedge (2 \cdot \text{PSC1}) .$$

On applique la règle de distributivité de l'opérateur \vee , on obtient donc :

$$c_{col}(\tau_1) = (1 \cdot \text{IF1} \wedge 2 \cdot \text{PSC1} \wedge 2 \cdot \text{PSC1}) \vee (1 \cdot \text{GOC1} \wedge 2 \cdot \text{PSC1} \wedge 2 \cdot \text{PSC1})$$

comme dans la formule 5.6. Les deux conjonctions comportent toutes les deux des compétences identiques ; on utilise donc la formule 5.14 pour calculer dans chacune le nombre de personnes devant avoir la compétence PSC1, et on obtient :

$$c_{col}(\tau_1) = (1 \cdot \text{IF1} \wedge 2 \cdot \text{PSC1}) \vee (1 \cdot \text{GOC1} \wedge 2 \cdot \text{PSC1}) .$$

5.4.3 Exemple de propagation de contraintes de faisabilité collective sur un arbre de tâches ACTIVITY-DL

Nous présentons dans la figure 5.4 les contraintes de faisabilité collective en termes de (i) nombre minimal de personnes et (ii) compétences nécessaires attachées aux tâches composées. Ces contraintes sont présentées dans les bulles orange en haut à droite de chaque tâche composée.

Cette figure est également présentée en pleine page en annexe ; voir B.4.

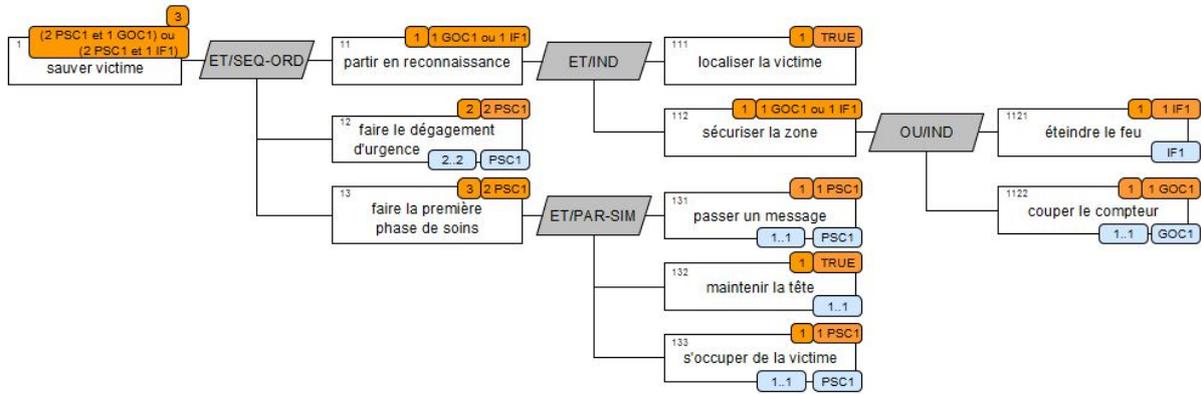


FIGURE 5.4 – Même arbre de tâches que celui présenté en figure 5.2 sur lequel apparaissent, dans les bulles orange en haut à droite de chaque tâche, les contraintes de faisabilité collective en termes de nombre minimal de personnes et de compétences nécessaires.

5.4.4 Conclusion

Nous avons présenté dans cette section notre proposition de propagation de contraintes de faisabilité collective des tâches filles à leur tâche mère, depuis les tâches feuilles jusqu'à la racine de l'arbre de tâches. Cette proposition répond au problème suivant : les arbres de tâches peuvent être de taille conséquente, et pour éviter de prendre en compte plusieurs centaines de tâches à

la fois, les agents raisonnent en priorité sur les tâches abstraites. Or certaines informations contraignent la réalisation de l'activité par les agents et ne sont spécifiées que sur les tâches feuilles de l'arbre de tâches. Ces informations sont liées aux aspects collaboratifs : d'une part pour permettre la collaboration au niveau du scénario et pour faire le lien avec le modèle de confiance de Mayer et al. (1995), des compétences peuvent être requises pour certaines actions (i.e. tâches feuilles) ; d'autre part pour permettre la collaboration au niveau des actions, des nombres minimaux et maximaux de personnes pouvant participer sont spécifiés également au niveau des tâches feuilles.

Ces informations étant déterminantes pour les choix que vont faire les agents en termes d'action à réaliser, nous proposons de générer des contraintes qui sont propagées à l'ensemble de l'arbre de tâches. De cette manière, les agents peuvent tout de suite savoir s'ils pourront réaliser l'activité collective, et cela évitera les situations d'impasse.

Nous avons précisé plus tôt dans ce chapitre que par souci de simplicité nous ne développons pas toutes les combinaisons de contraintes de faisabilité et de contraintes de compétences qu'il est possible de rencontrer dans ACTIVITY-DL (voir 5.1.3). Toutefois, à l'exception des combinaisons comportant la contrainte temporelle PAR, toutes les combinaisons de contraintes logiques et temporelles qu'il est possible de former dans ACTIVITY-DL peuvent être traitées avec les mécanismes proposés ici (voir l'annexe D pour plus de détails).

5.5 Propagation dynamique d'informations

Les contraintes de faisabilité collective permettent aux agents de savoir, a priori, si l'équipe pourra réaliser l'activité collective. Préciser ces contraintes au niveau des tâches abstraites permet d'éviter aux agents de raisonner sur l'ensemble de l'arbre de tâches.

En reprenant l'exemple de l'activité collective *sauver la victime* développé dans la section précédente, imaginons le scénario suivant : trois agents, Anna, Béa et Cara assistent à l'effondrement de la maison. Par chance, Anna et Béa sont pompiers et ont suivi les formations PSC1, IF1 et GOC1. Ces trois agents vérifient donc la contrainte de faisabilité collective et pourront sauver la victime.

L'agent Cara, qui n'a aucune compétence particulière, pourra aider à localiser la victime, mais il faudra ensuite attendre que la zone soit sécurisée et que le dégagement d'urgence soit fait pour qu'elle puisse à nouveau participer, en maintenant la tête de la victime.

Par ailleurs, si l'agent Dora arrive pendant la réalisation de la première phase de soins, elle ne pourra pas participer au sauvetage de la victime : en effet, chacune des trois sous-tâches de *faire la première phase de soins* accepte au maximum un intervenant.

De la même manière que pour les contraintes de faisabilité collective, propager ces informations sur les tâches composées de l'arbre de tâches peut faciliter le raisonnement des agents.

À la différence des contraintes de faisabilité collective, ces informations ne sont pas déterminantes pour le succès ou l'échec de la tâche collective ; elles sont simplement pertinentes pour qu'un agent, à un instant t , puisse savoir s'il peut ou non participer à l'activité collective. Ces informations doivent donc être représentatives de l'avancement des agents dans l'activité collective ; elles doivent être modifiées *dynamiquement*, à chaque fois qu'une tâche de l'arbre de tâches est réalisée.

Ces informations servent aux agents, à partir du moment où (i) ils pensent que la tâche τ_k est réalisable (i.e. la contrainte de faisabilité collective est vérifiée) et (ii) ils envisagent de faire τ_k , à savoir s'ils peuvent participer à τ_k .

5.5.1 Information sur le nombre maximal de participants

Pour que Dora sache qu'elle ne peut pas participer à *sauver la victime* si Anna, Béa et Cara font la première phase de soins, on doit propager, pour chaque tâche composée τ_k , un nombre de participants maximal. On désigne par $n_{max}(\tau_k, t)$ le nombre maximal de personnes pouvant participer à τ_k à l'instant t .

Dans le cas où τ_k est une tâche feuille, on a $n_{max}(\tau_k, t) = n_{max}(\tau_k)$.

Comme pour les contraintes de faisabilité collective, le calcul de $n_{max}(\tau_k, t)$ n'est pas simple et il dépend des contraintes temporelles associées à τ_k . Nous développons dans les paragraphes suivants les règles de calcul pour chaque cas.

Cas ET/SEQ-ORD et OU/SEQ-ORD. Dans ces deux cas, les sous-tâches de τ_k sont réalisées séquentiellement dans l'ordre défini. Si τ_{kk} est la première sous-tâche de τ_k que l'on n'a pas encore essayé de réaliser, alors $n_{max}(\tau_k, t)$ est donné par la formule 5.15 :

$$n_{max}(\tau_k, t) = n_{max}(\tau_{kk}, t). \quad (5.15)$$

Cas ET/PAR-SIM, ET/IND et OU/IND. Lorsque la contrainte attachée à τ_k est ET/PAR-SIM, toutes les sous-tâches de τ_k doivent être réalisées simultanément.

Dans le cas où une contrainte ET/IND est attachée à τ_k , toutes les sous-tâches doivent être réalisées, et on peut toutes les réaliser en même temps.

Enfin si une contrainte OU/IND est attachée à τ_k , une seule des sous-tâches de τ_k doit être réalisée, mais on peut essayer de faire toutes les sous-tâches de τ_k en même temps. De plus, la contrainte logique attachée à τ_k étant un OU et non un OU-exclusif, on peut toujours réaliser d'autres sous-tâches de τ_k même après que τ_k ait été validée.

Le nombre maximal de personnes pouvant participer à τ_k à l'instant t ($n_{max}(\tau_k, t)$) est donc calculé de la même façon dans ces trois cas. Si $T_{NOTDONE}(\tau_k, t)$ est le sous-ensemble des sous-tâches de τ_k qui n'ont pas encore été réalisées, alors $n_{max}(\tau_k, t)$ est donné par la formule 5.16 :

$$n_{max}(\tau_k, t) = \sum_{\tau_{ki} \in T_{NOTDONE}(\tau_k, t)} n_{max}(\tau_{ki}, t). \quad (5.16)$$

Note : dans le cas ET/PAR-SIM, les sous-tâches doivent être réalisées simultanément ; aucune des sous-tâches ne sera terminée avant les autres. On a donc $T_{NOTDONE}(\tau_k, t) = T(\tau_k)$, et cette information ne sera mise à jour que lorsque toutes les sous-tâches seront terminées ($T_{NOTDONE}(\tau_k, t) = \emptyset$).

5.5.2 Information sur les compétences

L'information sur les compétences permettra à chaque agent de savoir si, personnellement, à l'instant t , il a les compétences nécessaires pour participer à l'activité collective. On note $c(\tau_k, t)$ l'information sur les compétences qui renseigne l'agent sur les compétences qu'il doit posséder

pour participer à τ_k à un instant t . Si τ_k est une tâche feuille et κ est la compétence attachée à τ_k , alors $c(\tau_k, t) = \kappa$. Par ailleurs, $c(\tau_k, t)$ est nécessairement une disjonction de compétences, puisque chaque tâche feuille nécessite au plus une compétence et pour les tâches composées, nous verrons que $c(\tau_k, t)$ est toujours construite à partir de disjonctions des $c(\tau_{ki}, t)$ liées aux sous-tâches τ_{ki} de τ_k .

L'information $c(\tau_k, t)$ permettra à Cara de savoir qu'une fois la victime localisée, elle ne peut plus participer à l'activité collective (tant que le dégagement d'urgence n'est pas fait).

Encore une fois, la génération de cette information pour une tâche composée se fait à partir des informations de ses tâches filles et dépend des contraintes temporelles spécifiées sur les tâches filles.

Cas ET/SEQ-ORD et OU/SEQ-ORD. Dans les cas ET/SEQ-ORD et OU/SEQ-ORD, un agent peut participer à la tâche τ_k s'il peut faire la sous-tâche τ_{kk} (i.e. la première sous-tâche de τ_k que l'on n'a pas encore réalisée). L'expression de $c(\tau_k, t)$ est donc donnée par la formule 5.17 :

$$c(\tau_k, t) = c(\tau_{kk}, t) . \quad (5.17)$$

Cas ET/PAR-SIM, ET/IND et OU/IND. Ces trois cas partagent la même règle de génération de l'information $c(\tau_k, t)$, pour les mêmes raisons que celles expliquées précédemment (voir 5.5.1).

Ici $c(\tau_k, t)$ exprime l'information que pour participer à τ_k , l'agent doit pouvoir participer à au moins l'une des sous-tâches de τ_k qui n'est pas encore réalisée, c'est-à-dire à l'une des tâches de l'ensemble $T_{\text{NOTDONE}}(\tau_k, t)$. Pour les cas ET/PAR-SIM, ET/IND et OU/IND, $c(\tau_k, t)$ est donc donnée par la formule 5.18 :

$$c(\tau_k, t) = c(\tau_{k1}, t) \vee \dots \vee c(\tau_{km}, t) \quad \text{avec } \{\tau_{k1}, \dots, \tau_{km}\} = T_{\text{NOTDONE}}(\tau_k, t) . \quad (5.18)$$

L'expression $c(\tau_k, t)$ peut être simplifiée si plusieurs disjonctions $c(\tau_{ki}, t)$ sont identiques, il ne faut en garder qu'une seule. Cependant contrairement à $c_{\text{col}}(\tau_k)$, aucun autre traitement n'a besoin d'être apporté à $c(\tau_k, t)$ car on traite ici de simples disjonctions de *compétences*, et non des disjonctions de *compétences associées à un nombre de personnes*.

Note : On ne simplifie pas ici les disjonctions qui comporteraient un élément *TRUE* en enlevant tous les éléments. En effet, nous verrons dans le chapitre 6 les agents auront besoin de savoir quelles compétences sont nécessaires pour les sous-tâches de τ_k même si l'une de ces sous-tâches ne nécessite aucune compétence (voir 6.4.1).

5.5.3 Exemple de propagation dynamique d'informations

Pour illustrer les mécanismes de propagation d'information, nous reprenons encore une fois l'exemple *sauver la victime*. Nous développons tout d'abord l'exemple à t_0 , puis nous reprenons ce même exemple à t_1 , et à t_2 pour illustrer le caractère dynamique de ces informations.

Propagation des informations à t_0

Nous exposons ici, pour chaque tâche composée τ_k , la propagation depuis ses tâches filles des informations de nombre maximal de personnes pouvant participer et de compétences nécessaires à un agent pour participer :

- τ_{112} = *sécuriser la zone* est composée avec une contrainte OU/IND de τ_{1121} = *éteindre le feu* et τ_{1122} = *couper le compteur*. On a donc :
 - $n_{max}(\tau_{112}, t_0) = n_{max}(\tau_{1121}, t_0) + n_{max}(\tau_{1122}, t_0) = 1 + \text{INF}$ soit $n_{max}(\tau_{112}, t_0) = \text{INF}$;
et
 - $c(\tau_{112}, t_0) = c(\tau_{1121}, t_0) \vee c(\tau_{1122}, t_0) = \text{IF1} \vee \text{GOC1}$.
- τ_{11} = *partir en reconnaissance* est composée avec la contrainte ET/IND des tâches τ_{111} = *localiser la victime* et τ_{112} = *sécuriser la zone*. On a donc :
 - $n_{max}(\tau_{11}, t_0) = n_{max}(\tau_{111}, t_0) + n_{max}(\tau_{112}, t_0) = \text{INF} + \text{INF}$ soit $n_{max}(\tau_{11}, t_0) = \text{INF}$;
et
 - $c(\tau_{11}, t_0) = c(\tau_{111}, t_0) \vee c(\tau_{112}, t_0) = \text{TRUE} \vee (\text{IF1} \vee \text{GOC1})$, puisque τ_{111} ne nécessite aucune compétence. On obtient donc $c(\tau_{11}, t_0) = \text{TRUE} \vee \text{IF1} \vee \text{GOC1}$: Cara, qui n'a aucune compétence, peut participer à τ_{111} (elle localisera la victime).
- τ_{13} = *faire la première phase de soins* est composée avec une contrainte ET/PAR-SIM de trois sous-tâches, qui sont τ_{131} = *passer un message*, τ_{132} = *maintenir la tête* et τ_{133} = *s'occuper de la victime*. On obtient alors :
 - $n_{max}(\tau_{13}, t_0) = n_{max}(\tau_{131}, t_0) + n_{max}(\tau_{132}, t_0) + n_{max}(\tau_{133}, t_0) = 1 + 1 + 1 = 3$. Au maximum, il ne peut y avoir que trois agents qui participent à τ_{13} ; et
 - $c(\tau_{13}, t_0) = c(\tau_{131}, t_0) \vee c(\tau_{133}, t_0) \vee c(\tau_{132}, t_0) = \text{PSC1} \vee \text{TRUE} \vee \text{PSC1}$. On peut donc simplifier, et on obtient : $c(\tau_{13}, t_0) = \text{PSC1} \vee \text{TRUE}$.
- Enfin, τ_1 = *sauver la victime* est composée des trois sous-tâches τ_{11} , τ_{12} = *faire le dégagement d'urgence* et τ_{13} ; et une contrainte ET/SEQ-ORD est attachée à τ_1 . À t_0 , la première sous-tâche de τ_1 qui n'est pas encore réalisée est τ_{11} ; on a donc :
 - $n_{max}(\tau_1, t_0) = n_{max}(\tau_{11}, t_0) = \text{INF}$; et
 - $c(\tau_1, t_0) = c(\tau_{11}, t_0) = \text{TRUE} \vee \text{IF1} \vee \text{GOC1}$.
 À t_0 , tous les agents qui le souhaitent peuvent participer à *sauver la victime*, sans restriction sur les compétences des agents. En effet, tous les agents pourront participer à la localisation de la victime.

La figure 5.5 fait apparaître ces informations sur l'arbre de tâches. Cette figure peut également être trouvée en pleine page en annexe B.5.

Propagation des informations à t_1

Si les agents localisent tout de suite la victime, à t_1 , les informations de compétences sont modifiées :

- Pour τ_{11} , $c(\tau_{11}, t_1)$ devient $c(\tau_{11}, t_1) = c(\tau_{112}, t_1)$, puisque que τ_{111} ne fait plus partie de l'ensemble $\text{TNOTDONE}(\tau_{11}, t_1)$; soit $c(\tau_{11}, t_1) = \text{IF1} \vee \text{GOC1}$. (Les informations de nombre de personnes sont également mises à jour, mais cela ne change rien ici.)
- Cette information est propagée à τ_1 , et on a donc $c(\tau_1, t_1) = \text{IF1} \vee \text{GOC1}$.

À t_1 , tous les agents qui souhaitent participer au sauvetage de la victime doivent avoir la compétence IF1 ou GOC1. Cara ne peut donc plus participer. Ces informations sont représentées dans la figure 5.6 et cette même figure est disponible en pleine page en annexe B.6.

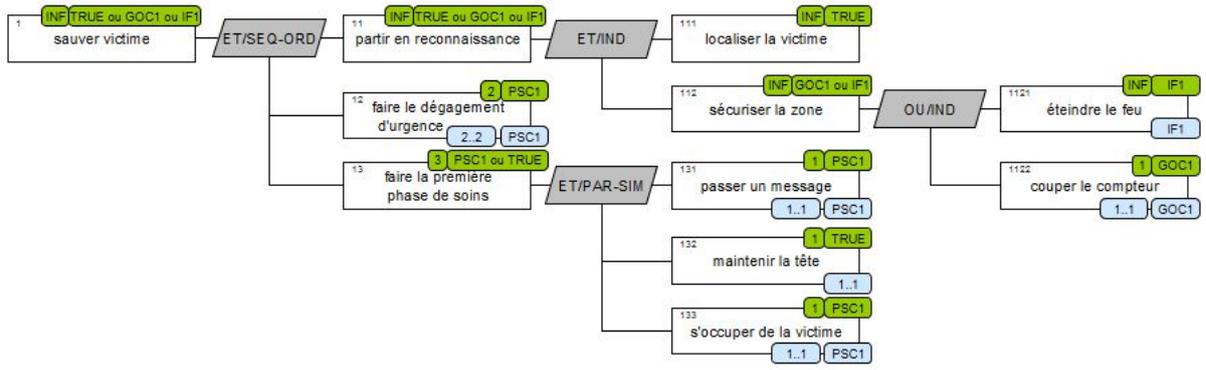


FIGURE 5.5 – Même arbre de tâches que celui présenté en figure 5.2 sur lequel apparaissent, dans les bulles vertes en haut à droite de chaque tâche, les informations concernant le nombre maximal de personnes pouvant participer à chaque tâche à t_0 et les informations sur les compétences nécessaires, pour un agent, pour pouvoir participer à la tâche à t_0 .

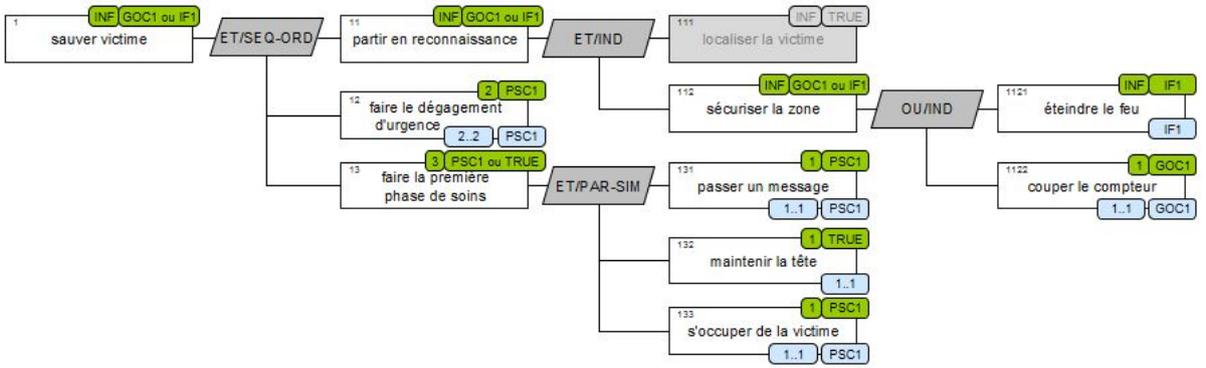


FIGURE 5.6 – Même arbre de tâches que celui présenté en figure 5.2 sur lequel apparaissent, dans les bulles vertes en haut à droite de chaque tâche, les informations concernant le nombre maximal de personnes pouvant participer à chaque tâche à t_1 et les informations sur les compétences nécessaires, pour un agent, pour pouvoir participer à la tâche à t_1 . La tâche grisée est celle déjà réalisée.

Propagation des informations à t_2

Lorsque la zone est sécurisée, à t_2 les agents doivent faire le dégagement d'urgence. Les informations attachées à τ_1 deviennent :

- $c(\tau_1, t_2) = c(\tau_{12}, t_2)$ soit $c(\tau_1, t_2) = \text{PSC1}$; et
- $n_{max}(\tau_1, t_2) = n_{max}(\tau_{12}, t_2) = 2$.

À t_2 , Cara ne peut donc toujours pas participer à *sauver la victime*. Ces informations sont représentées dans la figure 5.7 et cette même figure est disponible en pleine page en annexe B.7.

5.5.4 Conclusion

La propagation des informations permettra aux agents de déterminer, en raisonnant sur les tâches abstraites τ_k de l'arbre de tâches, s'ils peuvent, personnellement, participer à τ_k . Chaque information étant valable à un instant t , ces informations doivent être mises à jour chaque fois

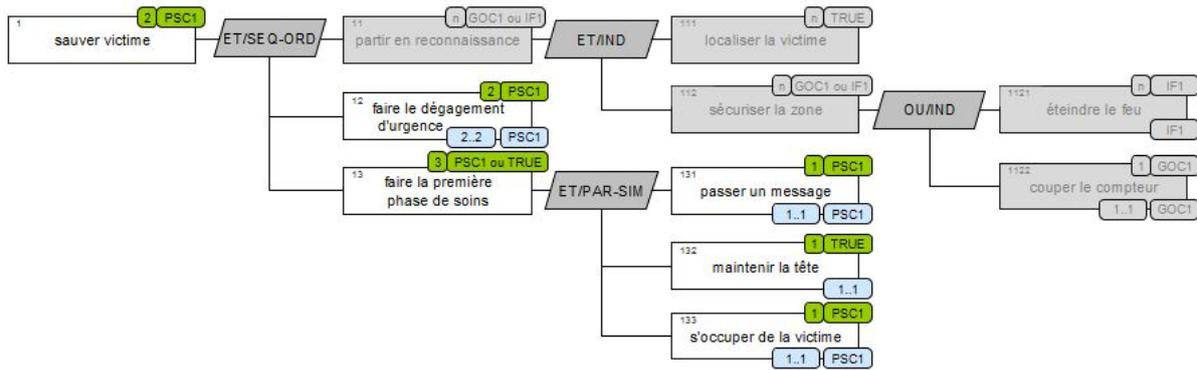


FIGURE 5.7 – Même arbre de tâches que celui présenté en figure 5.2 sur lequel apparaissent, dans les bulles vertes en haut à droite de chaque tâche, les informations concernant le nombre maximal de personnes pouvant participer à chaque tâche à t_2 et les informations sur les compétences nécessaires, pour un agent, pour pouvoir participer à la tâche à t_2 . Les tâches grisées sont les tâches déjà réalisées.

qu'une action est réalisée. Les mécanismes de propagation de ces informations sont plus simples que ceux qui permettent de propager les contraintes de faisabilité collective, notamment car on ne manipule pas ici de compétences associées à un nombre de personnes.

5.6 Bilan du chapitre

Nous nous sommes intéressés dans ce chapitre à la description des activités collectives. Nous avons dans un premier temps fait une rapide étude des langages de description de l'activité existants pour en choisir un qui réponde à nos besoins : nous souhaitons que l'activité ne soit pas décrite sous forme de rôles, pour ne pas limiter la liberté d'action des agents, et nous souhaitons que la description de l'activité prenne la forme d'une arborescence de buts/tâches, de manière à ce que les agents puissent raisonner sur des tâches abstraites.

Notre choix s'est porté sur le langage ACTIVITY-DL qui répond à ces besoins. Par ailleurs, ACTIVITY-DL permet, par une combinaison de contraintes logiques et temporelles attachées à chaque tâche composée, une description précise de l'ordonnancement des tâches. Les contraintes logiques ET et OU permettent de préciser, pour une tâche composée τ_k , si une seule des sous-tâches de τ_k ou si toutes les sous-tâches de τ_k doit/doivent être réalisée(s) pour valider τ_k . Les contraintes temporelles PAR-SIM, SEQ-ORD et IND permettent respectivement de spécifier si les sous-tâches de τ_k doivent être réalisées simultanément ou séquentiellement, ou si cela n'a pas d'importance. Ces contraintes devront être respectées par les agents.

Bien que ce soit un langage très expressif, ACTIVITY-DL ne permet néanmoins pas de décrire des activités collaboratives. Nous avons donc étudié quels étaient les différents éléments liés aux deux niveaux de collaboration : la collaboration au niveau du scénario et la collaboration au niveau des actions. Pour permettre ces deux types de collaboration, nous avons décidé de nous servir d'une spécification de compétences et de nombres d'intervenants nécessaires qui sont à définir sur les actions.

Nous sommes ensuite revenus à notre problématique principale : celle de la génération de

comportements pour des agents autonomes à partir d'une telle description de l'activité. Nous avons remarqué que la spécification des éléments de compétences et de nombre d'intervenants uniquement au niveau des tâches feuilles pose un problème par rapport au raisonnement des agents sur l'arbre de tâches : pour "voir" ces contraintes, les agents devraient raisonner sur l'ensemble de l'arbre de tâches, ce qui est problématique puisque ces arbres peuvent avoir une taille conséquente ; or, en prenant en compte ces arbres et les autres agents de l'environnement, un agent x doit prendre une décision en temps réel.

Nous avons proposé pour remédier à ce problème de propager d'une part des contraintes de faisabilité collective qui permettront aux agents de savoir a priori si l'équipe pourra réaliser la tâche, et d'autre part des informations dynamiques qui permettront à chaque agent, pendant le déroulement de la simulation, de savoir s'il peut personnellement participer à la tâche. Ces contraintes/informations sont générées à partir des éléments de compétence et de nombre de participants précisés sur les tâches feuilles de l'arbre de tâches. Nous avons cependant constaté que la propagation de ces informations n'est pas simple, puisque les mécanismes de propagation définis doivent tenir compte des contraintes logiques et temporelles attachées à chaque tâche composée.

Nous avons détaillé et illustré par des exemples ces mécanismes de propagation de contraintes et d'informations dans les deux dernières sections de ce chapitre.

Finalement, nous concluons en précisant que les contraintes et les informations générées grâce aux mécanismes que nous avons proposés dans ce chapitre peuvent être utilisées de différentes manières par les agents, par exemple selon l'activité considérée. Pour des activités telles que celle de sauvetage de victime qui a servi d'exemple dans ce chapitre, même si la contrainte de faisabilité collective n'est pas vérifiée (e.g. seuls deux agents pompiers sont présents sur les lieux de l'accident au lieu des trois nécessaires), de par le caractère urgent de la situation, il n'est pas réaliste que les agents ne fassent rien. Cette contrainte serait donc non-bloquante. Elle pourrait néanmoins être utilisée pour identifier le ou les éléments manquant(s) et essayer de remédier à ce problème (e.g. en appelant tout de suite le poste de secours pour avoir du renfort). Sur d'autres activités en revanche, comme le chargement de matière dangereuse dans un camion, la contrainte de faisabilité collective pourrait être bloquante, car l'activité concernée est dangereuse mais non urgente.

Bien que l'exemple utilisé présente une situation d'urgence, nos travaux de thèse ne sont pas orientés pour ce type de situations en particulier. En effet, nous nous intéressons à des activités collectives sur lesquelles interviennent un petit nombre d'agents. Les mécanismes proposés dans ce chapitre sont génériques et adaptables à différents types d'activités collectives et ils visent à produire les contraintes et informations sur lesquelles pourront raisonner les agents afin que la collaboration puisse émerger. Enfin, nous nous intéressons tout particulièrement aux mécanismes décisionnels qui permettent à un agent de choisir quoi faire tout en tenant compte de ses coéquipiers, et nous abordons ce point dans le chapitre suivant.

Chapitre 6

Moteur décisionnel

Nous avons détaillé dans les deux premiers chapitres de ce manuscrit les travaux qui ont trait à la génération de comportements de personnages autonomes, en nous attardant particulièrement sur les travaux dans lesquels sont utilisés des modèles de confiance : ces modèles ne permettent pas de générer des comportements qui répondent à nos besoins. Nous travaillons sur des activités collectives, nous souhaitons que les agents puissent tenir compte des autres pour prendre une décision quant à la tâche à faire dans l'environnement virtuel. Nous souhaitons en particulier doter les agents de mécanismes décisionnels qui leur permettent d'adopter les buts des autres de manière altruiste. Par ailleurs, nous souhaitons que l'équipe d'agents puisse s'auto-organiser sans communiquer explicitement sur une répartition des tâches. En effet, par observation du comportement des autres et en imaginant ce que voudraient faire les autres, nos agents doivent faire le meilleur choix possible au moment de sélectionner une tâche compte tenu de leurs envies et de leurs croyances sur les autres. Enfin, nous souhaitons produire des comportements qui ne sont pas toujours idéaux, de manière à entraîner l'apprenant à prêter attention à ses coéquipiers et à réagir en conséquence. Ces comportements doivent cependant rester crédibles et explicables (i.e. l'agent fait toujours le meilleur choix possible contenu de ses envies et croyances, simplement ces envies en croyances ne sont pas idéales par rapport à l'équipe). Le moteur décisionnel proposé dans ce chapitre devra permettre de répondre à ces objectifs.

Nous nous sommes donc intéressés dans le chapitre 4 aux travaux ayant trait à la confiance dans la littérature en sciences humaines et sociales. Cette étude nous a permis d'identifier un modèle de confiance sur lequel nous appuyer pour proposer un tel moteur décisionnel.

Nous présentons ce moteur décisionnel dans ce chapitre. La première section de ce chapitre est consacrée à la présentation du modèle d'agent sur lequel s'appuie notre moteur décisionnel. La seconde dépeint le fonctionnement général du moteur décisionnel proposé, et les détails du fonctionnement du moteur décisionnel sont présentés dans les sections qui suivent.

6.1 Modèle d'agent

Nous présentons ici le modèle d'agent sur lequel notre moteur décisionnel s'appuiera. Suite à notre étude de la littérature en SHS sur la confiance dans le chapitre 4, nous avons établi que le modèle de la confiance interpersonnelle de Mayer et al. (1995) répond à nos besoins : ce modèle adopte une vision multidimensionnelle et située de la confiance dyadique, ce qui permettra aux

agents de distinguer leur niveau de confiance en un autre selon la tâche dont il est question. Par la dimension de la bienveillance, ce modèle permet de prendre en compte la relation personnelle qui lie le trustor et le trustee¹. Cette dimension permettra de générer des comportements d'adoption altruiste (voir 4.3.1).

Comme précisé dans le chapitre 4, dans la suite de ce manuscrit, nous utiliserons l'abréviation MDS pour faire référence aux travaux de Mayer, Davis, and Schoorman (1995).

Nous décrirons les buts des agents (section 6.1.1), puis leurs dimensions personnelles (section 6.1.2) avant de détailler les croyances qu'ont les agents sur les autres (section 6.1.3).

Dans la suite de ce chapitre, nous désignons par x l'agent "principal" dont nous déroulons les processus décisionnels. Nous désignons par y_i ou y_j les autres agents de l'environnement dont l'agent x tient compte lors de sa prise de décision.

6.1.1 Buts des agents

Les agents que nous considérons dans notre système ont deux types de buts : des **buts personnels** et des **buts collectifs**. Un agent peut avoir plusieurs buts collectifs et plusieurs buts personnels. Nous considérons que l'ensemble des buts de chaque agent est cohérent, i.e. un agent x ne peut pas avoir les buts γ et $\neg \gamma$.

Partage des buts collectifs. Un but collectif n'est pas nécessairement partagé par plusieurs agents : un agent x peut avoir un but collectif γ qui n'est pas partagé par les autres. Cela peut paraître étrange qu'un but dit collectif ne soit pas partagé par plusieurs agents ; cependant, les adjectifs *collectifs* et *personnels* doivent être considérés du point de vue des agents et non du point de vue du système. L'agent x peut alors penser que le but γ est collectif, i.e. l'agent x pense que d'autres agents partagent ce but alors qu'en réalité ce n'est pas le cas.

Nous considérons qu'un agent ayant un but collectif a une forme d'engagement envers l'équipe avec laquelle il partage (ou il pense partager) ce but collectif : chacun des agents partageant ce but devrait participer à sa réalisation.

Par ailleurs, il pourrait arriver qu'un but collectif d'un ensemble d'agents corresponde à un but personnel d'un autre agent.

Spécification des buts. Nous ne travaillons pas, dans le cadre de cette thèse, sur la manière dont sont générés les buts des agents ; nous considérons qu'il revient au formateur ou à un éventuel système de scénarisation de renseigner les buts des agents.

Correspondance avec un arbre de tâches. Nous considérons que chaque but correspond à une activité décrite en ACTIVITY-DL. Ces activités peuvent nécessiter une ou plusieurs personnes, indépendamment du fait qu'elles soient associées à des buts personnels ou à des buts collectifs.

Nous considérons dans le cadre de cette thèse que pour un même but, les agents partagent une même représentation de l'activité. Autrement dit, si les agents ont le but collectif de *monter les*

1. Comme dans le chapitre 4, nous utilisons respectivement les termes trustor et trustee pour désigner l'agent qui fait confiance et l'agent qui est l'objet de la confiance.

meubles, tous partagent le même arbre de tâches décrivant cette activité. De même, si plusieurs agents ont un but personnel identique, ils partageront le même arbre de tâches décrivant ce but.

Par ailleurs, comme précisé dans le chapitre précédent, nous considérons des activités où les agents ne sont pas assignés à des rôles pour effectuer une tâche collective : les agents devront se répartir dynamiquement les tâches qui correspondent à des buts collectifs.

Notation. Les buts seront notés dans la suite γ , et $\Gamma_{x,self}$ et $\Gamma_{x,team}$ représentent respectivement les ensembles des buts personnels et collectifs de l'agent x . L'ensemble Γ_x des buts de x est donc $\Gamma_x = \Gamma_{x,self} \cup \Gamma_{x,team}$. Enfin, on note Γ l'ensemble des buts de tous les agents réunis.

"Les agents Anna (noté **a**) et Béa (noté **b**) ont le but collectif $\gamma_m = \text{monter les meubles}$ " sera noté $\gamma_m \in \Gamma_{a,team}$ et $\gamma_m \in \Gamma_{b,team}$.

6.1.2 Dimensions des agents

Nous avons mis en évidence dans le chapitre 4, en nous appuyant sur le modèle de confiance MDS, que c'est en raisonnant sur la confiance qu'ils ont en l'intégrité, la bienveillance et les compétences des autres que les agents pourront tenir compte des autres au moment de prendre une décision. En effet, ces trois caractéristiques sont d'après le modèle MDS déterminantes pour savoir si l'on peut faire confiance à une personne. La confiance influençant toutes nos décisions, ces trois caractéristiques sont déterminantes pour chacun de nos choix (Gambetta, 2000). Si Béa est bienveillante envers Anna, elle voudra l'aider ; et si Anna a confiance en la bienveillance de Béa envers elle, Anna aura aussi confiance dans le fait que Béa veuille l'aider (Mayer et al., 1995).

Nous avons par ailleurs établi également dans le chapitre 4 que nous souhaitons pouvoir faire en sorte qu'un agent adopte de manière altruiste le but d'un autre (voir 4.3.1). Nous souhaitons aussi pouvoir modéliser le fait qu'un agent sera plus ou moins investi dans les buts liés à l'équipe, et qu'un agent sera plus ou moins compétent pour faire certaines tâches. Nous choisissons pour modéliser de tels agents de nous appuyer sur les dimensions de confiance décrites dans le modèle MDS.

Par ailleurs, nous souhaitons pouvoir reproduire des comportements non idéaux liés à une *confiance mal placée* : parce qu'un agent a une fausse idée (i.e. de fausses croyances) sur les comportements futurs des autres, il fait de mauvais choix quant à l'action à faire (voir 4.1.2). Utiliser les mêmes dimensions pour générer les comportements de nos agents que pour modéliser la confiance d'un agent en un autre nous permettra de reproduire très facilement ce type de situation. Nous pourrons ainsi facilement créer des écarts entre ce qu'un agent est réellement et ce que les autres agents pensent de lui. Par exemple, nous pourrons facilement modéliser un agent Béa qui n'est pas bienveillant envers l'agent Anna, alors qu'Anna a confiance en la bienveillance de Béa à son égard et pensera donc que Béa cherchera à l'aider, ce qui ne sera pas le cas.

Nous définissons des dimensions personnelles pour les agents en correspondance avec les éléments du modèle MDS. Chaque agent est donc plus ou moins intègre, bienveillant et compétent². Par ailleurs, comme proposé dans le chapitre précédent et suivant la définition donnée

2. Nous n'intégrons pas ici la notion de disposition à faire confiance (voir 8.3.1).

par (Gambetta, 2000), chaque dimension de confiance sera modélisée par une valeur entre 0 et 1. Nous définissons chacun de ces éléments de la manière suivante :

- **Intégrité.** Le concept de l'intégrité tel que défini dans le modèle MDS est assez vague³, et fait référence à l'adhésion du trustee à un ensemble de principes que le trustor juge acceptables. Le concept de l'intégrité est également souvent lié à celui de l'honnêteté. Par exemple, Karsenty (2010), qui reprend le modèle MDS, lie explicitement les concepts de fiabilité et d'honnêteté à l'intégrité et précise qu'" *il est notamment nécessaire d'estimer [si le trustee] est honnête de telle façon que ses paroles traduisent réellement ses intentions et permettent de prévoir de façon fiable ses futurs agissements*".

Un agent a des buts collectifs qui sont partagés par l'équipe et pour lesquels il s'est engagé. On estime donc qu'un agent intègre privilégiera plutôt les buts collectifs par rapport à ses buts personnels. Cette définition de l'intégrité est restrictive par rapport à celle donnée dans le modèle MDS, mais elle nous suffit dans un premier temps pour construire notre moteur décisionnel.

Nous définissons donc l'intégrité de l'agent x comme une valeur numérique que l'on notera i_x , avec $i_x \in [0; 1]$. Une valeur de 0 désigne le fait que l'agent n'est pas du tout intègre et donc privilégiera ses buts personnels, alors qu'une valeur de 1 représente un agent très intègre. Lorsqu'un agent aura une intégrité de 0.5, il ne souhaitera favoriser ni ses buts personnels ni les buts collectifs. Son choix entre ses buts personnels et ses buts collectifs sera alors grandement influencé par ce qu'il pense des autres.

- **Bienveillance.** Dans la formalisation de la confiance de MDS, la seconde dimension de la confiance est celle de la bienveillance : le trustor a-t-il confiance en la bienveillance du trustee à son égard ? Dans le modèle MDS, un seul trustor et un seul trustee entrent en compte, puisque l'on considère une relation de confiance dyadique. Dans le cadre de nos travaux cependant, chaque agent a des relations différentes avec chaque autre, et un agent x peut donc être plus ou moins bienveillant envers chaque autre agent y_i de l'environnement.

Nous définissons donc la bienveillance de l'agent x envers l'agent y_i comme une valeur numérique notée b_{x,y_i} , avec $b_{x,y_i} \in [0; 1]$. S'il y a n agents dans l'environnement virtuel, la bienveillance de l'agent x est donc l'ensemble $\{b_{x,y_1}, b_{x,y_2}, \dots, b_{x,y_{n-1}}\}$ (x n'a pas de valeur de bienveillance envers lui-même). Tout comme pour l'intégrité, une valeur de 0 signifie que l'agent x n'est pas du tout bienveillant envers l'agent y_i alors qu'une valeur de 1 signifie que l'agent x est très bienveillant envers y_i . Avec une valeur de 0.5, l'agent x est neutre par rapport à l'agent y_i . Cela peut signifier que (i) l'agent x ne connaît pas l'agent y_i et n'a donc pas d'opinion de y_i sur laquelle fonder sa bienveillance, ou que (ii) l'agent x connaît l'agent y_i mais y est indifférent.

Nous choisissons ici de ne pas différencier les deux cas (i) et (ii), car ils seraient traités de la même manière par le moteur décisionnel (voir 6.3).

- **Compétence.** Le dernier élément de la définition d'une relation de confiance du modèle MDS est la compétence. Cet élément permet de situer la confiance par rapport à une tâche

3. Mayer et al. (1995) s'appuient sur la définition donnée dans les travaux fondateurs de McFall (1987). Cependant tout comme nous l'avons remarqué au sujet de la confiance dans le chapitre 4, il n'y a pas non plus de consensus dans la littérature en SHS sur une définition de l'intégrité.

particulière : le trustor a-t-il confiance en la compétence du trustee pour faire la tâche τ_k ? Nous avons proposé dans le chapitre 5 que des compétences particulières puissent être précisées sur certaines tâches de l'arbre de tâches ACTIVITY-DL.

De même que pour la bienveillance et l'intégrité, nous considérons que la compétence d'un agent n'est pas binaire : par exemple, si la compétence *bricolage – savoir utiliser un marteau* est précisée sur l'arbre de tâches correspondant au but *monter les meubles*, l'agent peut maîtriser plus ou moins cette compétence. Ces différents niveaux de compétence pourraient être liés par exemple à différentes qualités de résultat ou à différentes vitesses d'exécution pour les tâches nécessitant la compétence. On représente le niveau de compétence de l'agent x sur la compétence κ par la valeur $a_{x,\kappa} \in [0; 1]$ (a pour *ability*). Tout comme pour l'intégrité et la bienveillance, une valeur de 0 signifie que l'agent n'est pas du tout compétent alors qu'une valeur de 1 signifie que l'agent est parfaitement compétent.

Ces éléments peuvent être retrouvés dans le tableau récapitulatif des notations utilisées présenté en annexe C.1.

6.1.3 Croyances

D'après le modèle MDS, chaque agent a des croyances sur les autres agents. Nous décrivons ces croyances ici.

Croyances sur les buts des autres

Les agents ont des croyances sur les buts des autres : chaque agent x a des croyances sur les buts de chaque autre agent y_i . On note $\Gamma_{y_i, self}^x$ ce que x pense être l'ensemble des buts personnels de y_i ; et on note $\Gamma_{y_i, team}^x$ ce que x pense être l'ensemble des buts collectifs de y_i . L'expression $\Gamma_{y_i}^x$ correspond donc à ce que x pense être l'ensemble des buts personnels et collectifs de y_i . Comme nous l'avons précisé précédemment (voir 6.1.1), si l'agent x a le but collectif γ (i.e. $\gamma \in \Gamma_{x, team}$), alors l'agent x pense que d'autres agents (pas nécessairement tous les agents) ont également le but collectif γ : " x pense que y_i a aussi le but collectif γ " est noté $\gamma \in \Gamma_{y_i, team}^x$.

Croyances sur les dimensions des autres

Comme décrit dans le modèle de confiance de MDS, les agents ont également des croyances sur les caractéristiques personnelles des autres :

- **Croyances sur l'intégrité.** L'agent x a une croyance sur le niveau d'intégrité de l'agent y_i . On représente par $i_{y_i}^x \in [0; 1]$ cette croyance.
- **Croyances sur la bienveillance.** L'agent x a des croyances sur la bienveillance de l'agent y_i envers tous les autres agents y_j de l'environnement (x compris). On note $b_{y_i, y_j}^x \in [0; 1]$ la croyance de x sur la bienveillance de y_i envers y_j . La croyance de x sur la bienveillance de y_i envers lui est alors $b_{y_i, x}^x$.
- **Croyances sur les compétences.** Enfin l'agent x a des croyances sur les compétences de l'agent y_i . Si l'on considère la compétence κ , on note $a_{y_i, \kappa}^x \in [0; 1]$ la croyance de x sur le niveau de compétence de y_i pour κ .

Il est important de noter qu'ici nos agents *n'ont pas* de croyances graduées : $i_{y_i}^x = 0.5$ signifie que "l'agent x pense que l'agent y_i est moyennement intègre", et non que "l'agent x est moyennement sûr que l'agent y_i est intègre". Le fait de ne pas avoir de croyances graduées est l'une des limites (voir 8.2.2) et intégrer cette notion à notre moteur décisionnel fait partie des perspectives dont nous discutons en 8.3.1.

Nous ne proposons pas de formalisation de la confiance comme cela est fait par exemple dans ForTrust (voir 3.3.3) en combinant chacune de ces croyances. En effet, il est précisé dans le modèle MDS que ces différents éléments n'influencent pas de la même manière la confiance du trustor en le trustee. Par exemple, la bienveillance a trait à la confiance du trustor en le trustee pour l'aider, alors que la confiance du trustor en l'intégrité du trustee est liée à la perception du trustee comme quelqu'un respectant ses engagements.

Par ailleurs, nous rappelons que nous ne proposons pas de mécanisme de mise à jour des croyances des agents dans le cadre de cette thèse.

6.1.4 Exemple

Nous présentons ici un exemple de scénario⁴ *S1* qui nous servira à illustrer les processus décisionnels de nos agents tout au long de ce chapitre. Ce scénario servira également de scénario de référence pour la présentation d'autres exemples de fonctionnement du moteur décisionnel dans le chapitre suivant.

Scénario

Nous choisissons ici de prendre comme exemple une situation où il n'y a pas d'urgence : on considère une équipe de quatre agents doctorantes Anna, Béa, Cara et Dora qui viennent de s'installer dans un nouveau bureau.

- *Buts et tâches.* Anna, Cara et Dora ont les mêmes buts collectifs ; à savoir, le but *monter les meubles* noté γ_m et le but *installer l'équipement informatique* noté γ_e . L'agent Béa ne partage avec les autres que le but collectif *installer l'équipement informatique*.

On considère que le but *monter les meubles* peut être réalisé en faisant la tâche correspondante *monter les meubles* notée τ_m . Cette tâche est une tâche composée de deux sous-tâches feuilles (i.e. actions) $\tau_{mb} = \textit{monter le bureau}$ et $\tau_{mt} = \textit{monter les tiroirs}$. Pour les besoins de l'exemple, on considère que la sous-tâche τ_{mb} requiert la compétence *bricolage – savoir utiliser un marteau* notée BM et que la sous-tâche τ_{mt} requiert la compétence *bricolage – savoir utiliser un tournevis* notée BT. Une contrainte ET/IND est attachée à τ_m . Le but γ_e peut quant à lui être réalisé en faisant la tâche $\tau_e = \textit{installer l'équipement informatique}$ également composée de deux sous-tâches feuilles (i.e. actions) $\tau_{eo} = \textit{installer l'ordinateur}$ et $\tau_{ei} = \textit{installer l'imprimante}$. Ces deux tâches demandent la compétence *informatique* notée I. Une contrainte ET/IND est également attachée à τ_e .

4. Nous englobons ici sous le terme *scénario* l'ensemble des activités auxquelles peuvent participer les agents, la définition (ou paramétrisation) des agents présents dans l'environnement, ainsi que tout le déroulement de la simulation qui en suit (en particulier les actions effectuées par les agents). À noter que selon cette définition, le déroulement des actions effectuées par les agents peut être identique dans deux scénarios différents (i.e. la paramétrisation des agents serait différente, mais cela résulterait pourtant dans des comportements identiques).

L'agent Béa a comme but personnel d'écrire un papier, alors que le but personnel d'Anna est de formaliser le modèle. On note respectivement γ_p et γ_f ces buts. Ces buts correspondent respectivement aux tâches feuilles τ_f et τ_p .

L'ensemble de tous les buts personnels et collectifs de tous les agents confondus est donc $\Gamma = \{\gamma_m, \gamma_e, \gamma_p, \gamma_f\}$.

Pour les besoins de l'exemple on considère ici que toutes les tâches nécessitent exactement une personne. La représentation des arbres de tâches correspondant à ce scénario est donnée dans la figure 6.1.

Comme la situation ne présente pas d'urgence, nous considérons ici que les contraintes de nombre d'agents minimum et de compétences sont bloquantes : les agents ne pourront pas faire tâche pour laquelle il n'ont pas la compétence.

- *Caractéristiques des agents.* Nous considérons pour l'exemple que tous les agents sont plutôt bienveillants envers chacun des autres. L'intégrité d'Anna et Dora est plutôt élevée alors que celle de Béa et Cara est neutre.

Enfin, Béa et Cara ne sont pas du tout compétentes pour le bricolage (*savoir utiliser un marteau / un tournevis*) et sont moyennement compétentes pour l'informatique. Dora est quant à elle moyennement compétente pour le bricolage et peu compétente pour l'informatique, alors que qu'Anna est moyennement compétente pour *savoir utiliser un marteau*, et peu compétente pour l'informatique et pour *savoir utiliser un tournevis*.

- *Croyances des agents.* Dans ce scénario, nous considérerons par souci de simplification que les croyances des agents sur chacun des autres correspondent à la réalité (e.g. Anna pense que Béa a le but personnel d'écrire un papier, et Béa a effectivement le but personnel d'écrire un papier).

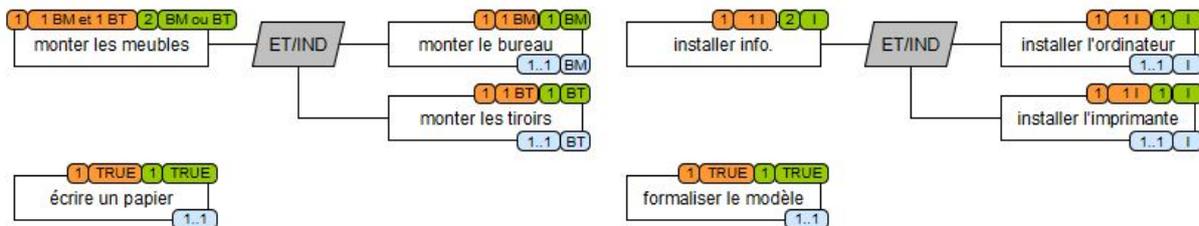


FIGURE 6.1 – Arbres de tâches décrits en ACTIVITY-DL correspondant au scénario détaillé en 6.1.4. Les bulles bleues en bas de chaque tâche correspondent aux informations renseignées lors de la description de l'activité. Les bulles oranges et les bulles vertes en haut de chaque tâche correspondent respectivement aux contraintes de faisabilité collectives et aux informations dynamiques propagées dynamiquement par le système (voir chapitre 5).

Modèle de l'agent Anna

Suivant le scénario défini, nous pouvons détailler le modèle de l'agent Anna (chaque agent est désigné par son initiale pour alléger la notation) :

- *Buts.* Les buts collectifs d'Anna sont *monter les meubles* (γ_m) et *installer l'équipement informatique* (γ_e), soit $\Gamma_{a,team} = \{\gamma_m, \gamma_e\}$, et le but personnel d'Anna est de *formaliser*

le modèle (γ_f) , soit $\Gamma_{a,self} = \{\gamma_f\}$. L'ensemble des buts personnels et collectifs d'Anna est donc $\Gamma_a = \{\gamma_m, \gamma_e, \gamma_f\}$.

- *Dimensions*. Les dimensions personnelles d'Anna sont les suivantes :
 - Anna est plutôt intègre, on a donc $i_a = 0.75$.
 - Anna est plutôt bienveillante envers Béa, Cara et Dora, on a donc $b_{a,b} = b_{a,c} = b_{a,d} = 0.75$.
 - Anna est moyennement compétente pour *utiliser un marteau* et peu compétente pour *utiliser un tournevis* : $a_{a,BM} = 0.5$ et $a_{a,BT} = 0.25$. Elle est également peu compétente pour *l'informatique* : $a_{a,I} = 0.25$.
- *Croyances*. Enfin, Anna a des croyances sur les autres agents. Pour simplifier, nous considérons ici que les croyances d'Anna sur les autres agents correspondent à la réalité. Nous détaillons ici uniquement les croyances d'Anna sur Béa de manière à donner une illustration des notations utilisées. Anna pense que :
 - Béa a elle aussi le but collectif d'*installer l'équipement informatique*, soit $\Gamma_{b,team}^a = \{\gamma_e\}$, et Béa a le but personnel d'*écrire un papier*, soit $\Gamma_{b,self}^a = \{\gamma_p\}$. Anna pense donc que l'ensemble des buts personnels et collectifs de Béa est $\Gamma_b^a = \{\gamma_e, \gamma_p\}$.
 - Béa est moyennement intègre : $i_b^a = 0.5$.
 - Béa est plutôt bienveillante envers tous les autres agents : $b_{b,a}^a = b_{b,c}^a = b_{b,d}^a = 0.75$.
 - Béa est moyennement compétente pour l'informatique : $a_{b,I}^a = 0.5$ mais n'est pas compétente pour le bricolage en général $a_{b,BM}^a = a_{b,BT}^a = 0$.

Modèle des autres agents

Nous ne détaillons pas la description du modèle de chacun des autres agents, mais les caractéristiques (buts et dimensions personnelles) de chacun des agents sont présentées dans le tableau 6.1.

TABLE 6.1 – *S1* – Caractéristiques (en termes de buts et de dimensions personnelles) des agents des agents dans le scénario *S1* présenté à titre d'exemple dans ce chapitre.

		Anna	Béa	Cara	Dora
Buts	Collectifs	$\Gamma_{a,team} = \{\gamma_m, \gamma_e\}$	$\Gamma_{b,team} = \{\gamma_e\}$	$\Gamma_{c,team} = \{\gamma_m, \gamma_e\}$	$\Gamma_{d,team} = \{\gamma_m, \gamma_e\}$
	Personnels	$\Gamma_{a,self} = \{\gamma_f\}$	$\Gamma_{b,self} = \{\gamma_p\}$	$\Gamma_{c,self} = \emptyset$	$\Gamma_{d,self} = \emptyset$
	Ensemble	$\Gamma_a = \{\gamma_m, \gamma_e, \gamma_f\}$	$\Gamma_b = \{\gamma_e, \gamma_p\}$	$\Gamma_c = \{\gamma_m, \gamma_e\}$	$\Gamma_d = \{\gamma_m, \gamma_e\}$
Intégrité		$i_a = 0.75$	$i_b = 0.5$	$i_c = 0.5$	$i_d = 0.75$
Bienveillance	Anna	—	$b_{b,a} = 0.75$	$b_{c,a} = 0.75$	$b_{d,a} = 0.75$
	Béa	$b_{a,b} = 0.75$	—	$b_{c,b} = 0.75$	$b_{d,b} = 0.75$
	Cara	$b_{a,c} = 0.75$	$b_{b,c} = 0.75$	—	$b_{d,c} = 0.75$
	Dora	$b_{a,d} = 0.75$	$b_{b,d} = 0.75$	$b_{c,d} = 0.75$	—
Compét.	Marteau	$a_{a,BM} = 0.5$	$a_{b,BM} = 0$	$a_{c,BM} = 0$	$a_{d,BM} = 0.5$
	Tournevis	$a_{a,BT} = 0.25$	$a_{b,BT} = 0$	$a_{c,BT} = 0$	$a_{d,BT} = 0.5$
	Informatique	$a_{a,I} = 0.25$	$a_{b,I} = 0.5$	$a_{c,I} = 0.5$	$a_{d,I} = 0.25$

Nous présentons dans la section suivante le fonctionnement général du moteur décisionnel

associé au modèle d'agent présenté dans cette section. L'exemple abordé ici sera utilisé tout au long de ce chapitre pour illustrer nos propos.

6.2 Fonctionnement général du moteur décisionnel

Les agents sont indépendants les uns des autres en fonctionnent séparément : chaque agent dispose de son propre processus décisionnel et fonctionne en parallèle des autres. Les agents suivent un cycle *perception - décision - action*. Nous précisons le déroulement de chacune de ces trois phases dans les paragraphes suivants.

Perception

Dans la phase de perception, les agents observent ce qu'il se passe dans l'environnement qui les entoure. Nous précisons encore une fois que nous ne considérons que des activités qui s'effectuent dans un même espace : tous les agents peuvent percevoir tout ce qu'il se passe dans l'environnement ; i.e. il n'y a pas de notion de champ de vision qui limiterait l'espace de perception des agents.

Plus spécifiquement :

- Les agents ont accès à l'état d'avancement de chacune des activités qu'ils peuvent effectuer, c'est-à-dire les activités liées à des buts collectifs telles que *monter les meubles*, mais aussi les activités liées à des buts personnels des autres agents. Par exemple, si l'agent Anna fait l'activité associée à son but personnel *formaliser le modèle*, Béa perçoit également l'état d'avancement de cette activité.
- Les agents peuvent savoir ce que les autres sont en train de faire ; que les autres soient réellement en train de faire une action ou en train d'*attendre* de l'aide pour faire une action. Si l'on reprend l'exemple de sauver la victime développé dans le chapitre 5, *x* peut voir que l'agent *y* attend de l'aide pour *faire le dégagement d'urgence*, étant donné que cette action requiert un minimum de deux personnes.

Cette phase de *perception* est suivie d'une phase de *décision*.

Décision

Dans cette thèse, nous allons nous concentrer particulièrement sur cette phase de *décision* : à l'issue de cette phase, les agents auront sélectionné l'action qu'ils souhaitent faire dans l'environnement virtuel ou, éventuellement, auront décidé de ne rien faire. Pour cela, les agents doivent réfléchir à l'importance qu'ils accordent à chacun de leurs buts, puis ils doivent sélectionner une tâche qui leur permettra d'accomplir un but auquel ils accordent de l'importance.

Importance des buts. Dans un premier temps, les agents doivent choisir quel(s) but(s) ils souhaitent privilégier. En effet, les agents ayant différentes sortes de buts, à savoir les buts personnels et les buts collectifs, il est nécessaire de décider lesquels ils souhaitent réaliser en priorité. Ce choix ne peut pas être le même pour tous les agents (e.g. tous les agents privilégient les buts collectifs), car cela ne serait pas représentatif des comportements humains. De plus, ce

choix est influencé par les autres : si l'agent x pense qu'il sera le seul à privilégier le but collectif, cela le démotivera (Kerr, 1983).

Par ailleurs, nous avons mis en avant dans le chapitre 4 la notion d'*adoption altruiste* de but : nous souhaitons pouvoir faire en sorte que l'agent Anna aide bénévolement l'agent Béa, c'est-à-dire que l'agent Anna adopte de manière altruiste les buts de l'agent Béa (voir 4.3.1). C'est en prenant en compte les autres au moment de calculer l'importance qu'elle accorde à ses buts qu'Anna pourra adopter les buts de Béa, par exemple aider Béa à *rédiger un papier*.

Pour savoir quels sont les buts que les agents auront envie de privilégier, nous proposons un mécanisme de calcul de l'importance des buts, que nous développons section 6.3.

Utilité des tâches. Une fois que les agents ont calculé l'importance qu'ils accordent à chacun de leurs buts, ils doivent chercher comment réaliser les buts auxquels ils accordent le plus d'importance. Pour cela, ils raisonnent sur les arbres de tâches correspondant à leurs buts : les agents doivent alors prendre en compte les compétences qui peuvent être spécifiées sur l'arbre de tâches. En particulier, on considère que l'agent préférera réaliser une tâche pour laquelle il est doué (i.e. pour laquelle son niveau de compétence est élevé).

Nous définissons donc un mécanisme de calcul d'utilité d'une tâche qui permettra aux agents de déterminer quelle(s) tâche(s) il serait le plus judicieux de réaliser pour atteindre leurs buts. Nous détaillons ce mécanisme dans la section 6.4.

Répartition des tâches. Pour choisir une tâche à accomplir, les agents doivent encore une fois prendre les autres en compte. En effet, nous avons notamment spécifié dans le chapitre 4 que nous souhaitons pouvoir produire des comportements de *délégation faible* (voir 4.1.2) ; e.g. si l'on reprend l'exemple développé précédemment, on peut imaginer que l'agent Anna laisse les agents Béa et Cara *installer l'équipement informatique* car elle sait que Béa et Cara ne savent pas bricoler et elle pense que Béa et Cara accordent également beaucoup d'importance au but *installer l'équipement informatique*. Anna délègue donc une partie des tâches à Béa et Cara sans explicitement communiquer avec ces dernières sur cette délégation. Évidemment si Anna a une *confiance mal placée* en Béa et/ou Cara, sa prédiction sur le comportement de ces deux agents pourra s'avérer fautive (voir 4.1.2).

Nous proposons, pour pouvoir produire de tels comportements, que les agents calculent ce qu'ils pensent être l'utilité des tâches pour chacun des autres agents afin de choisir la meilleure répartition de tâches possible. Nous détaillons cette étape du processus de prise de décision section 6.5.

Processus récursif. Enfin, nous avons précisé dans le chapitre précédent que les agents raisonnent en priorité sur les tâches abstraites d'un arbre de tâches, puis sur des tâches de plus en plus concrètes pour finalement sélectionner une tâche qui corresponde à une action (voir 5.1.1). Cela permet de limiter la complexité du raisonnement des agents en leur évitant d'avoir à considérer plusieurs centaines de tâches à la fois. Le processus de *calcul de l'utilité des tâches - répartition des tâches - sélection d'une tâche* est donc un processus récursif, qui se termine lorsque la tâche sélectionnée est une action (i.e. une tâche feuille de l'arbre de tâches).

Nous développons le fonctionnement de ce processus section 6.6.

Action

Une fois que les agents ont décidé d'une action à faire, ils tentent de faire cette action dans l'environnement virtuel. En effet, il y aura des cas où les agents ne pourront pas exécuter l'action sélectionnée : par exemple si l'agent x est le seul à avoir sélectionné une action pour laquelle il faut au moins être deux.

Dans ce cas, l'agent x devra *attendre* qu'un autre agent sélectionne également cette action. Pour éviter les cas de blocage où chaque agent de l'environnement attend qu'un autre sélectionne l'action qu'il a déjà choisie, nous proposons que le temps d'attente de chaque agent soit limité. Cette limite peut varier selon l'application considérée.

Nos travaux sont centrés sur la proposition d'un moteur décisionnel, et nous ne détaillerons pas plus le déroulement de la phase *action* dans ce document. Nous présentons dans la section suivante le fonctionnement précis du moteur décisionnel.

6.3 Importance des buts

Calculer l'importance qu'ils accordent à chacun de leurs buts permettra aux agents de décider le(s)quel(s) privilégier. Nous détaillons ici le calcul de l'importance que les agents accordent à leurs buts.

6.3.1 Calcul initial de l'importance des buts

Initialement, l'agent x raisonne uniquement sur ses propres buts. Comme nous l'avons vu en 6.1.2, l'intégrité d'un agent est liée à son honnêteté et au fait que l'agent respecte ses engagements. On considère donc qu'un agent intègre va privilégier les buts liés à l'équipe (i.e. ses buts collectifs) alors qu'un agent qui n'est pas intègre privilégiera ses buts personnels. Par ailleurs, nous considérons que l'importance qu'un agent accorde à son but correspond à sa motivation à réaliser ce but. Nous emploierons les deux termes tout au long de ce chapitre.

L'importance initiale d'un but γ pour l'agent x est notée $iImp_x(\gamma)$ et cette valeur appartient à l'intervalle $[0; 1]$:

$$\begin{cases} iImp_x(\gamma) = i_x & \text{si } \gamma \in \Gamma_{x,team} \\ iImp_x(\gamma) = 1 - i_x & \text{si } \gamma \in \Gamma_{x,self} \\ iImp_x(\gamma) = 0 & \text{si } \gamma \notin \Gamma_x \end{cases} \quad . \quad (6.1)$$

En appliquant la formule 6.1, on obtient, pour l'agent x , la même valeur d'importance pour chacun de ses buts collectifs et pour chacun de ses buts personnels. Pour tous les buts qui ne sont pas des buts de l'agent x , la valeur initiale d'importance accordée à ces buts est nulle.

Exemple. Si l'on reprend l'exemple présenté en 6.1.4, nous pouvons calculer l'importance initiale qu'Anna donne à chacun des buts de l'ensemble Γ :

- $\gamma_m \in \Gamma_{a,team}$ et $\gamma_e \in \Gamma_{a,team}$, donc $iImp_a(\gamma_m) = iImp_a(\gamma_e) = i_a = 0.75$.
- $\gamma_f \in \Gamma_{a,self}$, donc $iImp_a(\gamma_f) = 1 - i_a = 0.25$.
- Enfin, $iImp_a(\gamma_p) = 0$ car $\gamma_p \notin \Gamma_a$.

6.3.2 Prise en compte des autres

L'agent x doit ensuite prendre en compte les autres agents de l'environnement.

Calcul de l'importance initiale de leurs buts pour les autres agents selon x

Dans un premier temps, x calcule, pour chaque autre agent y_i , l'importance que y_i accorde à chacun de ses buts selon lui. Pour cela, x raisonne de la même manière que pour lui-même, mais sur ce qu'il pense être l'ensemble des buts personnels et l'ensemble des buts collectifs de y_i ($\Gamma_{y_i, self}^x$ et $\Gamma_{y_i, team}^x$), ainsi que sur ce qu'il pense être l'intégrité de y_i ($i_{y_i}^x$).

On note $iImp_{y_i}^x(\gamma) \in [0; 1]$ l'importance initiale du but γ pour y_i selon x , et cette valeur est calculée avec la formule 6.2 :

$$\begin{cases} iImp_{y_i}^x(\gamma) = i_{y_i}^x & \text{si } \gamma \in \Gamma_{y_i, team}^x \\ iImp_{y_i}^x(\gamma) = 1 - i_{y_i}^x & \text{si } \gamma \in \Gamma_{y_i, self}^x \\ iImp_{y_i}^x(\gamma) = 0 & \text{sinon} \end{cases} \quad (6.2)$$

Exemple. Anna calcule ensuite ce qu'elle pense être l'importance initiale de leurs buts pour les autres agents. Les valeurs d'importance initiale des buts calculées par Anna sont données dans le tableau 6.2. Nous ne développons ce calcul que pour l'agent Béa : Anna calcule $iImp_b^a(\gamma_e) = i_b^a = 0.5$ car $\gamma_e \in \Gamma_{b, team}^a$ et $iImp_b^a(\gamma_p) = 1 - i_b^a = 0.5$ car $\gamma_p \in \Gamma_{b, self}^a$. Formaliser le modèle et monter les meubles n'étant pas, d'après Anna, dans les buts de Béa, Anna calcule également $iImp_b^a(\gamma_f) = iImp_b^a(\gamma_m) = 0$.

TABLE 6.2 – $S1$ – Importance initiale des buts calculée par Anna. La valeur $iImp_a(\gamma)$ représente l'importance pour Anna du but γ . La valeur $iImp_{y_i}^a(\gamma)$ représente l'importance du but γ pour y_i selon Anna.

	<i>monter les meubles</i>	<i>installer info.</i>	<i>formaliser le modèle</i>	<i>écrire un papier</i>
Anna	$iImp_a(\gamma_m) = 0.75$	$iImp_a(\gamma_e) = 0.75$	$iImp_a(\gamma_f) = 0.25$	$iImp_a(\gamma_p) = 0$
Bea	$iImp_b^a(\gamma_m) = 0$	$iImp_b^a(\gamma_e) = 0.5$	$iImp_b^a(\gamma_f) = 0$	$iImp_b^a(\gamma_p) = 0.5$
Cara	$iImp_c^a(\gamma_m) = 0.5$	$iImp_c^a(\gamma_e) = 0.5$	$iImp_c^a(\gamma_f) = 0$	$iImp_c^a(\gamma_p) = 0$
Dora	$iImp_d^a(\gamma_m) = 0.75$	$iImp_d^a(\gamma_e) = 0.75$	$iImp_d^a(\gamma_f) = 0$	$iImp_d^a(\gamma_p) = 0$

Modulation de l'importance de ses buts pour x

Avoir des croyances sur l'utilité de leurs buts pour chacun des autres agents permet à x de moduler l'importance qu'il accorde à ses propres buts en fonction des autres. Notamment, selon la bienveillance de x envers un autre agent y_i , x aura plus ou moins envie d'aider y_i à réaliser ses buts. L'agent x calcule donc une valeur $\delta_{x, y_i}(\gamma) \in [-1; 1]$ qui représente l'influence de y_i sur l'importance qu'il accorde à son but γ . Pour chacun des buts $\gamma \in \Gamma_{y_i}^x$, cette valeur est donnée

par la formule 6.3 :

$$\left\{ \begin{array}{ll} \delta_{x,y_i}(\gamma) = \alpha_{ifl} \times (iImp_{y_i}^x(\gamma) - iImp_x(\gamma)) & \text{si } iImp_{y_i}^x(\gamma) < iImp_x(\gamma) \\ \delta_{x,y_i}(\gamma) = 2\alpha_{ifl} \times \max(0; b_{x,y_i} - 0.5) & \text{si } iImp_{y_i}^x(\gamma) \geq iImp_x(\gamma) \\ & \times (iImp_{y_i}^x(\gamma) - iImp_x(\gamma)) \end{array} \right. \quad (6.3)$$

où $\alpha_{ifl} \in [0; 1]$ représente un coefficient d'influence des autres agents sur x . En effet, selon le type d'application considérée, les agents ne vont pas être sensibles de la même manière aux autres. Par exemple, pour des applications où l'urgence de la situation est forte, comme dans l'exemple de sauvetage de victime, même s'il pense que ses coéquipiers ne sont pas très motivés pour sauver la victime (ce qui serait de toute manière irréaliste), cela ne doit pas démotiver l'agent x . Dans ce cas, la valeur de α_{ifl} doit être très faible voire nulle.

L'agent x ne calcule une valeur de $\delta_{x,y_i}(\gamma)$ que pour les buts des autres agents y_i . Cela implique donc que l'agent x ne calcule pas de valeur de $\delta_{x,y_i}(\gamma)$ pour les buts γ qui lui sont personnels : en effet, si le but est personnel, la motivation des autres pour ce but (qui sera nulle) ne doit pas influencer la motivation de x .

Le tableau 6.3 donne les valeurs de $\delta_{x,y_i}(\gamma)$ en fonction des valeurs de b_{x,y_i} et de la comparaison entre $iImp_{y_i}^x(\gamma)$ et $iImp_x(\gamma)$. Ainsi, on peut reproduire les effets suivants :

- *Adoption altruiste de but.* Si x est bienveillant envers un agent y_i alors $\delta_{x,y_i}(\gamma)$ sera positif si x pense que y_i accorde plus d'importance au but γ qu'il n'en accorde lui-même. Autrement dit, si γ est un but de y_i mais n'est pas un but de x (i.e. $\gamma \in \Gamma_{y_i}^x$ et $\gamma \notin \Gamma_x$), alors si x est bienveillant envers y_i , x adoptera le but γ (i.e. on aura $iImp_x(\gamma) > 0$)⁵.

Dans ce cas, on aura une influence *positive* de y_i sur la motivation de x pour atteindre le but γ .

Si au contraire, x n'est pas bienveillant envers y_i , x n'adoptera pas le but de y_i : on aura en effet $\delta_{x,y_i} = 0$ si x pense que y_i accorde plus d'importance que lui au but γ , et donc l'agent y_i n'aura pas d'incidence sur l'importance accordée à son but par x .

Ces effets sont représentés notamment dans la colonne (1) $iImp_{y_i}^x(\gamma) > iImp_x(\gamma)$ du tableau 6.3.

- *Perte de motivation.* La formule 6.3 permet de tenir compte des effets de baisse de motivation lorsque nous devons travailler à une tâche avec des personnes qui ne sont elles-mêmes pas motivées, indépendamment du fait que nous soyons bienveillants ou non envers ces personnes. Dans ce cas, nous nous démotivons et réduisons nos efforts, car nous ne souhaitons pas que les autres profitent de nos efforts si eux-mêmes n'en font pas : c'est ce que l'on appelle le *sucker-effect* dans la littérature SHS (Kerr, 1983).

Si x pense que les autres sont peu motivés par le but γ auquel il accorde lui-même beaucoup d'importance, alors cela induira une diminution de l'importance accordée par x à son but γ : on aura une influence *négative* de y_i sur la motivation de x pour atteindre le but γ .

Ces effets sont représentés par la colonne (3) $iImp_{y_i}^x(\gamma) < iImp_x(\gamma)$ du tableau 6.3.

- *Pas d'incidence.* Enfin, lorsque l'agent y_i accorde la même importance à son but γ que

5. L'agent x adoptera le but γ de l'agent y_i si x est bienveillant envers y_i et si il n'y a pas d'autre(s) agent(s) y_j qui a (ont) également le but γ et envers le(s)quel(s) x n'est pas bienveillant.

l'agent x , la prise en compte de y_i par x n'aura pas d'incidence sur l'importance accordée à γ par ce dernier : c'est la colonne (2) $iImp_{y_i}^x(\gamma) = iImp_x(\gamma)$ du tableau 6.3.

TABLE 6.3 – Valeurs de $\delta_{x,y_i}(\gamma)$ en fonction des valeurs de b_{x,y_i} et de la comparaison entre $iImp_{y_i}^x(\gamma)$ et $iImp_x(\gamma)$.

	(1) $iImp_{y_i}^x(\gamma) > iImp_x(\gamma)$	(2) $iImp_{y_i}^x(\gamma) = iImp_x(\gamma)$	(3) $iImp_{y_i}^x(\gamma) < iImp_x(\gamma)$
$b_{x,y_i} > 0.5$	$\delta_{x,y_i}(\gamma) > 0$	$\delta_{x,y_i}(\gamma) = 0$	$\delta_{x,y_i}(\gamma) < 0$
$b_{x,y_i} = 0.5$	$\delta_{x,y_i}(\gamma) = 0$	$\delta_{x,y_i}(\gamma) = 0$	$\delta_{x,y_i}(\gamma) < 0$
$b_{x,y_i} < 0.5$	$\delta_{x,y_i}(\gamma) = 0$	$\delta_{x,y_i}(\gamma) = 0$	$\delta_{x,y_i}(\gamma) < 0$

Une fois les valeurs $\delta_{x,y_i}(\gamma)$ calculées pour les $n(\gamma)$ agents y_i ayant le but γ ($\gamma \in \Gamma_{y_i}^x$), l'agent x calcule une valeur $\Delta_x(\gamma) \in [-1 ; 1]$ comme définie dans la formule 6.4 :

$$\Delta_x(\gamma) = \frac{\sum_{i=1}^{n(\gamma)} \delta_{x,y_i}(\gamma)}{n(\gamma)} . \quad (6.4)$$

La formule 6.4 permet de calculer $\Delta_x(\gamma)$ de telle sorte que sa valeur soit représentative de l'ensemble des influences des agents y_i sur l'importance pour x du but γ . En effet :

- La valeur de $\Delta_x(\gamma)$ sera négative si x pense que, en moyenne, les $n(\gamma)$ autres agents y_i ayant γ dans leurs buts accordent moins d'importance que lui à γ .
- La valeur de $\Delta_x(\gamma)$ sera positive si (i) x pense que, dans l'ensemble, les $n(\gamma)$ autres agents y_i ayant γ dans leurs buts accordent plus d'importance que lui au but γ et si (ii) x apprécie ces autres agents.
- La valeur $\Delta_x(\gamma)$ sera proche de la valeur nulle si x pense que les $n(\gamma)$ autres agents y_i ayant γ dans leurs buts accordent chacun autant d'importance au but γ que lui-même, ou si x pense que dans l'ensemble ces agents accordent plus d'importance que lui au but γ mais x n'apprécie pas ces autres agents.
- Enfin, $|\Delta_x(\gamma)|$ sera proportionnelle à $\alpha_{i,fl}$.

Encore une fois, la motivation de x sur ses buts personnels ne varie pas, et la valeur de $\Delta_x(\gamma)$ sur ces buts sera nulle.

Par ailleurs, ce calcul permet à l'importance accordée à leurs buts par les agents de converger : si un agent x n'est pas très motivé par un but γ alors que d'autres agents qu'il apprécie le sont, l'importance accordée par x à ce but augmentera fortement alors que l'importance accordée par les autres agents à ce but diminuera légèrement.

Exemple. Ici nous choisissons une valeur de $\alpha_{i,fl}$ de 0.5 car le scénario présenté ne correspond pas à une situation d'urgence. Nous considérons que chaque agent est donc relativement influençable par les autres.

L'agent Anna calcule donc l'influence $\delta_{a,y_i}(\gamma)$ que chaque agent y_i a sur chacun des buts γ de l'ensemble $\Gamma_{y_i}^a$. Les valeurs de $\delta_{a,y_i}(\gamma)$ calculées par Anna sont données dans le tableau 6.4. Nous détaillons le calcul pour l'agent Béa :

- Anna pense que Béa est indifférente au but *monter les meubles* γ_m ($\gamma_m \notin \Gamma_b^a$). Anna ne calcule donc pas de valeur de $\delta_{a,b}$ pour ce but.

- Anna pense que Béa accorde une importance de 0.5 au but d'*installer l'équipement informatique* alors qu'elle-même y accorde initialement une importance de 0.75. On a donc $iImp_b^a(\gamma_e) < iImp_a(\gamma_e)$. La valeur de $\delta_{a,b}(\gamma_e)$ est donc calculée de la manière suivante :

$$\delta_{a,b}(\gamma_e) = \alpha_{ifl} \times (iImp_b^a(\gamma_e) - iImp_a(\gamma_e)) = 0.5 \times (0.5 - 0.75) = -0.125 \quad .$$

- Anna pense que Béa accorde également une importance de 0.5 à son but de *rédiger un papier* alors qu'elle-même n'accorde aucune importance à ce but puisqu'il ne fait pas partie de son ensemble de buts Γ_a . On est donc dans le cas où $iImp_b^a(\gamma_p) > iImp_a(\gamma_p)$. La valeur de $\delta_{a,b}(\gamma_p)$ est donc calculée de la manière suivante :

$$\begin{aligned} \delta_{a,b}(\gamma_p) &= 2\alpha_{ifl} \times \max(0; b_{a,b} - 0.5) \times (iImp_b^a(\gamma_p) - iImp_a(\gamma_p)) \\ &= 2 \times 0.5 \times \max(0; 0.75 - 0.5) \times (0.5 - 0) = 0.125 \quad . \end{aligned}$$

TABLE 6.4 – *S1* – Influence de chacun des autres agents sur l'importance des buts pour Anna. La valeur $\delta_{a,y_i}(\gamma)$ représente l'influence de l'agent y_i sur l'importance accordée au but γ par Anna. La valeur $\Delta_a(\gamma)$ représente l'influence de l'ensemble des agents sur l'importance accordée au but γ par Anna.

	<i>monter les meubles</i>	<i>installer info.</i>	<i>formaliser le modèle</i>	<i>écrire un papier</i>
Bea	—	$\delta_{a,b}(\gamma_e) = -0.125$	—	$\delta_{a,b}(\gamma_p) = 0.125$
Cara	$\delta_{a,c}(\gamma_m) = -0.125$	$\delta_{a,c}(\gamma_e) = -0.125$	—	—
Dora	$\delta_{a,d}(\gamma_m) = 0.0$	$\delta_{a,d}(\gamma_e) = 0.0$	—	—
	$\Delta_a(\gamma_m) \simeq -0.062$	$\Delta_a(\gamma_e) \simeq -0.083$	$\Delta_a(\gamma_f) = 0.0$	$\Delta_a(\gamma_p) = 0.125$

Nous donnons également ici l'exemple du calcul de Δ_a pour le but *installer l'équipement informatique* partagé par Béa, Cara et Dora : on a donc $n(\gamma_e) = 3$. D'après les valeurs de δ_{a,y_i} listées dans le tableau 6.4 et d'après la formule 6.4, on calcule $\Delta_a(\gamma_e)$ de la façon suivante :

$$\Delta_a(\gamma_e) = (\delta_{a,b}(\gamma_e) + \delta_{a,c}(\gamma_e) + \delta_{a,d}(\gamma_e)) / n(\gamma_e) = (-0.125 - 0.125 + 0) / 3 \simeq -0.083 \quad .$$

Les valeurs de Δ_a pour les buts de *monter les meubles* et d'*écrire un papier* sont listées dans le tableau 6.4. Aucun autre agent n'ayant le but de *formaliser le modèle*, aucune valeur d'influence des autres sur ce but n'est calculée.

D'après le tableau 6.4, Béa, Cara et Dora ont une influence légèrement négative sur l'importance accordée par Anna à *installer l'équipement informatique* : cela vient du fait que Béa et Cara sont moins motivées par ce but qu'Anna et Dora. L'importance accordée par Béa à son but *écrire un papier* selon Anna étant élevée et Anna appréciant Béa, l'importance accordée par Anna à ce but, qui était nulle, augmentera légèrement.

Calcul de l'importance finale des buts pour x

Finalement la valeur $\Delta_x(\gamma)$ est utilisée par l'agent x pour mettre à jour l'importance qu'il accorde au but γ . L'importance finale $fImp_x(\gamma)$ que x accorde au but γ est donnée par la

formule 6.5 :

$$\left\{ \begin{array}{ll} fImp_x(\gamma) = \max(0 ; iImp_x(\gamma) + \Delta_x(\gamma)) & \text{si } \Delta_x(\gamma) \leq 0 \text{ et } \gamma \notin \Gamma_{x,self} \\ fImp_x(\gamma) = \min(1 ; iImp_x(\gamma) + \Delta_x(\gamma)) & \text{si } \Delta_x(\gamma) > 0 \text{ et } \gamma \notin \Gamma_{x,self} \\ fImp_x(\gamma) = iImp_x(\gamma) & \text{sinon} \end{array} \right. \quad . \quad (6.5)$$

Avec la formule 6.5, l'importance que x accorde à ses propres buts est modulée par l'importance que les autres agents accordent à ces mêmes buts selon x , et la nouvelle valeur d'importance calculée reste dans l'intervalle $[0;1]$.

L'agent x obtient donc un classement de ses buts selon leur importance :

- Plusieurs buts, personnels ou collectifs, peuvent avoir la même importance pour x .
- Initialement, tous les buts collectifs de x avaient la même importance, et tous les buts personnels de x avaient également la même importance. À l'issue de l'étape de prise en compte des autres agents, cela n'est plus nécessairement le cas, comme on peut le constater dans l'exemple ci-dessous.

Exemple. Anna calcule l'importance finale des buts en appliquant la formule 6.5. Par exemple pour le but *installer l'équipement informatique* (γ_e), Anna calcule :

$$fImp_a(\gamma_e) \simeq \max(0 ; iImp_a(\gamma_e) + \Delta_a(\gamma_e)) \simeq \max(0 ; 0.75 - 0.083) \simeq 0.666 \quad .$$

Toutes les valeurs d'importance finale des buts pour Anna peuvent être trouvées dans le tableau 6.6.

En se rapportant aux valeurs du tableau 6.6, on peut constater que la motivation d'Anna pour le but *installer l'équipement informatique* a légèrement diminué⁶. Anna a par ailleurs adopté le but de Béa d'*écrire un papier*, même si elle y accorde une importance assez faible. On peut également remarquer que les motivations des agents convergent vers une valeur moyenne : par exemple pour le but *installer l'équipement informatique*, les motivations d'Anna et Dora d'une part et de Béa et Cara d'autre part convergent ; les motivations d'Anna et Dora diminuant alors que celles de Béa et Cara augmentent.

Calcul de l'importance finale des buts pour les autres selon x

L'agent x pense également que chaque autre agent y_i est aussi influencé par le groupe de la même manière que lui-même. Pour chaque agent y_i , x cherche donc à calculer $fImp_{y_i}^x(\gamma)$, l'importance selon lui du but γ pour y_i en tenant compte de l'influence de chaque autre agent y_j (x compris) sur y_i .

Encore une fois, x applique le même raisonnement que pour lui-même pour calculer δ_{y_i,y_j}^x . Néanmoins, x ne sachant pas ce que pense y_i à propos de y_j ⁷, x utilise ses propres croyances sur y_j . La valeur de δ_{y_i,y_j}^x est donnée par la formule 6.6 :

6. Les croyances d'Anna sur les autres agents correspondant à la réalité, ce qu'Anna pense être la motivation finale des autres agents pour leurs buts correspond effectivement à la motivation finale des autres agents pour leurs buts.

7. L'agent x a une croyance sur la bienveillance de y_i envers y_j mais n'a pas de croyance sur les croyances de y_i sur y_j .

$$\left\{ \begin{array}{ll} \delta_{y_i, y_j}^x(\gamma) = \alpha_{ifl} \times (iImp_{y_j}^x(\gamma) - iImp_{y_i}^x(\gamma)) & \text{si } iImp_{y_j}^x(\gamma) < iImp_{y_i}^x(\gamma) \\ \delta_{y_i, y_j}^x(\gamma) = 2\alpha_{ifl} \times \max(0; b_{y_i, y_j}^x - 0.5) & \text{si } iImp_{y_j}^x(\gamma) \geq iImp_{y_i}^x(\gamma) \\ & \times (iImp_{y_j}^x(\gamma) - iImp_{y_i}^x(\gamma)) \end{array} \right. \quad (6.6)$$

où, lorsque $y_j = x$, $iImp_{y_j}^x(\gamma) = iImp_x^x(\gamma) = iImp_x(\gamma)$.

En effet, $iImp_{y_j}^x(\gamma) = v$ signifie que x pense que l'importance du but γ pour y_j est v . La notation $iImp_x^x(\gamma) = v$ signifie alors que x pense que l'importance du but γ pour x est v . Or si x pense que l'importance qu'il accorde à son but γ est v , alors l'importance qu'il accorde à son but γ est v . Autrement dit, $iImp_x^x(\gamma) = v \Rightarrow iImp_x(\gamma) = v$: les agents sont capables d'introspection positive sur leurs croyances (Rao and Georgeff, 1991).

L'agent x calcule ensuite la valeur de $\Delta_{y_i}^x(\gamma)$ qui représente l'influence globale de tous les autres agents de l'environnement sur y_i selon x . La valeur de $\Delta_{y_i}^x(\gamma)$ est donnée par la formule 6.7 :

$$\Delta_{y_i}^x(\gamma) = \frac{\sum_{j=1}^{n(\gamma)} \delta_{y_i, y_j}^x(\gamma)}{n(\gamma)} \quad (6.7)$$

où $n(\gamma)$ représente le nombre d'agents y_j ayant γ dans leur ensemble de buts selon x .

De la même manière que pour lui-même, x ne calcule pas de valeur de $\delta_{y_i, y_j}^x(\gamma)$ et de $\Delta_{y_i}^x(\gamma)$ pour les buts personnels de y_i : x pense que la motivation de y_i sur ses buts personnels n'est pas influencée par les autres.

Puis x calcule l'importance finale $fImp_{y_i}^x(\gamma)$ qu'il pense que chaque agent y_i accorde au but γ encore une fois en utilisant le même procédé que pour lui-même, comme dans la formule 6.8 :

$$\left\{ \begin{array}{ll} fImp_{y_i}^x(\gamma) = \max(0; iImp_{y_i}^x(\gamma) + \Delta_{y_i}^x(\gamma)) & \text{si } \Delta_{y_i}^x(\gamma) \leq 0 \text{ et } \gamma \notin \Gamma_{y_i, self}^x \\ fImp_{y_i}^x(\gamma) = \min(1; iImp_{y_i}^x(\gamma) + \Delta_{y_i}^x(\gamma)) & \text{si } \Delta_{y_i}^x(\gamma) > 0 \text{ et } \gamma \notin \Gamma_{y_i, self}^x \\ fImp_{y_i}^x(\gamma) = iImp_{y_i}^x(\gamma) & \text{sinon} \end{array} \right. \quad (6.8)$$

Exemple. Nous nous servons encore une fois de l'exemple présenté en début de chapitre. Ici, Anna doit calculer ce qu'elle pense être l'importance finale des buts pour Béa. Pour cela, elle doit tenir compte de sa propre influence sur Béa, mais aussi de l'influence de Cara et Dora. Nous développons l'exemple du calcul de ce qu'Anna pense être son influence sur Béa pour le but *monter les meubles*.

Anna accorde une importance de $iImp_a^a(\gamma_m) = iImp_a(\gamma_m) = 0.75$ au but de *monter les meubles*, et elle pense que Béa y accorde une importance de $iImp_b^a(\gamma_m) = 0$. On a donc $iImp_b^a(\gamma_m) < iImp_a^a(\gamma_m)$. Anna calcule alors $\delta_{b,a}^a(\gamma_m)$ de la manière suivante, comme dans la formule 6.6 :

$$\begin{aligned} \delta_{b,a}^a(\gamma_m) &= 2\alpha_{ifl} \times \max(0; b_{b,a}^a - 0.5) \times (iImp_a^a(\gamma_m) - iImp_b^a(\gamma_m)) \\ &= 2 \times 0.5 \times (0.75 - 0.5) \times (0.75 - 0) \simeq 0.188 \quad . \end{aligned}$$

Nous ne détaillons pas le calcul des autres valeurs de $\delta_{b,y_i}^a(\gamma)$ qui sont présentées dans le tableau 6.5.

TABLE 6.5 – *S1* – Ce qu’Anna pense être l’influence des autres agents sur l’importance des buts pour Béa, Cara et Dora. La valeur $\delta_{y_i, y_j}^a(\gamma)$ représente l’influence de l’agent y_j sur l’importance accordée au but γ par y_i selon Anna. $\Delta_{y_i}^a(\gamma)$ représente l’influence de l’ensemble des agents sur l’importance accordée au but γ par y_i selon Anna.

	<i>monter les meubles</i>	<i>installer info.</i>	<i>formaliser le modèle</i>	<i>écrire un papier</i>
Pour Béa				
Anna	$\delta_{b,a}^a(\gamma_m) \simeq 0.188$	$\delta_{b,a}^a(\gamma_e) \simeq 0.062$	$\delta_{b,a}^a(\gamma_f) \simeq 0.062$	—
Cara	$\delta_{b,c}^a(\gamma_m) = 0.125$	$\delta_{b,c}^a(\gamma_e) = 0.0$	—	—
Dora	$\delta_{b,d}^a(\gamma_m) \simeq 0.188$	$\delta_{b,d}^a(\gamma_e) \simeq 0.062$	—	—
	$\Delta_b^a(\gamma_m) \simeq 0.167$	$\Delta_b^a(\gamma_e) \simeq 0.042$	$\Delta_b^a(\gamma_f) \simeq 0.062$	$\Delta_b^a(\gamma_p) = 0.0$
Pour Cara				
Anna	$\delta_{c,a}^a(\gamma_m) \simeq 0.062$	$\delta_{c,a}^a(\gamma_e) \simeq 0.062$	$\delta_{c,a}^a(\gamma_f) \simeq 0.062$	—
Bea	—	$\delta_{c,b}^a(\gamma_e) = 0.0$	—	$\delta_{c,b}^a(\gamma_p) = 0.125$
Dora	$\delta_{c,d}^a(\gamma_m) \simeq 0.062$	$\delta_{c,d}^a(\gamma_e) \simeq 0.062$	—	—
	$\Delta_c^a(\gamma_m) \simeq 0.062$	$\Delta_c^a(\gamma_e) \simeq 0.042$	$\Delta_c^a(\gamma_f) \simeq 0.062$	$\Delta_c^a(\gamma_p) = 0.125$
Pour Dora				
Anna	$\delta_{d,a}^a(\gamma_m) = 0.0$	$\delta_{d,a}^a(\gamma_e) = 0.0$	$\delta_{d,a}^a(\gamma_f) \simeq 0.062$	—
Bea	—	$\delta_{d,b}^a(\gamma_e) = -0.125$	—	$\delta_{d,b}^a(\gamma_p) = 0.125$
Cara	$\delta_{d,c}^a(\gamma_m) = -0.125$	$\delta_{d,c}^a(\gamma_e) = -0.125$	—	—
	$\Delta_d^a(\gamma_m) \simeq -0.062$	$\Delta_d^a(\gamma_e) \simeq -0.083$	$\Delta_d^a(\gamma_f) \simeq 0.062$	$\Delta_d^a(\gamma_p) = 0.125$

Anna calcule ensuite $\Delta_b^a(\gamma)$, c’est-à-dire ce qu’elle pense être l’influence globale de Cara, Dora et elle-même sur l’importance accordée par Béa au but γ . Ces valeurs sont présentées dans le tableau 6.5. Nous donnons ici l’exemple du calcul de $\Delta_b^a(\gamma_f)$.

Anna étant la seule à avoir le but de *formaliser le modèle*, on a $n(\gamma_f) = 1$, ce qui donne :

$$\Delta_b^a(\gamma_f) = \delta_{b,a}^a(\gamma_f)/1 \simeq 0.062 \quad .$$

L’importance finale du but *formaliser le modèle* pour Béa selon Anna est donc calculée de la façon suivante :

$$fImp_b^a(\gamma_f) = \min(1; iImp_b^a(\gamma_f) + \Delta_b^a(\gamma_f)) \simeq \min(0; 0 + 0.062) \simeq 0.062 \quad .$$

Enfin Anna procède au même calcul pour les agents Cara et Dora. Nous ne détaillons pas ces calculs ici, mais nous résumons dans le tableau 6.6 les valeurs finales d’importance des buts pour les autres agents selon Anna.

6.4 Utilité des tâches

Une fois l’importance des buts calculée, l’agent x doit calculer l’utilité des tâches. Du point de vue de l’agent x , il existe deux possibilités pour que la tâche τ_k soit réalisée : x la fait lui-même ou laisse un autre agent la faire. L’agent x doit donc calculer l’utilité de faire la tâche τ_k , mais il doit également calculer l’utilité pour les autres de faire chaque tâche. Ainsi, x pourra faire un

TABLE 6.6 – $S1$ – Valeurs d’importance finale accordée par Anna, Béa, Cara et Dora aux buts monter les meubles, installer l’équipement informatique, écrire un papier et formaliser le modèle selon Anna.

	<i>monter les meubles</i>	<i>installer info.</i>	<i>formaliser le modèle</i>	<i>écrire un papier</i>
Anna	$fImp_a(\gamma_m) \simeq 0.688$	$fImp_a(\gamma_e) \simeq 0.667$	$fImp_a(\gamma_f) = 0.25$	$fImp_a(\gamma_p) = 0.125$
Bea	$fImp_b^a(\gamma_m) \simeq 0.167$	$fImp_b^a(\gamma_e) \simeq 0.542$	$fImp_b^a(\gamma_f) \simeq 0.062$	$fImp_b^a(\gamma_p) = 0.5$
Cara	$fImp_c^a(\gamma_m) \simeq 0.562$	$fImp_c^a(\gamma_e) \simeq 0.542$	$fImp_c^a(\gamma_f) \simeq 0.062$	$fImp_c^a(\gamma_p) = 0.125$
Dora	$fImp_d^a(\gamma_m) \simeq 0.688$	$fImp_d^a(\gamma_e) \simeq 0.667$	$fImp_d^a(\gamma_f) \simeq 0.062$	$fImp_d^a(\gamma_p) = 0.125$

choix informé quant à la tâche à sélectionner. Les processus de calcul pour ces deux cas sont développés dans les sections suivantes.

6.4.1 Utilité pour x de faire la tâche lui-même

L’agent x doit maintenant chercher la tâche la plus judicieuse à réaliser en fonction de l’importance qu’il accorde à ses buts : pour cela x raisonne sur les arbres de tâches décrits en ACTIVITY-DL.

Si les contraintes de faisabilité collectives sont bloquantes, l’agent x doit tout d’abord réfléchir aux tâches qui sont réalisables par l’équipe : il n’est pas pertinent pour x de calculer l’utilité des tâches qui ne sont pas réalisables.

L’agent x calcule alors l’utilité de chaque tâche, en commençant par les tâches racines des arbres de tâches. Deux facteurs sont à prendre en compte pour le calcul de l’utilité d’une tâche :

- L’utilité d’une tâche pour x est proportionnelle à l’importance finale du but auquel elle contribue. Si une tâche ne contribue à aucun but, alors son utilité est nulle.
- D’autre part, x doit prendre en compte ses compétences : on veut que l’agent x privilégie les tâches pour lesquelles il est le plus compétent. L’agent x raisonnant en priorité sur des tâches composées auxquelles peuvent être attachées plusieurs compétences, il faut alors qu’il calcule son niveau de compétence général sur une tâche composée.

Nous expliquons dans les sections suivantes le processus de calcul de l’utilité d’une tâche.

Exemple. Le scénario ne représentant pas une situation d’urgence, nous considérons ici les contraintes de faisabilité collective comme bloquantes. Cependant, ces contraintes sont vérifiées pour chaque tâche : les agents calculeront donc l’utilité de toutes les tâches ($\tau_m, \tau_e, \tau_p, \tau_f$).

Niveau de compétence général de x sur une tâche quelconque

Nous rappelons que les compétences sont spécifiées par les experts du domaine ou par les ergonomes sur les tâches feuilles (i.e. actions) de l’arbre de tâches uniquement. Cependant, comme présenté dans le chapitre 5, ces compétences sont propagées à l’ensemble de l’arbre de tâches sous forme de contraintes de faisabilité collective (voir 5.4) ou sous forme d’informations représentatives de l’avancement des agents sur l’activité collective (voir 5.5).

Ici nous cherchons à calculer, pour l’agent x , l’utilité des tâches afin d’en sélectionner une. Pour cela, l’agent x peut raisonner sur les informations se rapportant aux compétences attachées

à chaque tâche. En effet, la disjonction de compétences $c(\tau_k, t)$ représente les compétences que l'agent x doit avoir pour faire la tâche τ_k à l'instant t . L'agent x peut donc calculer son niveau de compétence général pour τ_k en raisonnant sur $c(\tau_k, t)$.

Nous rappelons qu'un tableau récapitulatif des notations utilisées dans ce chapitre et dans le chapitre précédent ainsi que de leur signification est présenté en annexe C.1.

Valeur par défaut pour les tâches ne nécessitant pas de compétence

La disjonction $c(\tau_k, t)$ peut contenir des éléments *TRUE* qui correspondent à une absence de compétence requise ; on a alors $a_{x, TRUE} = \alpha_{true}$, où $\alpha_{true} \in]0; 1]$ est une valeur par défaut. La valeur exacte de α_{true} dépend de ce que l'on souhaite obtenir :

- Si l'on veut que l'agent x privilégie les tâches sur lesquelles aucune compétence n'est spécifiée, on choisira $\alpha_{true} = 1$.
- Si à l'inverse on souhaite que l'agent x privilégie les tâches sur lesquelles des compétences sont spécifiées, on choisira une valeur proche de la valeur nulle ; à noter que dans ce cas, l'agent ne privilégiera les tâches pour lesquelles des compétences κ sont spécifiées que si son niveau de compétence est tel que $a_{x, \kappa} > 0$.
- Enfin, si l'on souhaite que l'agent x privilégie les tâches nécessitant la compétence κ pour lesquelles il est doué (i.e. $a_{x, \kappa} > 0.5$) par rapport aux tâches ne nécessitant aucune compétence, il faudra avoir $\alpha_{true} = 0.5$.

Le choix de la valeur de α_{true} dépendra généralement d'autres éléments : si l'on considère qu'un agent qui n'a pas la compétence κ attachée à la tâche τ_k ne peut pas faire la tâche τ_k , alors il peut être intéressant que les agents qui ont cette compétence privilégient cette tâche (i.e. $\alpha_{true} = 0$). Dans notre exemple, Cara ne sait pas du tout bricoler : même si Anna est plus douée qu'elle pour l'informatique, il serait opportun de laisser à Cara le soin d'*installer l'équipement informatique* puisqu'Anna peut *monter les meubles*.

Enfin, indépendamment de la valeur de α_{true} , si, toutes choses égales par ailleurs, τ_1 et τ_2 nécessitent respectivement les compétences κ_1 et κ_2 et si x a un meilleur niveau de compétence sur κ_1 que sur κ_2 , x privilégiera toujours τ_1 par rapport à τ_2 (voir paragraphe suivant).

Exemple. Ce scénario ne représentant pas une situation d'urgence, nous assignons ici à α_{true} la valeur 0.5 de manière à ce que les agents privilégient les tâches pour lesquelles ils sont doués.

Calcul du niveau de compétence général de x sur une tâche quelconque

On exprime par $a_x(\tau_k) \in [0; 1]$ le niveau de compétence général de x sur la tâche τ_k . La valeur de $a_x(\tau_k)$ est calculée avec la formule 6.9 :

$$a_x(\tau_k) = \max_{\forall \kappa \text{ de } c(\tau_k, t)} (a_{x, \kappa}) \quad . \quad (6.9)$$

La formule 6.9 est définie de telle façon que :

- Lorsque τ_k est une tâche feuille, $c(\tau_k, t)$ ne peut contenir qu'une seule compétence, et $a_x(\tau_k)$ prend alors la valeur du niveau de x pour cette compétence.
- Lorsque τ_k est une tâche composée, $c(\tau_k, t)$ est une disjonction de compétences, et $a_x(\tau_k)$ prend alors la valeur maximale de tous les niveaux de compétence dont la compétence est spécifiée dans $c(\tau_k, t)$.

Cela permettra à x de privilégier les tâches pour lesquelles il est compétent ; c'est-à-dire, dans le cas d'une tâche composée, de privilégier la tâche composée qui contient la sous-tâche pour laquelle x est le plus compétent ⁸.

Exemple. Comme on peut le voir sur la figure 6.1, l'information dynamique relative aux compétences nécessaires pour qu'un agent puisse participer à la tâche *monter les meubles* à t_0 est $c(\tau_m, t_0) = \text{BM} \vee \text{BT}$. Si l'on prend l'exemple de l'agent Anna, la valeur de compétence générale d'Anna sur la tâche *monter les meubles* est calculée comme dans la formule 6.9 :

$$a_a(\tau_m) = \max(a_{a,\text{BM}}; a_{a,\text{BT}}) = \max(0.5; 0.25) = 0.5 \quad .$$

La tâche *installer l'équipement informatique* (τ_e) demande la compétence *informatique* notée I, et on a donc $c(\tau_e, t_0) = \text{I}$. La valeur de compétence général d'Anna sur cette tâche est donc

$$a_a(\tau_e) = a_{a,\text{I}} = 0.25$$

Les autres tâches ne demandant pas de compétences, on a $c(\tau_p, t_0) = c(\tau_f, t_0) = \text{TRUE}$ pour *écrire un papier* et *formaliser le modèle*. Les valeurs de compétences d'Anna sur ces tâches sont donc

$$a_a(\tau_f) = a_a(\tau_p) = a_{a,\text{TRUE}} = 0.5 \quad .$$

Calcul de l'utilité d'une tâche

On note $U_x(\tau_k) \in [0; 1]$ l'utilité de la tâche τ_k pour l'agent x . Cette utilité est donnée par la formule 6.10 dans le cas où τ_k contribue ⁹ au but γ :

$$U_x(\tau_k) = \text{fImp}_x(\gamma) \times a_x(\tau_k) \quad . \quad (6.10)$$

L'utilité de τ_k pour x est donnée par la formule 6.11 dans le cas où τ_k ne contribue à aucun but :

$$U_x(\tau_k) = 0 \quad . \quad (6.11)$$

Exemple. On peut calculer l'utilité de la tâche *monter les meubles* (τ_m) pour l'agent Anna avec la formule 6.10 en utilisant la valeur d'importance du but *monter les meubles* (γ_m) auquel τ_m contribue :

$$U_a(\tau_m) = \text{fImp}_a(\gamma_m) \times a_a(\tau_m) \simeq 0.688 \times 0.5 \simeq 0.344 \quad .$$

Les utilités pour Anna de faire les tâches τ sont données dans le tableau 6.7.

8. Si l'on avait pris la valeur minimale de tous les niveaux de compétence dont la compétence est spécifiée dans $c(\tau_k, t)$, x n'aurait pas pu sélectionner la tâche τ_k dès lors qu'il n'a pas la compétence nécessaire pour faire l'une des sous-tâches de τ_k , ce qui n'a pas de sens. De même, si l'on avait pris une valeur moyenne, x pourrait éviter de sélectionner une tâche composée τ_k contenant une sous-tâche τ_{kk} pour laquelle il est parfaitement compétent uniquement s'il n'est pas ou s'il est peu compétent pour d'autres des sous-tâches de τ_k .

9. Tout comme Lochbaum et al. (1990), nous considérons qu'une tâche contribue à un but si elle fait partie des tâches à réaliser pour atteindre ce but.

6.4.2 Utilité des tâches pour les autres selon x

Calculer l'utilité qu'il pense qu'a chacune des tâches pour les autres agents permettra à un agent x d'imaginer ce que les autres agents voudront faire (i.e. l'utilité des tâches pour les autres selon x). De la sorte, l'agent x pourra adopter des comportements de *délégation faible* (voir 4.1.2) : par exemple si Anna pense que Cara et Dora vont réaliser la tâche *installer l'équipement informatique* qui contribue à réaliser leur but collectif γ_e , il est dans l'intérêt d'Anna de les laisser faire pour se consacrer à autre chose.

Cas des agents qui attendent

Nous avons vu dans la section 6.2 que, lors de la phase de *perception* qui précède la phase de *décision*, l'agent x sait ce que les autres agents sont en train de faire. Plus spécifiquement, un autre agent y_i peut être en train de (i) faire une action quelconque ou (ii) attendre pour faire une action¹⁰.

Pour le cas (i), l'agent x calculera tout de même ce qu'il pense que l'agent y_i voudra faire dès lors qu'il aura terminé son action en cours¹¹ ; x cherche à déterminer la *prochaine* action que voudrait faire y_i .

En revanche, nous considérons pour le cas (ii) que la prochaine action que voudrait faire y_i est simplement celle pour laquelle il attend. Il n'y a alors pas besoin de calculer l'utilité pour y_i de chaque tâche τ_k selon x , mais simplement de calculer l'utilité selon x de la tâche τ_k correspondant à l'action pour laquelle y_i attend.

Calcul de l'utilité des tâches pour les autres

Pour chaque agent y_i , l'agent x calcule, de la même manière que pour lui-même, une valeur $a_{y_i}^x(\tau_k) \in [0;1]$ qu'il pense être la compétence générale de y_i sur la tâche τ_k . Pour cela, x raisonne sur $a_{y_i,\kappa}^x$ ce qu'il pense être les niveaux de compétence de y_i sur les compétences κ précisées dans $c(\tau_k, t)$. La formule 6.12 donne la valeur de $a_{y_i}^x(\tau_k)$:

$$a_{y_i}^x(\tau_k) = \max_{\forall \kappa \text{ de } c(\tau_k, t)} (a_{y_i, \kappa}^x) \quad . \quad (6.12)$$

L'agent x peut ensuite calculer ce qu'il pense être l'utilité de la tâche τ_k pour l'agent y_i . On note $U_{y_i}^x(\tau_k) \in [0;1]$ cette utilité, et elle est donnée par l'équation 6.13 si la tâche τ_k contribue selon x à l'un des buts γ de y_i (i.e. $\gamma \in \Gamma_{y_i}^x$) :

$$U_{y_i}^x(\tau_k) = \text{Imp}_{y_i}^x(\gamma) \times a_{y_i}^x(\tau_k) \quad . \quad (6.13)$$

Si, selon x , la tâche τ_k ne contribue à aucun but de y_i , la valeur de $U_{y_i}^x(\tau_k)$ est donnée par la formule 6.14 :

$$U_{y_i}^x(\tau_k) = 0 \quad . \quad (6.14)$$

Cette utilité $U_{y_i}^x(\tau_k)$ peut être vue comme la probabilité, selon x , que y_i fasse la tâche τ_k . En effet, plus cette valeur est élevée, plus l'agent y_i a d'intérêt à faire la tâche τ_k (selon x).

10. On rappelle que l'on peut avoir $n(\tau_k)$ agents qui attendent pour faire l'action τ_k , avec $1 \leq n(\tau_k) < n_{\min}(\tau_k)$ si la contrainte de faisabilité collective en termes de nombre de personnes est bloquante.

11. Nous ignorons dans cette thèse les aspects liés à la durée des actions. Autrement dit, l'agent x imagine ce que voudra faire l'agent y_i car par défaut nous considérons que l'action que fait l'agent y_i ne va pas durer éternellement.

Exemple. À t_0 , aucun agent n'est en train d'attendre pour faire une tâche nécessitant plusieurs personnes. Anna calcule donc pour Béa, Cara et Dora ce qu'elle pense être l'utilité pour elles de faire la tâche τ . Nous prenons ici l'exemple de Béa. Béa n'ayant pas la compétence *bricolage* (Béa n'a ni la compétence BM ni la compétence BT), elle ne peut pas participer à la tâche *monter les meubles*. Anna ne calcule donc pas de valeur d'utilité de cette tâche pour Béa.

Pour la tâche *installer l'équipement informatique* τ_e , Anna calcule dans un premier temps ce qu'elle pense être la valeur de compétence globale de Béa avec la formule 6.12 :

$$a_b^a(\tau_e) = a_{b,I}^a = 0.5 \quad .$$

Dans un second temps, Anna calcule l'utilité, selon elle, pour Béa d'*installer l'équipement informatique* avec la formule 6.13 :

$$U_b^a(\tau_e) = fImp_b^a(\gamma_e) \times a_b^a(\tau_e) \simeq 0.542 \times 0.5 \simeq 0.271 \quad .$$

Anna calcule de la même manière l'utilité des tâches *formaliser le modèle* et *écrire un papier* pour Béa, puis elle fait la même chose pour mes agents Cara et Dora. Les valeurs d'utilité pour toutes les tâches et pour chacun des autres agents calculées par Anna sont données dans le tableau 6.7.

TABLE 6.7 – *S1* – Valeurs d'utilité pour toutes les tâches et pour tous les agents calculées par Anna : $U_a(\tau)$ représente l'utilité pour Anna de la tâche τ et $U_{y_i}^a(\tau)$ représente l'utilité pour y_i de faire la tâche τ selon Anna.

	<i>monter les meubles</i>	<i>installer info.</i>	<i>formaliser le modèle</i>	<i>écrire un papier</i>
Anna	$U_a(\tau_m) \simeq 0.344$	$U_a(\tau_e) \simeq 0.167$	$U_a(\tau_f) = 0.125$	$U_a(\tau_p) \simeq 0.062$
Bea	—	$U_b^a(\tau_e) \simeq 0.271$	$U_b^a(\tau_f) \simeq 0.031$	$U_b^a(\tau_p) = 0.25$
Cara	—	$U_c^a(\tau_e) \simeq 0.271$	$U_c^a(\tau_f) \simeq 0.031$	$U_c^a(\tau_p) \simeq 0.062$
Dora	$U_d^a(\tau_m) \simeq 0.344$	$U_d^a(\tau_e) \simeq 0.167$	$U_d^a(\tau_f) \simeq 0.031$	$U_d^a(\tau_p) \simeq 0.062$

6.5 Sélection d'une tâche

L'agent x ayant calculé l'utilité pour lui de faire chaque tâche ainsi que ce qu'il pense être l'utilité pour les autres agents de faire chaque tâche, l'agent x doit maintenant sélectionner une tâche. Pour cela, x imagine différentes répartitions possibles de tâches dans le but de choisir la meilleure.

Nous détaillons ce processus dans les sections suivantes.

6.5.1 Génération des répartitions de tâches

L'agent x peut maintenant générer des répartitions possibles de tâches entre tous les agents. Nous expliquons dans un premier temps quelle est la forme des répartitions de tâches, puis nous développons le processus de génération des répartitions de tâches.

Forme des répartitions de tâches

Si l'on note $(y_i, \tau_{k_{y_i,j}})$ le fait que l'agent y_i soit assigné à la tâche $\tau_{k_{y_i,j}}$ dans la répartition j , alors chaque répartition j générée par x sera de la forme 6.15 :

$$R_{x,j} = \left\{ (x, \tau_{k_{x,j}}), (y_1, \tau_{k_{y_1,j}}), \dots, (y_{n-1}, \tau_{k_{y_{n-1},j}}) \right\} \quad (6.15)$$

où n représente le nombre d'agents sur lesquels x raisonne (voir 6.6).

La génération de répartitions de tâches respecte cependant une contrainte : puisque nous travaillons sur des activités collectives, certaines tâches peuvent être collectives. Chaque tâche τ_k nécessite donc entre $n_{min}(\tau_k)$ et $n_{max}(\tau_k, t)$ agents pour être réalisée ; $n_{min}(\tau_k)$ et $n_{max}(\tau_k, t)$ étant respectivement spécifiés dans la contrainte de faisabilité collective en termes de nombre de personnes (voir 5.4.1) et dans les informations en termes de nombre de personnes (voir 5.5.1) propagées à τ_k . Soit $n(\tau_k)$ le nombre d'agents assignés à la tâche τ_k dans la répartition $R_{x,j}$: on ne génère pas de répartitions telles que $n_{max}(\tau_k, t) < n(\tau_k)$. En effet, cela signifierait que certains agents assignés à τ_k ne pourraient finalement pas y participer.

Par ailleurs, un autre cas n'est pas idéal : si $n(\tau_k) < n_{min}(\tau_k)$, selon que le nombre d'agents minimum pour réaliser une tâche est bloquant ou non, la tâche τ_k risque d'avoir un résultat de qualité médiocre, d'échouer, voire de ne pas pouvoir être réalisée du tout. Ces répartitions non idéales ne doivent cependant pas être éliminées. En effet, prenons l'exemple suivant : les agents Xéna, Ynès et Zoé (désignés dans les formules par leur initiale) doivent réaliser les tâches τ_1 et τ_2 telles que $n_{min}(\tau_1) = 2$ et $n_{min}(\tau_2) = 2$. Si Ynès a déjà choisi la tâche τ_1 et Zoé a déjà choisi la tâche τ_2 (i.e. Ynès attend pour la tâche τ_1 et Zoé attend pour la tâche τ_2), alors l'agent Xéna ne gère que les répartitions dans lesquelles Ynès et Zoé sont assignées respectivement à τ_1 et τ_2 , soit :

$$R_{x,1} = \{(x, \tau_1), (y, \tau_1), (z, \tau_2)\} \quad R_{x,2} = \{(x, \tau_2), (y, \tau_1), (z, \tau_2)\} \quad .$$

Dans ce cas, ces répartitions correspondent au cas où $n(\tau_k) < n_{min}(\tau_k)$, et toutes les éliminer reviendrait pour Xéna à choisir de ne rien faire, ce qui mènerait les agents dans une situation de blocage.

Processus de génération des répartitions de tâches

Générer toutes les répartitions de tâches possibles entre tous les agents revient à générer un nombre factoriel de répartitions et devrait donc être évité autant que possible. Pour cela, nous proposons les mécanismes suivants de génération de répartitions de tâches :

Dans un premier temps l'agent x cherche, pour chaque agent, l'ensemble des tâches ayant la plus forte utilité. Soit $T_{Umax,x}$ l'ensemble des tâches ayant la plus forte utilité pour x , et T_{Umax,y_i}^x l'ensemble des tâches ayant la plus forte utilité pour y_i selon x . Ces ensembles sont respectivement donnés par les formules 6.16 et 6.17 :

$$T_{Umax,x} = \arg \max_{\tau \in T_{x,n}} (U_x(\tau)) \quad (6.16)$$

$$T_{Umax,y_i}^x = \arg \max_{\tau \in T_{x,n}} (U_{y_i}^x(\tau)) \quad (6.17)$$

où $T_{x,n}$ représente l'ensemble des tâches sur lesquelles x raisonne (voir 6.6).

Deux cas peuvent se présenter : (i) chacun de ces ensemble ne contient qu'une seule tâche ou (ii) au moins l'un de ces ensembles contient plusieurs tâches. Ces deux cas sont traités de manière différente :

- (i) Dans ce cas, une seule répartition de tâches est possible à partir de ces éléments. Si la répartition convient (i.e. elle vérifie, pour chacune des tâches τ_k de la répartition, $n(\tau_k) \leq n_{max}(\tau_k, t)$), c'est celle qui est sélectionnée par l'agent x . Si ce n'est pas le cas :
 - (a) x établit une liste des agents "bloquants" y_i étant assignés à des tâches telles que $n(\tau_k) > n_{max}(\tau_k, t)$, et
 - (b) pour ce sous-ensemble d'agents, x génère toutes les répartitions de tâches possibles et sélectionne la *meilleure* (voir 6.5.3).
 - (c) L'agent x remplace ensuite dans la répartition d'origine les tâches des agents bloquants par celles de la nouvelle répartition. Enfin,
 - (d) l'agent x vérifie que la nouvelle répartition est valide. Si c'est le cas, c'est celle qui est sélectionnée. Si ce n'est pas le cas, x recommence récursivement les étapes a¹², b, c et d.
- (ii) Dans ce cas, plusieurs répartitions de tâches peuvent être générées. L'agent x génère donc toutes les répartitions possibles à partir de ces éléments. Si une ou plusieurs des répartitions conviennent, x sélectionne la *meilleure* (voir 6.5.3). Dans le cas où aucune des répartitions ne convient, x génère l'ensemble de toutes les répartitions possibles à partir de tout l'ensemble des tâches.

6.5.2 Calcul de l'utilité des répartitions

L'utilité de chaque répartition j correspondra à la moyenne des utilités de chaque tâche de la répartition.

L'utilité de la répartition $R_{x,j}$ est notée $U_x(R_{x,j})$ et est calculée avec la formule 6.18 :

$$U_x(R_{x,j}) = \frac{U_x(\tau_{k_{x,j}}) + \sum_{i=1}^{n-1} U_{y_i}^x(\tau_{k_{y_i,j}})}{n} \quad (6.18)$$

où n est le nombre d'agents sur lesquels x raisonne (voir 6.6).

6.5.3 Sélection de la meilleure répartition

Après avoir calculé l'utilité de chaque combinaison, l'agent x doit en sélectionner une. On notera τ_k la tâche sélectionnée par x . Encore une fois sélectionner une répartition c'est pas trivial. En effet, s'il paraît naturel de sélectionner la répartition ayant l'utilité la plus élevée, plusieurs répartitions peuvent correspondre à ce critère. Nous proposons des mécanismes de choix décrits dans les paragraphes suivants.

12. À noter que si l'agent x répète plusieurs fois les étapes a , b , c et d , la liste d'agents bloquants n'est pas écrasée à chaque fois mais est conservée et agrandie. En effet, dans le cas contraire, on pourrait répéter ces étapes à l'infini (e.g. les agents bloquants sont Anna et Béa qui veulent toutes les deux *écrire un papier* ; on génère une nouvelle répartition selon laquelle Anna va *formaliser le modèle* et Béa va *écrire un papier* ; une fois ces tâches réintégrées à la répartitions d'origine, on a maintenant Anna et Cara qui veulent *formaliser le modèle*. Si l'on ne considère qu'Anna et Cara dans la nouvelle liste des agents bloquants, dans la nouvelle répartition générée Anna fera à nouveau *écrire un papier* et Cara fera *formaliser le modèle* : Anna et Béa seront donc à nouveau les agents bloquants.).

Répartitions ayant l'utilité la plus élevée

Afin de sélectionner une tâche qui corresponde à la meilleure répartition de tâches possible selon lui, l'agent x cherche les répartitions pour lesquelles l'utilité est la plus élevée.

Soit $R_x = \{R_{x,1}, \dots, R_{x,m}\}$ l'ensemble des m répartitions de tâches générées par x .

Si l'on note $R_{max_{R,x}} = \{R_{x,1}, \dots, R_{x,m(max_{R,x})}\}$ l'ensemble des $m(max_{R,x})$ tâches pour lesquelles l'utilité de la répartition est maximale, $R_{max_{R,x}}$ est donné par la formule 6.19 :

$$R_{max_{R,x}} = \arg \max_{R_{x,j} \in R_x} (U_x(R_{x,j})) \quad . \quad (6.19)$$

Si cet ensemble ne contient qu'une seule répartition, alors x sélectionne la tâche correspondant à cette répartition.

Tâches ayant l'utilité la plus élevée

Si l'ensemble $R_{max_{R,x}}$ contient plus d'une répartition de tâches ($m(max_{R,x}) > 1$), x doit choisir l'une de ces répartitions.

Pour cela, nous proposons que x cherche l'ensemble $R_{max_{\tau,x}} = \{R_{x,1}, \dots, R_{x,m(max_{\tau,x})}\}$ des $m(max_{\tau,x})$ répartitions pour lesquelles l'utilité de la tâche à laquelle il est assigné est la plus élevée. Ce sous-ensemble est donné par la formule 6.20 :

$$R_{max_{\tau,x}} = \arg \max_{\tau_{k,x,j} \text{ de } R_{x,j} \in R_{max_{R,x}}} (U_x(\tau_{k,x,j})) \quad . \quad (6.20)$$

Si cet ensemble ne contient qu'une seule répartition, alors x sélectionne la tâche $\tau_k = \tau_{k,x,1}$ correspondant à cette répartition.

Si cet ensemble contient plus d'une répartition de tâches, alors x en sélectionne une au hasard¹³.

Choix des autres agents

Il est important de noter ici que, si l'agent x sélectionne une tâche $\tau_k = \tau_{k,x,j}$ correspondant à une répartition des tâches $R_{x,j}$ imaginée par lui-même, x n'impose ici rien aux autres agents : x essaie simplement de prendre la meilleure décision possible en tenant compte des autres. De leur côté, les autres agents font de même. Dans le cas où la confiance de x en les autres agents est mal placée, les prévisions de x sur ce que feront les autres ne seront pas justes.

6.5.4 Exemple

Dans notre exemple, les ensembles de tâches ayant une utilité maximale pour chaque agent et selon Anna sont les suivants :

- $T_{U_{max,a}} = \{\tau_m\}$,
- $T_{U_{max,b}^a} = \{\tau_e\}$,

13. On pourrait imaginer encore d'autres stratégies de sélection de tâches : par exemple, si l'ensemble $R_{max_{\tau,x}}$ contient plus d'une répartition de tâches, l'agent x pourrait privilégier les répartitions dans lesquelles les agents envers qui il est bienveillant sont assignés à des tâches utiles pour eux (i.e. des tâches τ_k telles que $U_{y_i}^x(\tau_k)$ est élevée).

- $T_{U_{max,c}}^a = \{\tau_e\}$, et
- $T_{U_{max,d}}^a = \{\tau_m\}$.

Selon Anna, la meilleure répartition possible des tâches est donc la suivante :

$$R_{a,1} = \{(a, \tau_m), (b, \tau_e), (c, \tau_e), (d, \tau_m)\}$$

Selon cette répartition Dora et elle-même feront la tâche *monter les meubles*, alors que Béa et Cara s'occuperont d'*installer l'équipement informatique*. Ici, Anna n'a pas besoin de calculer l'utilité de la répartition ; étant donné que c'est la répartition formée à partir des tâches qui ont l'utilité maximale pour chaque agent, cette répartition est nécessairement la meilleure. Nous verrons en 7.2.2 et 7.2.3 des exemples où cela ne sera pas le cas et où Anna devra générer d'autres répartitions de tâches.

Dans l'exemple que nous avons développé, les croyances d'Anna sur les autres agents correspondent à la réalité. Ses calculs sur l'importance des buts pour les autres agents et l'utilité des actions pour les autres agents sont donc justes (i.e. $U_{y_i}^a = U_{y_i}$). Béa et Cara choisiront effectivement les tâches *installer l'équipement informatique* et Dora choisira *monter les meubles*.

6.6 Processus récursif

Le processus de sélection de tâche est récursif : en effet, l'agent x raisonne initialement sur les tâches racines des arbres de tâches ACTIVITY-DL. Une fois que la tâche est sélectionnée, l'agent x ne peut rien faire si cette tâche ne correspond pas à une action. L'agent x doit donc maintenant à nouveau répéter son processus décisionnel sur des tâches de plus bas niveau, et ce jusqu'à sélectionner une tâche qui corresponde à une action. Nous étudions dans les sections suivantes les ensembles de tâches et d'agents sur lesquels raisonne l'agent x à chaque appel récursif de son processus de sélection de tâche.

6.6.1 Ensemble de tâches sur lesquelles x raisonne

On note $T_{x,n} = \{\tau_{k1}, \dots, \tau_{k\ell}\}$ l'ensemble des ℓ tâches sur lesquelles x raisonne à l'étape n de son processus récursif de sélection de tâche.

Initialement, l'agent x raisonne sur les tâches racines des arbres de tâches. Si l'on note Tr l'ensemble de ces tâches racines, on a donc $T_{x,1} = Tr$.

Soit τ_k la tâche sélectionnée par x à l'issue de l'étape n de la récursivité. Si τ_k est aussi une tâche feuille, alors la récursivité s'arrête et x tente de faire τ_k , comme expliqué précédemment (voir 6.2 – Action). Si τ_k est une tâche composée, x raisonne sur les sous-tâches de τ_k pour en sélectionner une :

- Dans le cas où une contrainte temporelle SEQ-ORD est attachée à τ_k , la seule tâche que x doit prendre en compte à l'étape $n + 1$ est la première des sous-tâches de τ_k qui ne sont pas encore faites. Soit τ_{kk} cette sous-tâche : $T_{x,n+1} = \{\tau_{kk}\}$. Dans ce cas, x n'aura donc pas de choix à faire à l'étape $n + 1$.

Par ailleurs, du fait que les informations dynamiques (notamment $c(\tau_k, t)$) sur lesquelles x a raisonné pour sélectionner τ_k sont représentatives de l'avancement des agents sur l'activité, il est impossible que x sélectionne une tâche τ_k ayant une contrainte SEQ-ORD

à l'étape n et finalement ne puisse pas sélectionner τ_{kk} à l'étape $n+1$ (e.g. x n'aurait pas les compétences pour faire τ_{kk}).

- Dans les autres cas (i.e. PAR-SIM et IND), x doit raisonner sur l'ensemble des sous-tâches de τ_k qui ne sont pas encore faites. Tout comme dans le chapitre précédent, on note $T_{\text{NOTDONE}}(\tau_k, t)$ l'ensemble de ces sous-tâches. On a donc $T_{x,n+1} = T_{\text{NOTDONE}}(\tau_k, t)$. De même que pour le cas SEQ-ORD et pour les mêmes raisons, il est également impossible que x sélectionne à l'étape n une tâche τ_k à laquelle est attachée une contrainte temporelle PAR-SIM ou IND et ne puisse faire aucune des tâches de $T_{\text{NOTDONE}}(\tau_k, t)$ à l'étape $n+1$.

Exemple. Dans notre exemple, Anna raisonne sur les tâches racines des arbres d'activité correspondant à ses buts : on a

$$T_{a,1} = Tr = \{\tau_m, \tau_e, \tau_p, \tau_f\} \quad .$$

Anna choisit la tâche $\tau_m = \textit{monter les meubles}$, qui est une tâche composée. La tâche τ_m comportant une contrainte temporelle IND, Anna doit maintenant choisir entre les deux sous-tâches de τ_m qui sont *monter le bureau* et *monter les tiroirs*. En effet, aucune de ces deux tâches n'étant terminée, on a donc

$$T_{x,2} = T_{\text{NOTDONE}}(\tau_m, t_0) = \{\tau_{mb}, \tau_{mt}\} \quad .$$

6.6.2 Ensemble des agents sur lesquels x raisonne

On note $A_{x,n} = \{x, y_1, \dots, y_{\ell-1}\}$ l'ensemble des ℓ agents sur lesquels x raisonne à l'étape n de son processus récursif de sélection de tâche.

Initialement, x raisonne sur l'ensemble des agents présents dans l'environnement. Soit A cet ensemble : on a $A_{x,1} = A$.

Si à l'issue de l'étape n de son raisonnement récursif x a choisi la tâche $\tau_k = \tau_{k_{x,j}}$ de la répartition de tâches $R_{x,j}$, alors à l'étape $n+1$ l'agent x raisonne sur l'ensemble des agents qui étaient également assignés à τ_k dans la répartition $R_{x,j}$. Si l'on désigne par $A(\tau_k)$ l'ensemble de ces agents, on a donc $A_{x,n+1} = A(\tau_k)$, avec forcément $A_{x,n+1} \subseteq A_{x,n}$.

Exemple. Initialement, et comme nous l'avons vu dans l'exemple développé dans ce chapitre, Anna raisonne sur l'ensemble des agents de l'environnement :

$$A_{x,1} = A = \{a, b, c, d\} \quad .$$

Anna choisit la tâche *monter les meubles* τ_m et pense que Dora choisira également la tâche τ_m puisque Dora est assignée tout comme elle à cette tâche dans la répartition qu'elle pense être la meilleure. Lorsqu'elle va récursivement chercher à décider laquelle des sous-tâches de τ_m elle souhaiterait faire, Anna devra donc tenir compte de Dora, et uniquement de Dora : si Anna pense que Béa ne montera pas les meubles, il est inutile d'imaginer ce que Béa préférerait faire entre *monter le bureau* ou *monter les tiroirs*. Il en va de même pour Cara. On aura donc

$$A_{x,2} = A(\tau_m) = \{a, d\} \quad .$$

Nous ne détaillons pas l'appel récursif du processus de sélection d'une tâche pour Anna ; cependant, Anna et Dora étant aussi motivées l'une que l'autre pour réaliser le but *monter les meubles* et Anna étant plus compétente pour la tâche *monter le bureau* nécessitant la compétence *bricolage – savoir utiliser un marteau* que pour la tâche *monter les tiroirs* nécessitant la compétence *bricolage – savoir utiliser un tournevis* alors que Dora est aussi compétente pour les deux, Anna choisira de *monter le bureau* alors que Dora choisira de *monter les tiroirs*.

6.7 Bilan du chapitre

Nous avons présenté dans ce chapitre un modèle d'agents s'appuyant sur le modèle de confiance MDS présenté dans le chapitre 4. Les agents ont plusieurs types de buts, à savoir des buts personnels et des buts collectifs, et peuvent avoir plusieurs buts de chaque type. Chaque agent a des dimensions personnelles d'intégrité, de bienveillance et de compétences correspondant aux dimensions du modèle de confiance MDS. Enfin, chaque agent a des croyances sur les autres agents de l'environnement.

Nous avons ensuite donné un aperçu du fonctionnement du moteur décisionnel proposé avant d'en détailler les mécanismes. Le processus de prise de décision décrit dans ce chapitre se découpe en deux phases distinctes : dans un premier temps, l'agent raisonne sur ses buts et sur les buts des autres de manière à calculer l'importance qu'il accorde à chacun de ces buts. En effet, un agent ayant plusieurs buts, il doit choisir lequel est le plus intéressant à réaliser dans l'immédiat.

Dans un second temps, l'agent raisonne sur les arbres de tâches ACTIVITY-DL correspondant à ses buts. En fonction de l'importance qu'il accorde à la tâche (i.e. importance du but auquel la tâche contribue) et de son niveau de compétence sur la tâche, l'agent calcule l'utilité pour lui de cette tâche. L'agent calcule également ce qu'il pense être l'utilité de chaque tâche pour les autres agents. Cela lui permet ensuite de générer des répartitions de tâches entre les agents : nous avons proposé à mécanisme de génération des répartitions qui permet de limiter le nombre de répartitions à générer. Si nécessaire (i.e. s'il y a plusieurs répartitions générées), l'agent calcule l'utilité de chaque répartition générée, de manière à sélectionner une tâche correspondant à la meilleure répartition possible.

Les agents raisonnant en priorité sur les tâches abstraites des arbres de tâches, le processus de *calcul de l'utilité – sélection de tâche* est récursif, et à chaque appel récursif, l'agent raisonne sur un sous-ensemble des sous-tâches de la tâche sélectionnée et sur les agents qui d'après lui devraient également sélectionner la même tâche que lui.

Nous avons présenté un exemple de génération de comportements pour quatre agents Anna, Béa, Cara et Dora sur un scénario simple, ce qui nous a permis d'illustrer le modèle d'agents et le moteur décisionnel proposés. Au travers de cet exemple, nous avons notamment mis en évidence les éléments suivants :

- (1) *Adoption de but.* Un agent peut adopter le but d'un autre s'il est bienveillant envers l'autre agent en question. Anna étant bienveillante envers Béa, elle adopte son but d'*écrire un papier*. Anna n'aide pas immédiatement Béa à écrire un papier car elle a d'autres priorités, mais une fois les activités correspondant à ses buts prioritaires terminées, elle pourra aider Béa.
- (2) *Augmentation ou diminution de motivation.* La motivation des agents pour réaliser les

buts collectifs varie en fonction de ce qu'ils pensent de la motivation des autres. La motivation d'Anna pour *installer l'équipement informatique* diminue légèrement car ce but est collectif et Anna constate que Béa et Cara sont moins motivées que Dora et elle-même.

- (3) *Délégation faible*. Le moteur décisionnel proposé permet de générer des comportements de délégation faible. Lorsqu'Anna sélectionne la tâche *monter les meubles* en pensant que Cara va *installer l'équipement informatique*, Anna adopte un comportement de délégation faible par rapport à Cara. En effet, bien que de son côté Cara fasse également des prédictions sur le comportement d'Anna, Cara n'est pas au courant du plan (i.e. de la répartition de tâches) sélectionné par cette dernière.
- (4) *Tenir compte des autres pour sélectionner une tâche*. Plus généralement, le moteur décisionnel proposé permet à chaque agent d'adapter ses comportements en fonction de ses croyances sur les autres, pour par exemple laisser quelqu'un *installer l'équipement informatique* car on pense qu'il ne peut pas *monter les meubles*. Nous verrons un exemple de cette situation en 7.2.3.
- (5) *Confiance mal placée*. Les dimensions personnelles des agents correspondant aux dimensions de la confiance du modèle MDS, il est facile de modéliser des agents ayant une confiance mal placée en les autres. En outre, la confiance mal placée peut également être située au niveau des buts de agents. Dans l'exemple que nous avons développé dans ce chapitre, les croyances des agents sur les autres sont justes. Les prédictions de chaque agent sur les comportements des autres sont donc justes. Nous verrons dans le chapitre suivant que lorsque la confiance des agents est mal placée, leurs prédictions seront fausses, ce qui produira des comportements qui ne seront pas optimaux. (voir 7.2.4)
- (6) *Raisonnement sur un arbre de tâches* ACTIVITY-DL. Les agents raisonnent sur des arbres de tâches décrits en ACTIVITY-DL et utilisent en particulier les contraintes et informations propagées aux tâches abstraites depuis les tâches feuilles jusqu'aux tâches racines et dont nous avons détaillé les mécanismes de propagation dans le chapitre précédent. Bien que l'exemple utilisé ici soit très simple, nous avons notamment constaté que :
 - Les agents raisonnent sur la contrainte de faisabilité collective pour déterminer les tâches qui sont réalisables par l'équipe. Les agents ne raisonneront pas sur (et donc ne sélectionneront pas) les tâches qui ne sont pas réalisables par l'équipe. Dans l'exemple présenté, toutes les tâches étaient réalisables par l'équipe.
 - Les agents raisonnent sur les informations dynamiques attachées à chaque tâche. En particulier, l'information dynamique sur les compétences nécessaires à un agent est utilisée pour calculer l'utilité de chaque tâche et celle sur le nombre maximal de personnes pouvant faire une tâche est utilisée pour générer les répartitions de tâches qui sont valides.

Parmi les éléments énumérés ci-dessus, les points (1), (2), (4) et (6) permettent en particulier de produire des comportements humains collaboratifs où la répartition des tâches peut se faire sans communication, ce qui répond à l'un de nos principaux objectifs. Les éléments (3) et (5) permettent d'autre part de générer des comportements non idéaux, ce qui correspond à notre second objectif. Nous pourrions ainsi entraîner l'apprenant à faire face à des situations où ses

coéquipiers ne se comportent pas de manière optimale.

Le moteur décisionnel présenté ici est conçu pour être utilisé pour chaque agent de l'environnement. Néanmoins, les agents se projetant à la place des autres, le nombre de calculs à faire pour un agent augmente de manière exponentielle lorsque le nombre d'agents dans l'environnement augmente. Cela ne pose pas de problème au regard de nos objectifs : générer des comportements de personnages virtuels autonomes qui soient cohérents, crédibles, explicables et représentatifs des comportements observés sur le terrain pour des activités collaboratives effectuées par des petites équipes d'agents où tenir compte de chaque individu est important.

Cependant, on pourrait imaginer une autre utilisation du moteur décisionnel proposé, qui permettrait de l'utiliser sur des équipes de taille légèrement plus grande : il serait possible de ne l'utiliser que pour un seul agent *chef*, auquel cas les tâches assignées à chaque agent de l'environnement dans la répartition de tâches choisie par l'agent chef seraient imposées par ce dernier aux autres agents de l'environnement ¹⁴.

Le chapitre suivant est consacré à la présentation d'autres exemples illustrant les comportements générés avec notre moteur décisionnel ainsi qu'à la présentation d'évaluations préliminaires de nos travaux.

14. Utiliser le moteur décisionnel pour un seul agent chef qui dirigerait l'équipe et choisirait les actions à faire par les autres agents impliquerait quelques modifications du moteur décisionnel proposé. En effet, il faudrait alors que le processus de décision récursif ne soit pas centré uniquement autour de l'agent x , mais soit déroulé par x pour tous les agents de l'environnement jusqu'à assigner une tâche feuille à chacun : dans la proposition de moteur décisionnel présentée, on ne déroule la récursivité jusqu'aux tâches feuilles que pour l'agent x , et éventuellement les agents y_i qui sont toujours assignés aux mêmes tâches que x dans les répartitions successives de tâches générées par x au cours de la récursivité. Il faudrait alors dérouler la récursivité pour tous les agents y_i .

Chapitre 7

Cas d’usages et résultats préliminaires

Nous avons proposé dans le chapitre 5 une augmentation du langage de description de l’activité ACTIVITY-DL qui permet de supporter la description d’activités collectives. Nous avons également proposé des mécanismes qui permettent de propager des contraintes et des informations depuis les tâches feuilles des arbres de tâches jusqu’aux tâches racines et qui visent à supporter le raisonnement des agents sur de telles activités.

Nous avons ensuite présenté dans le chapitre 6 un moteur décisionnel qui permet de générer des comportements d’agents prenant part à des activités collectives. Ce moteur décisionnel permet en particulier aux agents de raisonner sur l’importance qu’ils accordent à leurs buts personnels et collectifs : l’importance accordée par un agent à ses buts est influencée par l’importance que les autres agents accordent à leurs propres buts selon lui. Une fois l’importance que l’agent accorde à ses buts calculée, l’agent recherche la tâche qu’il doit effectuer pour atteindre ses buts et raisonne pour cela sur les arbres de tâches ACTIVITY-DL. Encore une fois, l’agent tient compte des autres pour sélectionner une tâche.

Ce chapitre est consacré à la présentation de cas d’usage et de résultats préliminaires. Nous décrivons tout d’abord dans la section 7.1.1 l’architecture que nous avons mise en place et nous abordons la question de la performance de notre système. Nous présentons dans la section 7.2 plusieurs scénarios d’exemple qui nous permettront d’illustrer le fonctionnement du moteur décisionnel des agents et les comportements produits en fonction du paramétrage de ceux-ci. Enfin la section 7.3 est dédiée à la présentation de deux évaluations perceptives préliminaires que nous avons conduites.

7.1 Implémentation

Nous décrivons l’architecture de notre système dans la section 7.1.1, puis nous abordons la question de la performance du système dans la section 7.1.2.

7.1.1 Description de l’architecture

Le moteur décisionnel ainsi que les mécanismes de traitement des arbres de tâches ACTIVITY-DL proposés ont été implémentés en Python. Le fonctionnement de ces deux éléments est ex-

pliqué dans les sections suivantes.

Traitement de l'activité

Étant donné que nous ne considérons que des agents qui partagent un même environnement, et qui donc peuvent observer directement ce que font les autres, nous ne travaillons qu'avec une seule représentation de l'activité, partagée entre les agents, i.e. les agents *n'ont pas* de copies individuelles du même arbre de tâches. En début de simulation, la génération des contraintes de faisabilité collective (voir 5.4) ne sera donc faite qu'une seule fois sur chaque arbre de tâches différent. De même, lorsqu'un agent effectuera une action dans l'environnement, la mise à jour des informations spécifiées sur les tâches de l'arbre (voir 5.5) ne sera faite qu'une seule fois sur l'arbre partagé. Nous abordons ici l'implémentation de ces mécanismes.

Les arbres de tâches décrits en ACTIVITY-DL sont représentés en XML (voir annexe B.2). Le fichier contenant la description de l'activité est donc lu et des objets des classes `AbstractTask` ou `LeafTask` sont créés pour représenter respectivement les tâches abstraites et les tâches feuilles. Les deux classes `AbstractTask` et `LeafTask` héritent d'une classe abstraite `Task` qui contient certaines méthodes et attributs communs aux tâches abstraites et aux tâches feuilles, comme l'attribut booléen `done` qui indique si la tâche est terminée ou non, et l'attribut `parent` qui correspond à un pointeur sur la tâche mère.

Les classes `AbstractTask` et `LeafTask` contiennent chacune des éléments qui leurs sont spécifiques : par exemple la classe `AbstractTask` contient une liste de pointeurs vers ses tâches filles `subtasksList`, ainsi que des attributs `tempConstraint` et `logicConstraint` qui correspondent respectivement aux contraintes temporelles et logiques. La classe `LeafTask` contient par exemple les attributs `preconditionsList` et `effectsList` qui désignent respectivement la liste des pré-conditions et des effets de l'action correspondant à la tâche feuille.

Au moment de la création des objets `AbstractTask` et `LeafTask` sont générées les contraintes de faisabilité collective ainsi que les informations dynamiques comme expliqué respectivement dans les sections 5.4 et 5.5. Bien que nous ayons mentionné dans le chapitre 5 que, selon les cas d'application, il serait possible de considérer les contraintes de faisabilité collective en termes de compétences et de nombres minimaux de personnes comme non bloquantes, *nous considérons toujours les contraintes de faisabilité collective comme bloquantes*¹.

Chaque fois qu'un agent termine une action, l'état de l'objet de la classe `LeafTask` correspondant est mis à jour : la valeur de l'attribut `done` passe à `vrai`. La tâche feuille prévient ensuite son `parent`, qui est également mis à jour en conséquence : si l'action a permis de terminer la tâche mère, alors la valeur de l'attribut `done` de celle-ci est également mise à jour. Les informations dynamiques attachées à la tâche mère sont également mises à jour.

Fonctionnement des agents

Chaque agent est un objet de la classe `Agent`. Pour permettre un fonctionnement concurrentiel des agents, chaque agent fonctionne dans un thread qui lui est propre. Comme nous l'avons

1. Faire en sorte que les contraintes de faisabilité collective soient non bloquantes nécessiterait quelques changements dans notre implémentation mais demanderait surtout une plus grande réflexion sur le format de description de l'activité (voir 8.2.1).

expliqué dans le chapitre 6, les agents suivent un cycle *perception - décision - action*. Chaque agent fonctionne donc de manière indépendante des autres et les cycles *perception - décision - action* de chaque agent ne sont pas nécessairement synchronisés. Nous passons en revue le déroulement de chacune des phases du cycle de l'agent dans les paragraphes suivants.

Phase de *perception*. Notre module n'a pas pu être couplé à un environnement virtuel, nous avons donc un objet spécial `virtualEnvironment` de la classe `VirtualEnvironment` qui remplace l'environnement virtuel. Cet objet peut contenir les éléments de l'environnement virtuel, comme un élément `victime` dans le cadre d'un scénario de sauvetage de victime tel que celui décrit dans le chapitre 5. Le `virtualEnvironment` contient notamment les informations sur ce que chaque agent est en train de faire (e.g. l'agent *Anna* fait *couper le compteur*). Chaque agent a accès à ces informations lors de la phase de perception.

Chaque agent a également accès aux informations relatives à l'avancement des activités décrites en `ACTIVITY-DL` : l'agent perçoit l'état des tâches (i.e. si elles sont terminées ou pas – attribut `done` ; les contraintes de faisabilité collective et les informations dynamiques qui y sont attachées ; etc.).

Phase de *décision*. La phase de décision est elle-même découpée en deux processus distincts : (1) l'agent raisonne sur l'importance qu'il accorde à ses buts puis (2) l'agent sélectionne une action à faire.

Dans l'implémentation actuelle, le processus (1) de raisonnement sur l'importance des buts n'est exécuté qu'en début de simulation : les agents calculent donc une seule fois l'importance qu'ils accordent à leurs buts et cette importance restera donc constante tout au long de la simulation². Le calcul de l'importance des buts est fait comme expliqué dans la section 6.3.

Le processus (2) de sélection de tâche est exécuté à chaque fois. Il correspond à une méthode `decide(tasksList, agentsList)` qui retourne un pointeur `task` sur la tâche choisie et où les arguments `tasksList` et `agentsList` correspondent respectivement à la liste des tâches et à la liste des autres agents sur lesquelles l'agent raisonne. Initialement, `agentsList` contient tous les autres agents de l'environnement et `tasksList` contient toutes les tâches racines des arbres de tâches dont les pré-conditions et dont les contraintes de faisabilité collective sont vérifiées. En effet, comme nous l'avons précisé plus haut, nous considérons toujours ces contraintes comme bloquantes. Si elles ne sont pas vérifiées, les agents ne pourront pas faire les tâches correspondantes et il est donc inutile de prendre en compte ces tâches dans le processus de sélection de tâche.

Cette méthode appelle alors les méthodes suivantes, dans cet ordre :

- `compute_utilities(tasksList)` : permet à l'agent de calculer les utilités des tâches de `tasksList` comme décrit section 6.4.1.
- `select_task(tasksList, agentsList)` : permet à l'agent de générer les répartitions des tâches et de sélectionner la tâche correspondant à la meilleure répartition comme décrit section 6.5. Comme expliqué section 6.6, si la tâche sélectionnée est une tâche abstraite, alors `decide` est appelée récursivement avec comme arguments les nouvelles listes de

2. Ce choix est critiquable et on pourrait souhaiter que les agents recalculent l'importance qu'ils accordent à leurs buts durant la simulation (voir 8.3.1).

tâches et d'agents sur lesquels l'agent doit maintenant raisonner. Si la tâche sélectionnée est une tâche feuille, alors un pointeur sur cette tâche est retourné.

La méthode `decide` retourne finalement le pointeur `task` renvoyé par la méthode `select_task`. Le pointeur `task` renvoyé par `decide` correspond donc à une tâche feuille, c'est-à-dire à l'action que l'agent souhaite faire.

Phase d'action. Une fois l'action sélectionnée, l'agent tente de l'exécuter. Trois cas peuvent alors se produire :

- *Échec de l'exécution.* Il est possible que l'exécution de l'action échoue si l'état des pré-conditions de l'action a changé entre le moment où l'agent l'a choisie et le moment où il tente de l'exécuter. Par exemple, pour une action qui peut être effectuée par au plus une personne, si Anna et Béa choisissent l'action "en même temps", seule la première qui tentera de l'exécuter le pourra. Dans le cas où l'exécution échoue, la phase d'action se termine et l'agent recommence un nouveau cycle *perception - décision - action*.
- *Attente.* Si l'exécution n'échoue pas, il est néanmoins possible que l'agent doive attendre pour pouvoir faire l'action. En effet, pour une action τ_k nécessitant $n_{min}(\tau_k)$ personnes au minimum, si $n(\tau_k) < n_{min}(\tau_k)$ agents choisissent de faire cette action, ces $n(\tau_k)$ agents doivent attendre puisque la contrainte de faisabilité collective est bloquante. Autrement dit, lorsqu'une action nécessite plusieurs agents pour être exécutée, le premier agent qui la choisit doit obligatoirement attendre que d'autres la choisissent, ne serait-ce que quelques instants.

Dans notre implémentation, les agents n'ont pas la possibilité de faire autre chose en attendant de pouvoir exécuter l'action qu'ils ont choisie.

Pour éviter les situations d'inter-blocage où tous les agents attendent l'aide d'un autre, nous avons fait en sorte que les agents n'attendent pas plus longtemps qu'un seuil ϵ . Une fois ce seuil d'attente dépassé, les agents abandonnent l'action pour laquelle ils attendaient et la phase d'action se termine prématurément. Les agents commencent alors un nouveau cycle *perception - décision - action*.

- *Exécution normale.* Lorsque les agents peuvent exécuter l'action normalement, ils le font. Ils sont alors "bloqués" pendant la durée de l'action : ils ne peuvent pas abandonner une action en cours d'exécution pour faire autre chose, même si cette action est longue. Les actions peuvent avoir des durées variables, ce qui explique que les cycles *perception - décision - action* des agents ne soient pas nécessairement synchronisés.

Nous abordons dans la section suivante la performance de notre système.

7.1.2 Exécution du système

L'implémentation proposée supporte la simulation de comportements d'agents ; cependant les résultats obtenus en termes de performance informatique (i.e. capacité à générer les comportements des agents et vitesse de génération de ces comportements) dépendent d'un élément principal : le nombre de répartitions de tâches que les agents doivent générer pour pouvoir choisir une tâche (voir 6.5). En effet, ce nombre de répartitions peut être extrêmement élevé selon le nombre d'agents présents dans l'environnement et le nombre de tâches considérées. Or la génération d'un trop grand nombre de répartitions peut avoir deux conséquences néfastes :

- *Ralentissement du système.* Générer les répartitions demande du temps à l'agent. Si le nombre de répartitions est très grand, le temps de réflexion peut l'être aussi, auquel cas le moteur décisionnel ne permet pas de générer les comportements des agents en temps réel. Ceci constitue une limite de notre système qui peut néanmoins être facilement contournée en enregistrant les comportements produits par notre moteur décisionnel puis en les rejouant.
- *Échec du système.* Lorsqu'un agent x tente de générer un nombre de répartitions vraiment trop grand, des erreurs de mémoire se produisent et le thread dans lequel s'exécute l'agent s'arrête. Ce cas constitue alors un échec de notre système à générer les comportements des agents.

Les principaux facteurs qui influencent l'occurrence de l'une de ces deux situations sont les suivants :

- En premier lieu, la performance de notre système dépend du *nombre d'agents* dont on cherche à générer les comportements. En effet, le nombre de répartitions augmente de manière exponentielle avec le nombre d'agents présents.
- La performance de notre système dépend également de la *complexité de l'activité* sur laquelle les agents raisonnent. On entend par *complexité de l'activité* à la fois le nombre de tâches sur lesquelles les agents doivent raisonner, mais aussi et surtout la "forme" des arbres de tâches sur lesquels les agents raisonnent : si ces arbres ne comportent que des contraintes d'ordonnancement temporel SEQ-ORD, alors les agents n'ont pas vraiment de choix à faire concernant la répartition des tâches : à un instant t , une seule tâche peut être faite, il faut simplement décider lequel des agents fera cette tâche. En revanche si les arbres de tâches contiennent de nombreuses tâches τ_k dont les contraintes d'ordonnancement temporel sont PAR-SIM ou IND, les agents doivent à chaque fois choisir laquelle des sous-tâches de τ_k ils souhaitent effectuer. Or pour faire ce choix les agents peuvent être amenés à générer un très grand nombre de répartitions de tâches, ce qui nuit à la performance du système proposé. Plus l'arbre de tâches contient un grand nombre de tâches dont les contraintes d'ordonnancement temporel sont PAR-SIM ou IND, plus cet arbre sera considéré comme *complexe* pour notre système.
- Enfin, la performance de notre système dépend du *nombre de compétences* que les agents peuvent avoir et qui peuvent être spécifiées sur les arbres de tâches : plus le nombre de compétences différentes est grand, plus il est probable que les agents aient des niveaux de compétence différents sur chaque tâche (i.e. chaque agent aura des niveaux de compétence différents pour chacune des compétences spécifiées dans l'arbre de tâches), et donc des utilités différentes pour chaque tâche. Or l'agent x est amené à générer de nombreuses répartitions de tâches lorsque les ensembles de tâches $T_{U_{max},x}$ et T_{U_{max},y_i}^x pour lesquelles l'utilité de chaque agent est maximisée contiennent un grand nombre de tâches. La probabilité que ces ensembles contiennent un grand nombre de tâches diminue lorsque les tâches ont des utilités différentes et donc lorsque le nombre de compétences augmente.

Bien que la performance de notre système soit limitée par ces facteurs, nos objectifs nos sont atteints : notre système supporte la génération de comportements en temps réel pour de petites équipes d'une dizaine d'agents et sur des activités relativement complexes comportant plusieurs

centaines de tâches.

Nous présentons dans la section suivante des cas d'usage qui illustrent le fonctionnement du moteur décisionnel proposé dans le chapitre précédent.

7.2 Cas d'usage

Nous avons présenté dans le chapitre 6 un exemple du fonctionnement du moteur décisionnel proposé. Nous présentons dans cette section d'autres exemples permettant d'illustrer les comportements pouvant être produits par notre moteur décisionnel.

Nous reprenons le scénario d'exemple développé dans le chapitre précédent. Nous rappelons que dans ce scénario, quatre agents Anna, Béa, Cara et Dora ont la possibilité de réaliser les activités *monter les meubles*, *installer l'équipement informatique*, *formaliser le modèle* et *écrire un papier*. Ces activités sont plus précisément décrites section 6.1.4 et peuvent être représentées sous forme d'arbres de tâches ACTIVITY-DL comme dans la figure 7.1 (également présentée dans le chapitre 6 page 105).

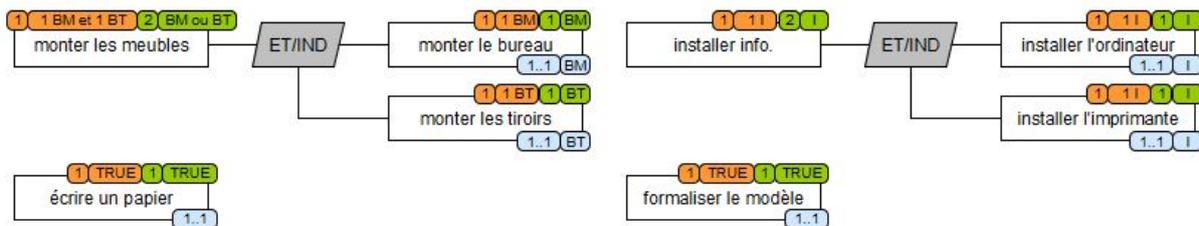


FIGURE 7.1 – Arbres de tâches décrits en ACTIVITY-DL correspondant au scénario détaillé en 6.1.4. Les bulles bleues en bas de chaque tâche correspondent aux informations renseignées lors de la description de l'activité. Les bulles oranges et les bulles vertes en haut de chaque tâche correspondent respectivement aux contraintes de faisabilité collective et aux informations dynamiques propagées dynamiquement par le système (voir chapitre 5).

Les buts, les dimensions et les croyances de chacun des agents seront donnés pour chacune des variantes du scénario présentées dans ce chapitre.

7.2.1 Influence de l'intégrité – scénarios *S2a* et *S2b*

Dans le scénario présenté dans le chapitre précédent (voir 6.1.4, tableau 6.1), Anna et Dora étaient plutôt intègres alors que Béa et Cara étaient moyennement intègres. Anna et Dora accordaient donc beaucoup d'importance aux buts collectifs de *monter les meubles* et d'*installer l'équipement informatique*, ce qui entraînait une hausse de la motivation de Béa et Cara pour ces buts. En particulier, Béa, tout comme Anna, privilégiait la réalisation du but collectif par rapport à celle de son but personnel. Nous présentons ici deux variantes du scénario de référence *S1* dans lesquelles l'intégrité de l'agent Anna varie.

Scénario $S2a$: Anna n'est pas très intègre

Dans ce scénario, les buts et dimensions des agents restent identiques par rapport au scénario $S1$, hormis l'intégrité de l'agent Anna : Anna n'est pas très intègre. Encore une fois nous considérons que les croyances des agents sur les autres correspondent à la réalité. Les caractéristiques des agents (buts et dimensions personnelles) sont données dans le tableau 7.1. L'élément en gras correspond à l'élément qui diffère par rapport au scénario de référence $S1$.

TABLE 7.1 – $S2a$ – Caractéristiques (en termes de buts et de dimensions personnelles) des agents dans le scénario $S2a$.

Sont notées en gras les données qui changent par rapport au scénario $S1$ (tableau 6.1 – ici l'intégrité d'Anna passe de 0.75 à 0.25).

		Anna	Béa	Cara	Dora
Buts	Collectifs	$\Gamma_{a,team} = \{\gamma_m, \gamma_e\}$	$\Gamma_{b,team} = \{\gamma_e\}$	$\Gamma_{c,team} = \{\gamma_m, \gamma_e\}$	$\Gamma_{d,team} = \{\gamma_m, \gamma_e\}$
	Personnels	$\Gamma_{a,self} = \{\gamma_f\}$	$\Gamma_{b,self} = \{\gamma_p\}$	$\Gamma_{c,self} = \emptyset$	$\Gamma_{d,self} = \emptyset$
	Ensemble	$\Gamma_a = \{\gamma_m, \gamma_e, \gamma_f\}$	$\Gamma_b = \{\gamma_e, \gamma_p\}$	$\Gamma_c = \{\gamma_m, \gamma_e\}$	$\Gamma_d = \{\gamma_m, \gamma_e\}$
Intégrité		$i_a = 0.25$	$i_b = 0.5$	$i_c = 0.5$	$i_d = 0.75$
Bienveillance	Anna	—	$b_{b,a} = 0.75$	$b_{c,a} = 0.75$	$b_{d,a} = 0.75$
	Béa	$b_{a,b} = 0.75$	—	$b_{c,b} = 0.75$	$b_{d,b} = 0.75$
	Cara	$b_{a,c} = 0.75$	$b_{b,c} = 0.75$	—	$b_{d,c} = 0.75$
	Dora	$b_{a,d} = 0.75$	$b_{b,d} = 0.75$	$b_{c,d} = 0.75$	—
Compét.	Marteau	$a_{a,BM} = 0.5$	$a_{b,BM} = 0$	$a_{c,BM} = 0$	$a_{d,BM} = 0.5$
	Tournevis	$a_{a,BT} = 0.25$	$a_{b,BT} = 0$	$a_{c,BT} = 0$	$a_{d,BT} = 0.5$
	Informatique	$a_{a,I} = 0.25$	$a_{b,I} = 0.5$	$a_{c,I} = 0.5$	$a_{d,I} = 0.25$

L'intégrité d'Anna étant déterminante pour le calcul de l'importance qu'Anna accorde à ses buts, cette importance varie par rapport au scénario $S1$:

- $\gamma_m \in \Gamma_{a,team}$ et $\gamma_e \in \Gamma_{a,team}$, donc $iImp_a(\gamma_m) = iImp_a(\gamma_e) = i_a = 0.25$.
- $\gamma_f \in \Gamma_{a,self}$, donc $iImp_a(\gamma_f) = 1 - i_a = 0.75$.
- Enfin, $iImp_a(\gamma_p) = 0$ car $\gamma_p \notin \Gamma_a$.

Anna est cependant influencée par la motivation des autres. Les autres agents étant tous plus motivés qu'Anna pour réaliser les buts collectifs, la motivation d'Anna pour réaliser ces buts augmente.

Nous ne détaillons pas les calculs intermédiaires et donnons dans le tableau 7.2 les valeurs d'importance finale accordée par chacun des agents à leurs buts, toujours selon Anna, pour les scénarios $S1$ et $S2a$. Néanmoins les valeurs calculées par Anna d'importance initiale et d'influence des agents sur la motivation des agents peuvent être trouvées en annexe respectivement dans les tableaux E.1 et E.2.

Dans le tableau 7.2, on peut remarquer que le seul but pour lequel les motivations sont constantes pour tous les agents dans les deux scénarios est le but *écrire un papier*. En effet, c'est le seul but qui n'est pas un but d'Anna, et donc la modification de l'intégrité d'Anna n'impacte pas la motivation des agents pour réaliser ce but.

La motivation de chaque agent pour le but *formaliser le modèle* augmente dans le scénario

TABLE 7.2 – *S2a* – Valeurs d'importance finale accordée par Anna, Béa, Cara et Dora aux buts *monter les meubles*, *installer l'équipement informatique*, *écrire un papier* et *formaliser le modèle* selon Anna dans les scénarios *S1* et *S2a*.

		<i>monter les meubles</i>	<i>installer info.</i>	<i>formaliser le modèle</i>	<i>écrire un papier</i>
Anna	<i>S1</i>	$fImp_a(\gamma_m) \simeq 0.688$	$fImp_a(\gamma_e) \simeq 0.667$	$fImp_a(\gamma_f) = 0.25$	$fImp_a(\gamma_p) = 0.125$
	<i>S2a</i>	$fImp_a(\gamma_m) \simeq 0.344$	$fImp_a(\gamma_e) \simeq 0.333$	$fImp_a(\gamma_f) = 0.75$	$fImp_a(\gamma_p) = 0.125$
Béa	<i>S1</i>	$fImp_b^a(\gamma_m) \simeq 0.167$	$fImp_b^a(\gamma_e) \simeq 0.542$	$fImp_b^a(\gamma_f) \simeq 0.062$	$fImp_b^a(\gamma_p) = 0.5$
	<i>S2a</i>	$fImp_b^a(\gamma_m) = 0.125$	$fImp_b^a(\gamma_e) \simeq 0.479$	$fImp_b^a(\gamma_f) \simeq 0.188$	$fImp_b^a(\gamma_p) = 0.5$
Cara	<i>S1</i>	$fImp_c^a(\gamma_m) \simeq 0.562$	$fImp_c^a(\gamma_e) \simeq 0.542$	$fImp_c^a(\gamma_f) \simeq 0.062$	$fImp_c^a(\gamma_p) = 0.125$
	<i>S2a</i>	$fImp_c^a(\gamma_m) \simeq 0.469$	$fImp_c^a(\gamma_e) \simeq 0.479$	$fImp_c^a(\gamma_f) \simeq 0.188$	$fImp_c^a(\gamma_p) = 0.125$
Dora	<i>S1</i>	$fImp_d^a(\gamma_m) \simeq 0.688$	$fImp_d^a(\gamma_e) \simeq 0.667$	$fImp_d^a(\gamma_f) \simeq 0.062$	$fImp_d^a(\gamma_p) = 0.125$
	<i>S2a</i>	$fImp_d^a(\gamma_m) \simeq 0.562$	$fImp_d^a(\gamma_e) \simeq 0.583$	$fImp_d^a(\gamma_f) \simeq 0.188$	$fImp_d^a(\gamma_p) = 0.125$

S2a par rapport au scénario *S1* : Anna donnant maintenant la priorité à son but personnel, les motivations des autres agents pour ce but augmentent également puisqu'ils apprécient tous Anna.

À l'inverse, les motivations des agents diminuent dans le scénario *S2a* par rapport au scénario *S1* : en effet, Anna est moins motivée par ces buts, or sa motivation entraînait dans le scénario *S1* une augmentation des motivations de Béa et Cara qui étaient moins motivées qu'elle, et elle n'avait pas d'influence sur la motivation de Dora car Dora et elle-même étaient autant motivées pour les buts collectifs. Dans le scénario *S2*, Anna entraîne maintenant une diminution de la motivation de Béa, Cara et Dora pour les buts collectifs puisqu'elle est moins motivée que les autres. Par ailleurs, on peut constater que Béa et Cara, qui sont initialement autant motivées pour leurs buts personnels que pour les buts collectifs (voir E.1), sont plus sensibles au manque de motivation d'Anna plutôt qu'à la grande motivation de Béa (i.e. l'importance finale accordée par ces agents aux buts collectifs est moindre que l'importance initiale) : cela s'explique par le fait que l'augmentation de la motivation d'un agent source lors de la prise en compte d'un agent cible est toujours proportionnelle à la bienveillance de l'agent source envers l'agent cible, ce qui n'est pas le cas pour la diminution de la motivation. Ici, Béa et Cara apprécient plutôt Dora, mais pas assez pour compenser la perte de motivation causée par Anna.

Finalement, Anna et Béa accordent dans le scénario *S2a* plus d'importance à leur but personnel respectif qu'aux buts collectifs, ce qui n'était pas le cas dans le scénario *S1*. Les tâches choisies par Anna, Béa, Cara et Dora dans le scénario *S2a* seront donc respectivement *formaliser le modèle*, *écrire un papier*, *installer l'équipement informatique* et *monter les meubles* (le tableau présentant les utilités des actions calculées par Anna pour tous les agents peut être trouvé en annexe E.3) alors que dans le scénario *S1*, ces mêmes agents choisissaient respectivement les tâches *monter les meubles*, *installer l'équipement informatique*, *installer l'équipement informatique* et *monter les meubles*.

Scénario S2b : Anna est moyennement intègre

Tout comme dans le scénario S2a, les buts et dimensions des agents restent identiques par rapport au scénario S1, hormis l'intégrité de l'agent Anna : Anna est moyennement intègre. Nous considérons toujours que les croyances des agents sur les autres correspondent à la réalité. Les caractéristiques des agents (buts et dimensions personnelles) sont données dans le tableau 7.3. L'élément en gras correspond à l'élément qui diffère par rapport au scénario de référence S1.

TABLE 7.3 – S2b – Caractéristiques (en termes de buts et de dimensions personnelles) des agents dans le scénario S2b.

Sont notées en gras les données qui changent par rapport au scénario S1 (tableau 6.1 – ici l'intégrité d'Anna passe de 0.75 à 0.5).

		Anna	Béa	Cara	Dora
Buts	Collectifs	$\Gamma_{a,team} = \{\gamma_m, \gamma_e\}$	$\Gamma_{b,team} = \{\gamma_e\}$	$\Gamma_{c,team} = \{\gamma_m, \gamma_e\}$	$\Gamma_{d,team} = \{\gamma_m, \gamma_e\}$
	Personnels	$\Gamma_{a,self} = \{\gamma_f\}$	$\Gamma_{b,self} = \{\gamma_p\}$	$\Gamma_{c,self} = \emptyset$	$\Gamma_{d,self} = \emptyset$
	Ensemble	$\Gamma_a = \{\gamma_m, \gamma_e, \gamma_f\}$	$\Gamma_b = \{\gamma_e, \gamma_p\}$	$\Gamma_c = \{\gamma_m, \gamma_e\}$	$\Gamma_d = \{\gamma_m, \gamma_e\}$
Intégrité		$i_a = 0.5$	$i_b = 0.5$	$i_c = 0.5$	$i_d = 0.75$
Bienveillance	Anna	—	$b_{b,a} = 0.75$	$b_{c,a} = 0.75$	$b_{d,a} = 0.75$
	Béa	$b_{a,b} = 0.75$	—	$b_{c,b} = 0.75$	$b_{d,b} = 0.75$
	Cara	$b_{a,c} = 0.75$	$b_{b,c} = 0.75$	—	$b_{d,c} = 0.75$
	Dora	$b_{a,d} = 0.75$	$b_{b,d} = 0.75$	$b_{c,d} = 0.75$	—
Compét.	Marteau	$a_{a,BM} = 0.5$	$a_{b,BM} = 0$	$a_{c,BM} = 0$	$a_{d,BM} = 0.5$
	Tournevis	$a_{a,BT} = 0.25$	$a_{b,BT} = 0$	$a_{c,BT} = 0$	$a_{d,BT} = 0.5$
	Informatique	$a_{a,I} = 0.25$	$a_{b,I} = 0.5$	$a_{c,I} = 0.5$	$a_{d,I} = 0.25$

Dans ce scénario, Anna étant moyennement intègre, on a :

- Pour tous les buts d'Anna : $iImp_a(\gamma_m) = iImp_a(\gamma_e) = iImp_a(\gamma_f) = 0.5$
- *écrire un papier* n'étant pas un but d'Anna, on a toujours $iImp_a(\gamma_p) = 0$.

L'influence des autres sur Anna sera donc capitale pour l'importance finale accordée aux buts par Anna et pour le choix d'Anna quant à une tâche à effectuer. Tout comme dans le scénario précédent, nous ne détaillons pas les étapes intermédiaires des valeurs calculées par Anna d'importance initiale et d'influence des agents sur la motivation des autres, mais ces valeurs peuvent être trouvées en annexe respectivement dans les tableaux E.4 et E.5. Le tableau 7.4 présente les valeurs d'importance finale accordée par chacun des agents à leurs buts, toujours selon Anna, dans les scénarios S1 et S2b.

Selon le tableau 7.4, et tout comme dans le scénario S2a, les motivations des agents pour *écrire un papier* ne varient pas dans le scénario S2b par rapport au scénario S1. Concernant le but *formaliser le modèle*, la motivation d'Anna étant plus forte dans le scénario S2b que dans le scénario S1, celle des agents Béa, Cara et Dora l'est également.

En ce qui concerne les buts collectifs, Anna est influencée par les autres : que ce soit pour *monter les meubles* ou *installer l'équipement informatique*, l'influence de Béa et Cara est nulle puisque ces deux agents ont également une intégrité de 0.5. Anna appréciant Dora, elle est influencée de manière positive par l'importance que Dora accorde aux buts collectifs. L'importance

TABLE 7.4 – *S2b* – Valeurs d'importance finale accordée par Anna, Béa, Cara et Dora aux buts *monter les meubles*, *installer l'équipement informatique*, *écrire un papier* et *formaliser le modèle* selon Anna.

		<i>monter les meubles</i>	<i>installer info.</i>	<i>formaliser le modèle</i>	<i>écrire un papier</i>
Anna	<i>S1</i>	$fImp_a(\gamma_m) \simeq 0.688$	$fImp_a(\gamma_e) \simeq 0.667$	$fImp_a(\gamma_f) = 0.25$	$fImp_a(\gamma_p) = 0.125$
	<i>S2b</i>	$fImp_a(\gamma_m) \simeq 0.531$	$fImp_a(\gamma_e) \simeq 0.521$	$fImp_a(\gamma_f) = 0.5$	$fImp_a(\gamma_p) = 0.125$
Béa	<i>S1</i>	$fImp_b^a(\gamma_m) \simeq 0.167$	$fImp_b^a(\gamma_e) \simeq 0.542$	$fImp_b^a(\gamma_f) \simeq 0.062$	$fImp_b^a(\gamma_p) = 0.5$
	<i>S2b</i>	$fImp_b^a(\gamma_m) \simeq 0.146$	$fImp_b^a(\gamma_e) \simeq 0.521$	$fImp_b^a(\gamma_f) = 0.125$	$fImp_b^a(\gamma_p) = 0.5$
Cara	<i>S1</i>	$fImp_c^a(\gamma_m) \simeq 0.562$	$fImp_c^a(\gamma_e) \simeq 0.542$	$fImp_c^a(\gamma_f) \simeq 0.062$	$fImp_c^a(\gamma_p) = 0.125$
	<i>S2b</i>	$fImp_c^a(\gamma_m) \simeq 0.531$	$fImp_c^a(\gamma_e) \simeq 0.521$	$fImp_c^a(\gamma_f) = 0.125$	$fImp_c^a(\gamma_p) = 0.125$
Dora	<i>S1</i>	$fImp_d^a(\gamma_m) \simeq 0.688$	$fImp_d^a(\gamma_e) \simeq 0.667$	$fImp_d^a(\gamma_f) \simeq 0.062$	$fImp_d^a(\gamma_p) = 0.125$
	<i>S2b</i>	$fImp_d^a(\gamma_m) = 0.625$	$fImp_d^a(\gamma_e) = 0.625$	$fImp_d^a(\gamma_f) = 0.125$	$fImp_d^a(\gamma_p) = 0.125$

finale qu'Anna accorde aux buts collectifs est donc plus élevée que l'importance qu'elle accorde à son but personnel.

Il en va de même pour les agents Béa et Cara : cette fois, Anna n'ayant plus une influence négative sur elle (contrairement au scénario *S2a*), Béa accorde finalement plus d'importance au but collectif d'*installer l'équipement informatique* qu'à son but personnel.

Les tâches choisies par les agents seront les suivantes : Anna et Dora vont *monter les meubles* alors que Béa et Cara vont *installer l'équipement informatique* : bien que l'intégrité d'Anna soit différente dans les scénarios *S1* et *S2b*, on obtiendra la même séquence d'actions.

7.2.2 Influence de la bienveillance – scénario *S3*

Nous nous intéressons dans le scénario qui suit à l'influence de la bienveillance des agents sur l'importance qu'ils accordent aux différents buts.

Nous comparons cette fois le scénario *S2b* et le scénario présenté ici *S3*. Les buts et dimensions des agents restent encore une fois identiques par rapport aux scénarios précédents, hormis la bienveillance des agents Anna et Dora : Anna n'est pas bienveillante envers Dora, et Dora n'est pas bienveillante envers Anna. Nous considérons toujours que les croyances des agents sur les autres correspondent à la réalité. Les caractéristiques des agents (buts et dimensions personnelles) sont données dans le tableau 7.5. Les éléments en gras correspondent aux éléments qui diffèrent par rapport au scénario *S2b*.

Les valeurs d'importance initiale et d'influence de chaque agent sur les motivations des autres peuvent être trouvées en annexe respectivement dans les tableaux E.7 et E.8. Le tableau 7.6 présente les valeurs d'importance finale accordée par chacun des agents à leurs buts, toujours selon Anna, dans les scénarios *S3* et *S2b*.

On peut observer dans le tableau 7.6 tout d'abord que, Dora n'étant pas bienveillante envers Anna, elle n'adopte pas dans le scénario *S3* le but personnel de cette dernière, contrairement aux scénarios précédents. Anna ne pourra donc plus compter sur l'aide éventuelle de Dora pour *formaliser le modèle*.

De plus, Anna a toujours une influence négative sur l'importance accordée par Dora aux buts

TABLE 7.5 – $S3$ – Caractéristiques (en termes de buts et de dimensions personnelles) des agents dans le scénario $S3$.

Sont notées en gras les données qui changent par rapport au scénario $S2b$ (tableau 7.3 – ici la bienveillance de Anna envers Dora ainsi que la bienveillance de Dora envers Anna passent de 0.75 à 0.25 ; Anna a toujours une intégrité de 0.5).

		Anna	Béa	Cara	Dora
Buts	Collectifs	$\Gamma_{a,team} = \{\gamma_m, \gamma_e\}$	$\Gamma_{b,team} = \{\gamma_e\}$	$\Gamma_{c,team} = \{\gamma_m, \gamma_e\}$	$\Gamma_{d,team} = \{\gamma_m, \gamma_e\}$
	Personnels	$\Gamma_{a,self} = \{\gamma_f\}$	$\Gamma_{b,self} = \{\gamma_p\}$	$\Gamma_{c,self} = \emptyset$	$\Gamma_{d,self} = \emptyset$
	Ensemble	$\Gamma_a = \{\gamma_m, \gamma_e, \gamma_f\}$	$\Gamma_b = \{\gamma_e, \gamma_p\}$	$\Gamma_c = \{\gamma_m, \gamma_e\}$	$\Gamma_d = \{\gamma_m, \gamma_e\}$
Intégrité		$i_a = 0.5$	$i_b = 0.5$	$i_c = 0.5$	$i_d = 0.75$
Bienveillance	Anna	—	$b_{b,a} = 0.75$	$b_{c,a} = 0.75$	$b_{d,a} = 0.25$
	Béa	$b_{a,b} = 0.75$	—	$b_{c,b} = 0.75$	$b_{d,b} = 0.75$
	Cara	$b_{a,c} = 0.75$	$b_{b,c} = 0.75$	—	$b_{d,c} = 0.75$
	Dora	$b_{a,d} = 0.25$	$b_{b,d} = 0.75$	$b_{c,d} = 0.75$	—
Compét.	Marteau	$a_{a,BM} = 0.5$	$a_{b,BM} = 0$	$a_{c,BM} = 0$	$a_{d,BM} = 0.5$
	Tournevis	$a_{a,BT} = 0.25$	$a_{b,BT} = 0$	$a_{c,BT} = 0$	$a_{d,BT} = 0.5$
	Informatique	$a_{a,I} = 0.25$	$a_{b,I} = 0.5$	$a_{c,I} = 0.5$	$a_{d,I} = 0.25$

TABLE 7.6 – $S3$ – Valeurs d'importance finale accordée par Anna, Béa, Cara et Dora aux buts *monter les meubles, installer l'équipement informatique, écrire un papier et formaliser le modèle* selon Anna.

		<i>monter les meubles</i>	<i>installer info.</i>	<i>formaliser le modèle</i>	<i>écrire un papier</i>
Anna	$S2b$	$fImp_a(\gamma_m) \simeq 0.531$	$fImp_a(\gamma_e) \simeq 0.521$	$fImp_a(\gamma_f) = 0.5$	$fImp_a(\gamma_p) = 0.125$
	$S3$	$fImp_a(\gamma_m) = 0.5$	$fImp_a(\gamma_e) = 0.5$	$fImp_a(\gamma_f) = 0.5$	$fImp_a(\gamma_p) = 0.125$
Béa	$S2b$	$fImp_b^a(\gamma_m) \simeq 0.146$	$fImp_b^a(\gamma_e) \simeq 0.521$	$fImp_b^a(\gamma_f) = 0.125$	$fImp_b^a(\gamma_p) = 0.5$
	$S3$	$fImp_b^a(\gamma_m) \simeq 0.146$	$fImp_b^a(\gamma_e) \simeq 0.521$	$fImp_b^a(\gamma_f) = 0.125$	$fImp_b^a(\gamma_p) = 0.5$
Cara	$S2b$	$fImp_c^a(\gamma_m) \simeq 0.531$	$fImp_c^a(\gamma_e) \simeq 0.521$	$fImp_c^a(\gamma_f) = 0.125$	$fImp_c^a(\gamma_p) = 0.125$
	$S3$	$fImp_c^a(\gamma_m) \simeq 0.531$	$fImp_c^a(\gamma_e) \simeq 0.521$	$fImp_c^a(\gamma_f) = 0.125$	$fImp_c^a(\gamma_p) = 0.125$
Dora	$S2b$	$fImp_d^a(\gamma_m) = 0.625$	$fImp_d^a(\gamma_e) = 0.625$	$fImp_d^a(\gamma_f) = 0.125$	$fImp_d^a(\gamma_p) = 0.125$
	$S3$	$fImp_d^a(\gamma_m) = 0.625$	$fImp_d^a(\gamma_e) = 0.625$	$fImp_d^a(\gamma_f) = 0.0$	$fImp_d^a(\gamma_p) = 0.125$

collectifs : on peut constater que les valeurs d'importance finale des buts collectifs pour Dora sont identiques dans les scénarios $S2b$ et $S3$. En effet, que l'on soit bienveillant ou non envers une personne, le manque de motivation de cette personne nous impacte (Kerr, 1983).

Par ailleurs, on remarque que dans le scénario $S3$ les valeurs d'importance finale accordée par Anna aux buts *monter les meubles, installer l'équipement informatique et formaliser le modèle* sont identiques : en effet, tout comme dans le scénario précédent, l'influence de Béa et Cara sur l'importance accordée par Anna aux buts collectifs est nulle. En outre, Dora accorde plus d'importance qu'elle aux buts collectifs, mais Anna n'est pas bienveillante envers Dora : cette dernière n'a donc plus d'influence sur Anna.

Dans ce scénario, Anna choisira aléatoirement l'une des deux tâches *monter les meubles* ou

formaliser le modèle. Béa et Cara choisiront d'installer l'équipement informatique alors que Dora choisira de monter les meubles. L'explication de ces choix est présentée en annexe E.3.3.

7.2.3 Importance des compétences – scénario $S4$

Nous avons déjà pu observer dans le scénario $S1$ présenté dans le chapitre précédent que les compétences jouent un grand rôle dans la répartition des tâches : dans ce scénario, les agents Béa et Cara n'ayant pas la compétence bricolage (BM ou BT), ils ne peuvent pas faire la tâche *monter les meubles*, et Anna ne calcule aucune valeur d'utilité pour ces deux agents et pour cette tâche.

Nous présentons ici un autre scénario $S4$ dans lequel nous verrons un second exemple de l'influence de la compétence des agents sur leurs choix. Par ailleurs, ce scénario nous permettra également de mettre en évidence l'importance de la prise en compte des autres lors du choix d'une tâche.

Dans le scénario $S4$, les caractéristiques des agents sont les mêmes que dans le scénario $S1$, hormis l'intégrité de Béa et le niveau de compétence d'Anna en *informatique* : Béa est plutôt intègre et Anna est ici plutôt compétente en *informatique*. Les caractéristiques de tous les agents sont données dans le tableau 7.7 dans lequel les éléments en gras correspondent aux éléments pour lesquels la valeur varie par rapport au scénario de référence $S1$.

TABLE 7.7 – $S4$ – Caractéristiques (en termes de buts et de dimensions personnelles) des agents dans le scénario $S4$.

Sont notées en gras les données qui changent par rapport au scénario $S1$ (tableau 6.1 – ici l'intégrité de Béa passe de 0.5 à 0.75 et la compétence d'Anna en informatique passe de 0.25 à 0.75).

		Anna	Béa	Cara	Dora
Buts	Collectifs	$\Gamma_{a,team} = \{\gamma_m, \gamma_e\}$	$\Gamma_{b,team} = \{\gamma_e\}$	$\Gamma_{c,team} = \{\gamma_m, \gamma_e\}$	$\Gamma_{d,team} = \{\gamma_m, \gamma_e\}$
	Personnels	$\Gamma_{a,self} = \{\gamma_f\}$	$\Gamma_{b,self} = \{\gamma_p\}$	$\Gamma_{c,self} = \emptyset$	$\Gamma_{d,self} = \emptyset$
	Ensemble	$\Gamma_a = \{\gamma_m, \gamma_e, \gamma_f\}$	$\Gamma_b = \{\gamma_e, \gamma_p\}$	$\Gamma_c = \{\gamma_m, \gamma_e\}$	$\Gamma_d = \{\gamma_m, \gamma_e\}$
Intégrité		$i_a = 0.75$	$i_b = \mathbf{0.75}$	$i_c = 0.5$	$i_d = 0.75$
Bienveillance	Anna	—	$b_{b,a} = 0.75$	$b_{c,a} = 0.75$	$b_{d,a} = 0.25$
	Béa	$b_{a,b} = 0.75$	—	$b_{c,b} = 0.75$	$b_{d,b} = 0.75$
	Cara	$b_{a,c} = 0.75$	$b_{b,c} = 0.75$	—	$b_{d,c} = 0.75$
	Dora	$b_{a,d} = 0.25$	$b_{b,d} = 0.75$	$b_{c,d} = 0.75$	—
Compét.	Marteau	$a_{a,BM} = 0.5$	$a_{b,BM} = 0$	$a_{c,BM} = 0$	$a_{d,BM} = 0.5$
	Tournevis	$a_{a,BT} = 0.25$	$a_{b,BT} = 0$	$a_{c,BT} = 0$	$a_{d,BT} = 0.5$
	Informatique	$\mathbf{a_{a,I} = 0.75}$	$a_{b,I} = 0.5$	$a_{c,I} = 0.5$	$a_{d,I} = 0.25$

Dans ce scénario, les valeurs d'importance finale que les agents accordent à leurs buts varient quelque peu par rapport au scénario $S1$: Béa étant intègre, elle accorde plus d'importance au but collectif d'*installer l'ordinateur* qu'à son but personnel d'*écrire un papier*. Toutes les valeurs d'importance initiale, d'influence des autres agents et d'importance finale accordée aux buts sont données en annexe E.4 respectivement dans les tableaux E.10, E.11 et E.12.

Nous donnons dans le tableau 7.8 les utilités calculées par Anna dans le scénario S_4 : nous observons ici que, du fait qu'Anna est plus compétente que dans le scénario S_1 pour l'informatique, l'utilité qu'elle accorde à cette action est différente.

TABLE 7.8 – S_4 – Valeurs d'utilité pour toutes les tâches et pour tous les agents calculées par Anna : $U_a(\tau)$ représente l'utilité pour Anna de la tâche τ et $U_{y_i}^a(\tau)$ représente l'utilité pour y_i de faire la tâche τ selon Anna. (*installer info.* correspond à la tâche *installer l'équipement informatique.*)

	<i>monter les meubles</i>	<i>installer info.</i>	<i>formaliser le modèle</i>	<i>écrire un papier</i>
Anna	$U_a(\tau_m) \simeq 0.344$	$U_a(\tau_e) = 0.5$	$U_a(\tau_f) = 0.125$	$U_a(\tau_p) \simeq 0.062$
Bea	—	$U_b^a(\tau_e) \simeq 0.271$	$U_b^a(\tau_f) \simeq 0.031$	$U_b^a(\tau_p) = 0.25$
Cara	—	$U_c^a(\tau_e) \simeq 0.271$	$U_c^a(\tau_f) \simeq 0.031$	$U_c^a(\tau_p) \simeq 0.062$
Dora	$U_d^a(\tau_m) \simeq 0.344$	$U_d^a(\tau_e) \simeq 0.167$	$U_d^a(\tau_f) \simeq 0.031$	$U_d^a(\tau_p) \simeq 0.062$

Selon le tableau 7.8, les ensembles de tâches $T_{U_{max,a}}$ et $T_{U_{max,y_i}}^a$ pour lesquelles l'utilité est maximisée sont les suivants :

- $T_{U_{max,a}} = \{\tau_e\}$,
- $T_{U_{max,b}}^a = \{\tau_e\}$,
- $T_{U_{max,c}}^a = \{\tau_e\}$, et
- $T_{U_{max,d}}^a = \{\tau_m\}$.

Chacun de ces ensembles ne contient qu'un seul élément. Selon le processus décrit section 6.5.3, Anna génère donc la répartition qui correspond à ces tâches :

$$R_{a,1} = \{(a, \tau_e), (b, \tau_e), (c, \tau_e), (d, \tau_m)\} \quad .$$

Cette répartition n'est pas valide, puisque le nombre maximal d'agents pouvant participer à *installer l'équipement informatique* τ_e est $n_{max}(\tau_e, t_0) = 2$ (voir figure 7.1) et dans cette répartition, le nombre d'agents assignés à τ_e est $n(\tau_e) = 3$.

Anna identifie donc les agents bloquants : ces agents sont Béa, Cara et elle-même. Anna génère donc toutes les répartitions possibles de tâches entre ces trois agents et calcule l'utilité de chacune de ces trois répartitions. Ces répartitions sont données en annexe dans le tableau E.13, celle ayant la plus forte utilité est la répartition numéro 21 :

$$R_{a,21} = \{(a, \tau_m), (b, \tau_e), (c, \tau_e)\} \quad .$$

Anna remplace ensuite dans la répartition d'origine $R_{a,1}$ les tâches des agents bloquants par celles de la nouvelle répartition $R_{a,21}$:

$$R_{a,1} = \{(a, \tau_m), (b, \tau_e), (c, \tau_e), (d, \tau_m)\} \quad .$$

Cette fois la répartition est valide : Anna sélectionne donc la tâche *monter les meubles* correspondant à cette répartition. Béa et Cara s'occuperont d'*installer l'équipement informatique* alors que Cara fera également *monter les meubles*.

On observe donc dans ce scénario S_4 qu'Anna, bien qu'ayant une plus forte valeur d'utilité pour la tâche *installer l'équipement informatique* que pour *monter les meubles*, choisira de *monter*

les meubles. En effet, elle pense que Béa et Cara voudront *installer l'équipement informatique*, il est donc dans son intérêt de les laisser faire, puisqu'elles ne sont pas compétentes pour *monter les meubles*.

7.2.4 Confiance mal placée – scénario *S5*

Nous présentons ici un scénario *S5* très simple dans lequel la confiance d'Anna en les compétences de Cara est *mal placée* (voir 4.1.2). Dans ce scénario, les caractéristiques des agents sont les mêmes que dans le scénario *S1*, hormis la compétence d'Anna pour l'informatique qui est de 0.75. Ces caractéristiques sont présentées dans le tableau 7.9. Cependant, alors que dans le scénario *S1* les croyances de tous les agents correspondaient à la réalité, ici Anna a confiance en le niveau de compétence de Cara pour le bricolage, mais n'a pas confiance en le niveau de Cara pour l'informatique. Anna pense donc que Cara est plutôt moyennement compétente pour le bricolage, mais pas du tout pour l'informatique :

$$a_{c,BM}^a = a_{c,BT}^a = 0.5 \quad \text{et} \quad a_{c,I}^a = 0 \quad .$$

TABLE 7.9 – *S5* – Caractéristiques (en termes de buts et de dimensions personnelles) des agents dans le scénario *S5*.

Sont notées en gras les données qui changent par rapport au scénario *S1* (tableau 6.1 – ici la compétence d'Anna pour l'informatique passe de 0.25 à 0.5.

		Anna	Béa	Cara	Dora
Buts	Collectifs	$\Gamma_{a,team} = \{\gamma_m, \gamma_e\}$	$\Gamma_{b,team} = \{\gamma_e\}$	$\Gamma_{c,team} = \{\gamma_m, \gamma_e\}$	$\Gamma_{d,team} = \{\gamma_m, \gamma_e\}$
	Personnels	$\Gamma_{a,self} = \{\gamma_f\}$	$\Gamma_{b,self} = \{\gamma_p\}$	$\Gamma_{c,self} = \emptyset$	$\Gamma_{d,self} = \emptyset$
	Ensemble	$\Gamma_a = \{\gamma_m, \gamma_e, \gamma_f\}$	$\Gamma_b = \{\gamma_e, \gamma_p\}$	$\Gamma_c = \{\gamma_m, \gamma_e\}$	$\Gamma_d = \{\gamma_m, \gamma_e\}$
Intégrité		$i_a = 0.75$	$i_b = 0.5$	$i_c = 0.5$	$i_d = 0.75$
Bienveillance	Anna	—	$b_{b,a} = 0.75$	$b_{c,a} = 0.75$	$b_{d,a} = 0.25$
	Béa	$b_{a,b} = 0.75$	—	$b_{c,b} = 0.75$	$b_{d,b} = 0.75$
	Cara	$b_{a,c} = 0.75$	$b_{b,c} = 0.75$	—	$b_{d,c} = 0.75$
	Dora	$b_{a,d} = 0.25$	$b_{b,d} = 0.75$	$b_{c,d} = 0.75$	—
Compét.	Marteau	$a_{a,BM} = 0.5$	$a_{b,BM} = 0$	$a_{c,BM} = 0$	$a_{d,BM} = 0.5$
	Tournevis	$a_{a,BT} = 0.25$	$a_{b,BT} = 0$	$a_{c,BT} = 0$	$a_{d,BT} = 0.5$
	Informatique	$a_{a,I} = 0.5$	$a_{b,I} = 0.5$	$a_{c,I} = 0.5$	$a_{d,I} = 0.25$

Dans ce cas, les valeurs d'importance des buts pour les agents selon Anna sont les mêmes que celles du scénario *S1* et peuvent être trouvées dans le tableau 6.6. Les valeurs des utilités des tâches sont donc aussi identiques, hormis (i) l'utilité qu'Anna accorde à la tâche *installer l'équipement informatique* et (ii) l'utilité que Cara accorde aux tâches *monter les meubles* et *installer l'équipement informatique* selon Anna. Ces valeurs sont données dans le tableau 7.10.

Selon le tableau 7.10, dans le scénario *S5* les ensembles de tâches $T_{Umax,a}$ et T_{Umax,y_i}^a pour lesquelles l'utilité est maximisée sont les suivants :

- $T_{Umax,a} = \{\tau_m\}$,
- $T_{Umax,b}^a = \{\tau_e\}$,

TABLE 7.10 – $S5$ – Valeurs d'utilité pour toutes les tâches et pour tous les agents calculées par Anna : $U_a(\tau)$ représente l'utilité pour Anna de la tâche τ et $U_{y_i}^a(\tau)$ représente l'utilité pour y_i de faire la tâche τ selon Anna. (*installer info.* correspond à la tâche *installer l'équipement informatique.*)

	<i>monter les meubles</i>	<i>installer info.</i>	<i>formaliser le modèle</i>	<i>écrire un papier</i>
Anna	$U_a(\tau_m) \simeq 0.344$	$U_a(\tau_e) \simeq 0.333$	$U_a(\tau_f) = 0.125$	$U_a(\tau_p) \simeq 0.062$
Béa	—	$U_b^a(\tau_e) \simeq 0.271$	$U_b^a(\tau_f) \simeq 0.031$	$U_b^a(\tau_p) = 0.25$
Cara	$U_c^a(\tau_m) \simeq 0.281$	—	$U_c^a(\tau_f) \simeq 0.031$	$U_c^a(\tau_p) \simeq 0.062$
Dora	$U_d^a(\tau_m) \simeq 0.344$	$U_d^a(\tau_e) \simeq 0.167$	$U_d^a(\tau_f) \simeq 0.031$	$U_d^a(\tau_p) \simeq 0.062$

— $T_{U_{max,c}}^a = \{\tau_m\}$, et

— $T_{U_{max,d}}^a = \{\tau_m\}$.

Chacun de ces ensembles ne contient qu'un seul élément. Selon le processus décrit section 6.5.3, Anna génère donc la répartition correspondant à ces tâches :

$$R_{a,1} = \{(a, \tau_m), (b, \tau_e), (c, \tau_m), (d, \tau_m)\} \quad .$$

Tout comme dans le scénario $S4$, cette répartition n'est pas valide. Cette fois le nombre maximal d'agents pouvant participer à *monter les meubles* τ_m est $n_{max}(\tau_m) = 2$ (voir figure 7.1) et dans cette répartition, le nombre d'agents assignés à τ_m est $n(\tau_m) = 3$.

Anna identifie donc les agents bloquants : ces agents sont Cara, Dora et elle-même. Anna génère donc toutes les répartitions possibles de tâches entre ces trois agents et calcule l'utilité de chacune de ces trois répartitions. Ces répartitions sont données en annexe dans le tableau E.14, celle ayant la plus forte utilité est la répartition numéro 34 :

$$R_{a,34} = \{(a, \tau_e), (c, \tau_m), (d, \tau_m)\} \quad .$$

Anna remplace ensuite dans la répartition d'origine $R_{a,1}$ les tâches des agents bloquants par celles de la nouvelle répartition $R_{a,34}$:

$$R_{a,1} = \{(a, \tau_e), (b, \tau_e), (c, \tau_m), (d, \tau_m)\} \quad .$$

Cette fois la répartition est valide : Anna sélectionne donc la tâche *installer l'équipement informatique* correspondant à cette répartition. Anna **pense que** Béa s'occupera d'*installer l'équipement informatique* avec elle, alors que Cara et Dora s'occuperont de *monter les meubles*.

En réalité, Cara n'étant pas du tout compétente pour le bricolage ($a_{c,BM} = a_{c,BT} = 0$), elle ne choisira pas la tâche *monter les meubles*.

Le déroulement de la séquence d'actions effectuées par les agents dépend alors des agents qui seront les plus rapides entre Anna, Béa et Cara :

- (i) Si Anna et Béa sont les plus rapides à prendre leur décision, elles feront la tâche *installer l'équipement informatique*. Cara ne pourra pas participer à cette tâche même si celle-ci avait la plus forte utilité pour elle. Dans ce cas, l'agent Cara se trouve en situation d'*échec d'exécution de l'action choisie* (voir 7.1).

Cara commence donc un nouveau cycle *perception - décision - action*. Les agents Anna et Béa étant en train d'exécuter la tâche *installer l'équipement informatique*, dans les

répartitions de tâches générées par Cara, les agents Anna et Béa sont assignés à la tâche *installer l'équipement informatique*. La répartition de tâche maximisant l'utilité selon Cara sera donc la suivante :

$$R_{c,1} = \{(a, \tau_e), (b, \tau_e), (c, \tau_p), (d, \tau_m)\} \quad .$$

Cara sélectionnera la tâche correspondant à cette répartition : elle commencera à *écrire un papier* pour aider Béa à réaliser son but personnel.

- (ii) Si Béa et Cara sont les plus rapides à prendre leur décision, elles feront la tâche *installer l'équipement informatique*. De même que Cara dans le cas (i), Anna sera en situation d'échec d'exécution d'action. Anna procèdera alors de la même manière que Béa et choisira finalement de *monter les meubles*.
- (iii) Enfin Anna et Cara feront la tâche *installer l'équipement informatique* si elles sont les plus rapides à prendre leur décision. Dans ce cas, Béa sera en cas d'échec d'exécution d'action. En appliquant les mêmes mécanismes décisionnels que Cara dans le cas (i), Béa sélectionnera finalement la tâche *écrire un papier*.

En somme, on peut remarquer que la confiance mal placée d'Anna en les compétences de Cara peut entraîner des comportements non optimaux : dans les cas (i) et (iii), alors que quatre agents sont présents dans l'environnement et que quatre tâches sont à faire pour réaliser les buts collectifs (i.e. soit une tâche par agent pour une répartition optimale des tâches liées aux buts collectifs), les agents Cara et Béa ne participeront pas du tout à la réalisation des buts collectifs. L'un des agents Anna ou Dora devra donc faire deux tâches contribuant à la réalisation des buts collectifs. Ces comportements non idéaux sont néanmoins crédibles et explicables au regard des croyances des agents les uns sur les autres, ce qui répond à nos objectifs.

Pour conclure, nous précisons que si nous avons donné un exemple de confiance mal placée d'un agent par rapport aux compétences d'un autre, il y a d'innombrables cas de figure de confiance mal placée : dans le scénario *S5*, Anna surestime la compétence de Cara pour le *bricolage*. À l'inverse, Anna pourrait sous-estimer la compétence de Cara pour l'*informatique*. La confiance mal placée pourrait aussi bien être en rapport avec ce qu'un agent pense des buts des autres agents ou avec ce qu'un agent pense de la bienveillance ou de l'intégrité d'un autre agent. Un agent peut avoir simplement une croyance fautive sur un autre comme c'est le cas dans le scénario *S5* présenté ici, ou alors plusieurs croyances voire même l'ensemble des croyances d'un agent sur un autre peuvent être fausses. Enfin, un agent peut avoir une confiance mal placée par rapport à plusieurs autres agents à la fois (e.g. Anna a une confiance mal placée en Béa et en Cara).

7.2.5 Conclusion

Nous avons présenté ici cinq nouveaux scénarios dans lesquels les agents Anna, Béa, Cara et Dora ont les mêmes buts et peuvent réaliser les mêmes actions que dans le scénario de référence *S1* déroulé dans le chapitre précédent. Ces cinq nouveaux scénarios ont été obtenus à partir d'un paramétrage différent des agents.

Les scénarios *S2a*, *S2b* et *S3* nous ont permis d'étudier l'influence de l'intégrité et de la bienveillance des agents sur l'importance qu'ils accordent aux différents buts, mais aussi sur l'importance que les autres agents accordent aux différents buts : alors qu'Anna est plutôt

intègre dans le scénario de référence, elle est peu intègre dans le scénario *S2a* et accorde donc plus d'importance à son but personnel et en privilégie la réalisation. Anna influence les autres agents qui accordent alors moins d'importance aux buts collectifs que dans le scénario de référence. Dans le scénario *S2b*, Anna est moyennement intègre et l'influence des autres sur elle est alors capitale : Anna pense notamment que Dora accorde beaucoup d'importance aux buts collectifs, or comme Anna est bienveillante envers Dora, cela augmente l'importance qu'elle-même accorde à ces buts. Anna privilégie alors la réalisation des buts collectifs.

Tout comme dans le scénario *S2b*, Anna est moyennement intègre dans le scénario *S3*. Toutefois dans le scénario *S3*, Anna et Dora n'étant pas bienveillantes l'une envers l'autre, Anna n'est plus influencée par la forte motivation de Dora à réaliser les buts collectifs. Si Dora, contrairement au scénario *S2b*, n'adopte pas dans le scénario *S3* le but personnel d'Anna, elle est en revanche toujours influencée par la motivation moyenne d'Anna pour réaliser les buts collectifs, ce qui diminue sa propre motivation.

L'intégrité et la bienveillance des agents sont donc déterminantes pour le calcul de l'importance qu'ils accordent à leurs buts, et ces deux facteurs ne sont pas indépendants.

Au travers du scénario *S4* nous avons une nouvelle fois illustré l'influence des compétences des agents sur l'utilité que les agents accordent aux actions. Ce scénario nous a également permis de mesurer l'importance de la prise en compte des autres lors du choix d'une tâche : l'utilité d'*installer l'équipement informatique* pour Anna est plus élevée que celle de *monter les meubles* ; cependant Anna constate que Béa ou Cara ne pourra pas participer à l'activité collective si elle fait *installer l'équipement informatique*. Elle choisit donc de *monter les meubles*.

Enfin, nous avons donné un exemple avec le scénario *S5* de *confiance mal placée* d'un agent en un autre et de ses conséquences potentiellement néfastes (voir 4.1.2) : Anna surestime les compétences de Cara pour le bricolage et pense que laisser Cara *monter les meubles* est la meilleure solution. Elle choisit donc d'*installer l'équipement informatique*. Cara ne sachant en réalité pas du tout bricoler, les prévisions d'Anna sont fausses. Si Anna commence à *installer l'équipement informatique* avant Cara et/ou Béa, alors Cara ou Béa ne pourra pas participer à l'activité collective.

L'ensemble des scénarios présentés ici nous permet de faire la constatation suivante : sous réserve qu'ils aient les compétences nécessaires, les agents auront des comportements optimaux du point de vue de l'équipe (i.e. réalisation des buts collectifs en priorité et bonne organisation) s'ils sont intègres et si leurs croyances sur les autres sont justes. Lorsque les comportements des agents ne sont pas optimaux (e.g. *S2a* et *S5*), ils n'en restent pas moins cohérents et explicables, ce qui répond à nos objectifs.

7.3 Évaluations perceptives préliminaires

Pour valider nos travaux, nous avons souhaité conduire une évaluation perceptive sur les comportements de nos agents produits avec notre moteur décisionnel. Nous n'avons pas pu conduire une évaluation complète, et l'évaluation présentée ici est une évaluation préliminaire. Nous nous sommes concentrés dans cette évaluation préliminaire sur deux éléments uniquement (i.e. l'intégrité et la confiance en les compétences) pour générer les comportements des agents, que nous avons étudiés de manière indépendante.

Bien que cela soit insuffisant pour valider notre modèle, nous avons ainsi pu établir et tester une méthode d'évaluation qui pourrait être réutilisée pour conduire une évaluation complète.

Nous présentons dans un premier temps la procédure que nous avons mise en place pour conduire nos deux évaluations préliminaires. Nous détaillons ensuite chacune de ces deux évaluations.

7.3.1 Procédure

L'objectif de ces évaluations préliminaires est l'évaluation par des participants des comportements produits par notre système après leur observation. Toutefois, étant donné que notre moteur décisionnel n'a pas pu être couplé à un environnement virtuel, nous avons dû trouver un autre moyen de présenter les comportements de nos agents aux participants.

Présentation des comportements des agents aux participants

Nous avons choisi de produire des vidéos dans lesquelles des graphes sont dynamiquement tracés pour représenter les comportements de nos agents. Sur chaque graphe les actions que les agents peuvent faire sont associées à des lignes. Les agents sont représentés par des points de couleur dont la position évolue dans le temps pour désigner les actions que les agents sont en train de faire. Le trait de couleur représente l'historique des actions des agents. Le temps est représenté en abscisse du graphe.

Un exemple de graphe correspondant au scénario *SA* décrit en 7.3.2 est présenté dans la figure 7.2.

Déroulement des évaluations préliminaires

Nous avons recruté 12 participants pour ces évaluations préliminaires. Chacun des participants a d'abord été introduit à l'expérience à laquelle il allait participer avec une explication de son but et de son déroulement. Cette expérience s'est déroulée en deux phases :

- (1) *Lecture des consignes.* Le participant devait d'abord lire les consignes et les explications lui permettant de comprendre les graphes représentant les comportements des agents. Un exemple de graphe et l'explication associée étaient donnés au participant. Ces consignes peuvent être trouvées en annexe F.
- (2) *Visionnage des vidéos et réponse aux questionnaires.* Les participants observaient ensuite chaque vidéo et répondaient à des questions sur cette vidéo. Chaque vidéo pouvait être visionnée autant de fois que souhaité par le participant, qui pouvait aussi faire pause ou revenir en arrière.

Nous décrivons en 7.3.2 et en 7.3.3 les deux évaluations que nous avons conduites.

7.3.2 Première évaluation préliminaire

Nous décrivons ici la première évaluation préliminaire que nous avons conduite portant sur l'influence de l'intégrité des agents sur la perception de l'efficacité de l'équipe et de la contribution d'un agent à l'équipe.

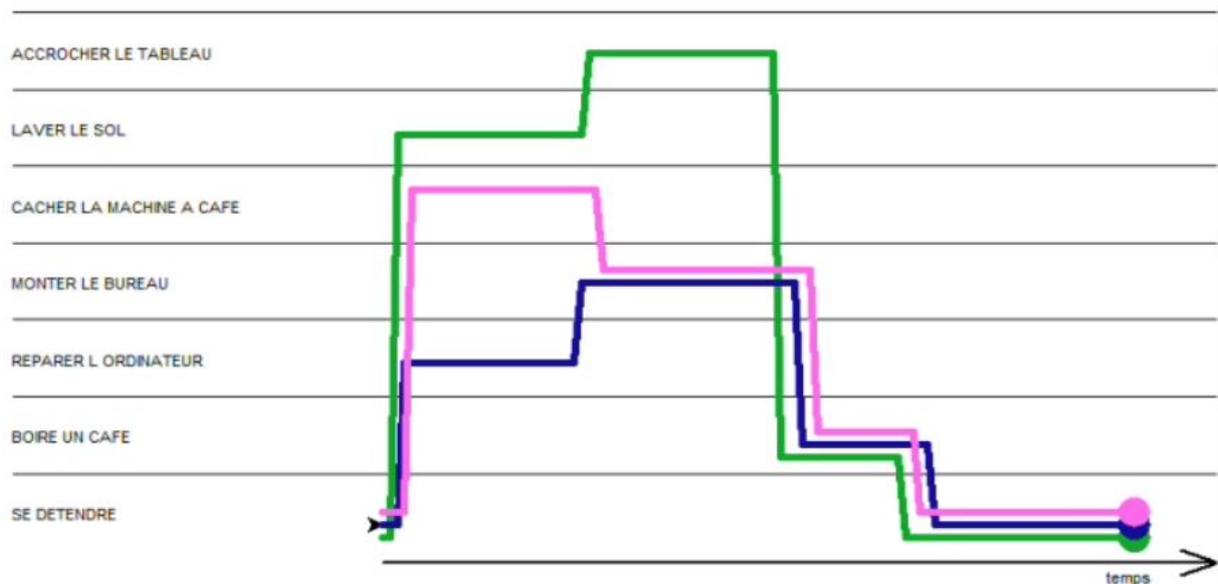


FIGURE 7.2 – Graphe dépeignant les comportements des agents Rose, Marine et Jade dans le scénario *SA*.

Le point rose et la ligne rose représentent respectivement ce qu'est en train de faire l'agent Rose et l'historique des actions de l'agent Rose. Le point vert et la ligne verte représentent respectivement ce qu'est en train de faire l'agent Jade et l'historique des actions de l'agent Jade. Le point bleu et la ligne bleue représentent respectivement ce qu'est en train de faire l'agent Marine et l'historique des actions de l'agent Marine.

Ce graphe correspond à la dernière image de la vidéo qui a été présentée aux participants lors de notre évaluation préliminaire.

Rangement de bureau – Scénarios produits

Le scénario proposé pour cette évaluation préliminaire met en scène trois agents Rose, Marine et Jade qui doivent ranger leur bureau. Ces trois agents ont donc un unique but collectif : celui de *ranger le bureau*. La tâche *ranger le bureau* est une tâche composée avec une contrainte ET/IND dont les sous-tâches sont les suivantes : *accrocher le tableau*, *laver le sol*, *monter le bureau*, *cacher la machine à café* et *réparer l'ordinateur*. Toutes ces sous-tâches correspondent à des actions qui doivent toutes être effectuées par une seule personne hormis l'action *monter le bureau* qui nécessite deux personnes.

Rose, Marine et Jade ont aussi le but personnel de *se détendre* et de *boire un café*. Ces deux buts correspondent respectivement aux actions *se détendre* et *boire un café*³.

Aucune compétence n'est nécessaire pour les tâches présentées ici.

Enfin, on considère que chaque agent est moyennement bienveillant envers les autres.

En faisant varier l'intégrité des agents, trois scénarios ont été produits à partir de la situation initiale que nous venons de décrire :

3. Pour modéliser cela nous avons créé deux tâches spécifiques pour chaque agent (e.g. tâche *se détendre* – *Marine*). En effet, ces tâches ne sont pas collectives (i.e. ne nécessitent pas que les agents soient synchronisés), mais peuvent être faites par chaque agent (voir 8.2.1).

(SA) Dans ce scénario, les agents sont tous plutôt intègres. Ils privilégient donc tous la réalisation du but collectif par rapport à leurs buts personnels. Les séquences d'actions produites sont alors les suivantes :

- À t_0 , Rose cache la machine à café, Marine répare l'ordinateur et Jade lave le sol.
- À t_1 , Rose et Marine montent le bureau pendant que Jade accroche le tableau.
- À t_2 , Rose, Marine et Jade boivent un café.
- Enfin à t_3 , Rose, Marine et Jade se détendent.

À l'issue de cette séquence d'actions, le but collectif est réalisé et les trois agents ont participé à sa réalisation.

(SB) Dans ce deuxième scénario, les agents Marine et Jade sont plutôt intègres, alors que l'agent Rose est peu intègre. Marine et Jade privilégient donc la réalisation du but collectif, alors que Rose privilégie ses buts personnels. Les séquences d'actions produites sont alors les suivantes :

- À t_0 , Marine cache la machine à café et Jade accroche le tableau alors que Rose boit un café.
- À t_1 , Marine répare l'ordinateur et Jade lave le sol, alors que Rose se détend.
- À t_2 , Marine et Jade montent le bureau alors que Rose se détend encore.
- À t_3 , Marine et Jade prennent un café alors que Rose se détend toujours.
- Enfin à t_4 , Marine, Rose et Jade se détendent.

À l'issue de cette séquence d'actions, le but collectif est réalisé mais seules Marine et Jade ont participé à sa réalisation.

(SC) Dans ce troisième scénario, tous les agents sont peu intègres, et tous privilégient donc la réalisation de leurs buts personnels. Les séquences d'actions produites sont alors les suivantes :

- À t_0 , Jade, Made et Rose prennent un café.
- À t_1 , Jade, Marine et Rose se détendent.

À l'issue de cette séquence d'actions, le but collectif n'est pas réalisé.

Pour chacun de ces scénarios, une vidéo dont la durée est de 16 secondes a été produite pour que les participants puissent observer le comportement des agents.

Ceux-ci ont dû lire une description textuelle des différentes actions que Rose, Jade et Marine pouvaient faire ainsi qu'une description de leur objectif collectif (i.e. que le bureau soit rangé) mais n'ont pas eu de description textuelle des scénarios comme nous venons de le faire ici, le déroulement du scénario étant dépeint dans la vidéo.

Hypothèses et questions aux participants

Nous avons souhaité évaluer par ces différents scénarios la manière dont étaient perçus les comportements des agents par les participants. En particulier, nous voulions, par notre moteur décisionnel, produire à la fois des comportements d'équipe idéaux et des comportements d'équipe non idéaux, de manière à confronter un apprenant à des situations non optimales. Nous nous sommes donc intéressés à deux éléments : (i) l'efficacité de l'équipe et (ii) la contribution d'un agent au bon fonctionnement de celle-ci ; les hypothèses suivantes ont été posées :

- **H1** : Une équipe dont les trois agents sont intègres sera perçue comme plus efficace qu'une équipe dont seulement deux agents sont intègres qui sera elle même perçue comme plus

TABLE 7.11 – Moyennes m et écarts types σ obtenus pour chacune des affirmations et chacun des scénarios de rangement de bureau.

	(i) A1	(ii) A2
<i>SA</i>	$m = 4.92, \sigma = 0.29$	$m = 4.83, \sigma = 0.39$
<i>SB</i>	$m = 3.17, \sigma = 1.19$	$m = 1.00, \sigma = 0.00$
<i>SC</i>	$m = 1.00, \sigma = 0.00$	$m = 1.17, \sigma = 0.58$

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.

Les éléments (i) et (ii) constituent donc les variables dépendantes de notre évaluation préliminaire et l'hypothèse H1 porte sur l'élément (i) alors que l'hypothèse H2 porte sur l'élément (ii).

Nous avons donc choisi d'évaluer le ressenti des participants par rapport à l'efficacité de l'équipe et à la contribution de l'agent Rose au bon fonctionnement de l'équipe. Une affirmation correspondant à chacun de ces éléments était présentée aux participants, qui devaient indiquer sur une échelle de Likert à cinq niveaux (de 1 – *pas du tout d'accord* à 5 – *tout à fait d'accord*) s'ils étaient d'accord avec cette affirmation. L'affirmation A1 correspondant à l'élément (i) était la suivante : *Les agents forment une équipe efficace* et l'affirmation A2 correspondant à l'élément (ii) était *L'agent Rose contribue au bon fonctionnement de l'équipe*.

Résultats

Le tableau 7.11 présente les résultats obtenus sur cette évaluation préliminaire en termes de score moyen et écart type des réponses des participants sur les affirmations A1 et A2 : on observe que l'équipe a été perçue par les participants comme très efficace dans le scénario *SA* ($m = 4.92$ et $\sigma = 0.29$). L'équipe a été perçue comme nettement moins efficace dans le scénario *SB* ($m = 3.17, \sigma = 1.19$) et comme pas du tout efficace dans le scénario *SC* ($m = 1.00, \sigma = 0.00$). L'agent Rose a quant à lui été perçu comme participant activement au bon fonctionnement de l'équipe dans le scénario *SA* ($m = 4.83, \sigma = 0.39$), mais comme ne participant pas du tout au bon fonctionnement de l'équipe dans les scénarios *SB* et *SC* ($m = 1.00, \sigma = 0.00$ et $m = 1.17, \sigma = 0.58$ respectivement).

Pour tester nos hypothèses H1 et H2 nous avons appliqué deux tests ANOVA à un facteur sur (i) l'efficacité perçue de l'équipe (affirmation A1) et (ii) sur la participation de l'agent Rose au bon fonctionnement de l'équipe (affirmation A2). Le facteur intra-sujet testé est dans les deux cas (i) et (ii) l'intégrité de l'ensemble des agents de l'équipe. Les résultats obtenus ((i) $F(2,33) = 62, p < 0.05$ et (ii) $F(2,33) = 349, p < 0.05$) montrent que les participants ont une appréciation significativement différente de (i) l'efficacité de l'équipe et (ii) la participation de Rose au bon fonctionnement de l'équipe dans les trois variantes du scénario proposées. Pour vérifier que les participants distinguent bien les variations de comportements des agents dans chacune de ces variantes, nous avons ensuite appliqué plusieurs comparaisons paire par paire avec le test de Bonferroni dont les résultats sont présentés dans le tableau 7.12.

TABLE 7.12 – Valeurs t obtenues par les tests de Bonferroni sur chaque combinaison de variantes et sur (i) l'affirmation A1 et (ii) l'affirmation A2.

	(i) A1	(ii) A2
<i>SA</i> et <i>SB</i>	t = 6.05	t = 23.36
<i>SA</i> et <i>SC</i>	t = 13.53	t = 22.34
<i>SB</i> et <i>SC</i>	t = 7.49	t = 1.0167

Les différences dans l'évaluation (i) du comportement de l'équipe par les participants sont donc fortement significatives pour toutes les variantes proposées. Les différences dans l'appréciation (ii) du comportement de l'agent Rose sont significatives dans les scénarios *SA* et *SB*, et *SA* et *SC* ; mais non significatives dans les scénarios *SB* et *SC*, ce qui était attendu puisque l'agent Rose a la même intégrité et se comporte de manière identique dans les deux variantes. Nos deux hypothèses H1 et H2 sont donc vérifiées. On peut néanmoins constater que l'efficacité perçue de l'équipe diminue significativement dès que l'un des agents n'est pas intègre.

7.3.3 Seconde évaluation préliminaire

Nous décrivons ici la seconde évaluation préliminaire que nous avons conduite portant sur la perception de la confiance d'un agent en les compétences des autres.

Sauvetage de victime – Scénarios produits

Pour cette évaluation, trois agents également sont mis en scène : un agent médecin, un agent infirmier et un agent ambulancier. Tous ces agents ont le but collectif de *sauver la victime* mais n'ont pas de buts personnels.

Les actions à faire pour *sauver la victime* sont les suivantes : *rassembler les informations* sur la victime puis *transmettre les informations* au poste de secours, et dans le même temps *faire les premiers soins* sur la victime, et *préparer l'ambulance* pour le transport. Une fois ces actions faites, il faut *transporter la victime* à l'hôpital. La représentation de cette activité sous forme d'arbre de tâches ACTIVITY-DL est donnée dans la figure 7.3.

Ces actions ne peuvent pas être faites par tous les agents :

- l'action *faire les premiers soins* ne peut être faite que par l'infirmier ou par le médecin.
- l'action *préparer l'ambulance* ne peut être faite que par l'ambulancier ou par le médecin.
- les actions *rassembler les informations* et *transmettre les informations* ne peuvent être faites que par le médecin.

Nous mettons en place ces restrictions sur les possibilités d'exécution d'action par l'introduction de compétences :

- L'action *faire les premiers soins* nécessite la compétence notée I. Les niveaux de compétences du médecin et de l'infirmier pour I sont de 1, alors que celui de l'ambulancier est de 0.
- L'action *préparer l'ambulance* nécessite la compétence notée A. Les niveaux de compétences du médecin et de l'ambulancier pour A sont de 1, alors que celui de l'infirmier est

de 0.

- Enfin, les actions *rassembler les informations* et *transmettre les informations* demandent la compétence M. Le niveau de compétence du médecin pour M est de 1 alors que ceux de l'infirmier et de l'ambulancier sont de 0.

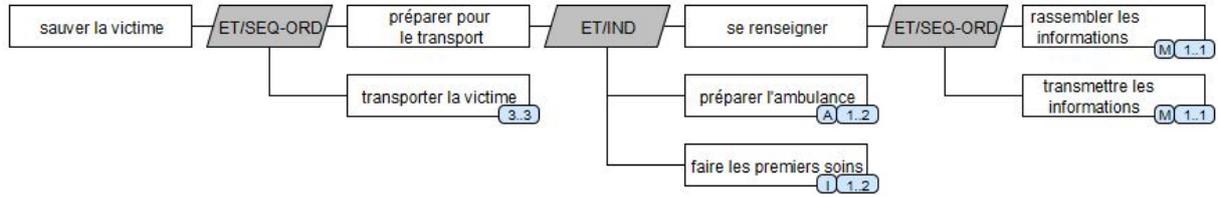


FIGURE 7.3 – Arbre de tâches ACTIVITY-DL représentant l'activité collective *sauver la victime*.

Nous considérons que les agents sont intègres : ils privilégieront leur but collectif. Nous considérons également que tous les agents sont moyennement bienveillants les uns envers les autres.

Nous avons produit deux scénarios différents avec ces agents et à partir de cette activité :

(SD) Dans ce scénario, le médecin a confiance en les compétences de l'infirmier et de l'ambulancier (i.e. il pense que l'infirmier est compétent pour I et que l'ambulancier est compétent pour A). Les séquences d'actions produites sont alors les suivantes :

- À t_0 , le médecin rassemble les informations pendant que l'infirmier fait les premiers soins et que l'ambulancier prépare l'ambulance.
- À t_1 , le médecin transmet les informations pendant que l'infirmier termine les premiers soins et que l'ambulancier termine de préparer l'ambulance.
- Enfin à t_2 , les agents transportent ensemble la victime à l'hôpital.

Dans ce scénario l'organisation des agents est optimale et la victime est rapidement transportée à l'hôpital. Ce scénario correspond à une situation de *confiance bien placée*, puisque les croyances du médecin sur les compétences de l'infirmier et de l'ambulancier sont exactes.

(SE) Dans ce scénario, le médecin n'a pas confiance en les compétences de l'infirmier et de l'ambulancier (i.e. il pense que l'infirmier n'est pas du tout compétent pour I et que l'ambulancier n'est pas du tout compétent pour A). Les séquences d'actions produites sont alors les suivantes :

- À t_0 , le médecin et l'ambulancier préparent ensemble l'ambulance, pendant que l'infirmier commence les premiers soins.
- À t_1 , l'action préparer l'ambulance est terminée. L'ambulancier ne peut plus rien faire et attend, pendant que le médecin aide l'infirmier pour terminer les premiers soins.
- À t_2 , les premiers soins sont terminés et l'infirmier ne peut plus rien faire. Il attend donc avec l'ambulancier pendant que le médecin rassemble les informations sur la victime.
- À t_3 , l'ambulancier et l'infirmier attendent toujours pendant que le médecin transmet les informations au poste de secours.
- Enfin à t_4 les agents transportent la victime à l'hôpital.

Nous avons vu dans le scénario *S5* un exemple de comportements produits lorsque la

confiance d'un agent en un autre était mal placée, dans le sens où un agent *surestimait* les compétences d'un autre. Nous avons vu que dans ce cas, les comportements générés n'étaient pas nécessairement optimaux. Ce scénario *SE* constitue un exemple de comportements produits lorsque la *confiance est mal placée*, dans le sens où un agent *sous-estime* les compétences d'un autre. Les comportements générés ne sont pas optimaux : l'équipe est mal organisée et plus de temps s'écoule avant que la victime ne soit transportée à l'hôpital.

Tout comme dans l'évaluation préliminaire précédente, les participants ont pu lire une description de l'activité que les agents avaient à faire mais aucune description textuelle des scénarios ne leur a été fournie : les participants ont observé deux vidéos de 15 secondes correspondant respectivement aux scénarios *SD* et *SE*.

Hypothèses et questions aux participants

Nous nous intéressons ici à la manière dont est perçu le comportement du médecin dans les deux scénarios, et en particulier, nous cherchons à savoir si les participants perçoivent le manque de confiance du médecin en ses coéquipiers dans le scénario *SE*. Nous faisons donc l'hypothèse suivante :

- **H3** : Les participants seront capables d'identifier le niveau de confiance d'un agent en les compétences d'un autre de par le comportement des agents.

Nous identifions donc ici deux variables dépendantes : (iii) le niveau de confiance perçu du médecin en l'infirmier et (iv) le niveau de confiance perçu du médecin en l'ambulancier. Ces deux variables serviront à tester l'hypothèse H3.

Une affirmation correspondant à chacun des éléments (iii) et (iv) était présentée aux participants, qui devaient indiquer sur une échelle de Likert à cinq niveaux (de 1 – *pas du tout d'accord* à 5 – *tout à fait d'accord*) s'ils étaient d'accord avec cette affirmation. L'affirmation A3 correspondant à l'élément (iii) était : *Le médecin a confiance en l'ambulancier* et l'affirmation A4 correspondant à l'élément (iv) était *Le médecin a confiance en l'infirmier*.

Résultats

Le tableau 7.13 présente les résultats obtenus sur cette évaluation préliminaire en termes de score moyen et écart type des réponses des participants sur les affirmations A3 et A4 : on peut observer que le médecin est perçu comme ayant fortement confiance en l'infirmier et en l'ambulancier dans le scénario *SD* ($m = 4.92$, $\sigma = 0.29$ dans les deux cas). En revanche dans le scénario *SE* le médecin est perçu comme ayant très peu confiance en l'ambulancier et moyennement confiance en l'infirmier ($m = 1.5$, $\sigma = 0.91$ et $m = 3.08$, $\sigma = 1.73$ pour les scénarios *SD* et *SE* respectivement). D'après les commentaires des participants, le fait que dans la seconde variante, la confiance du médecin en l'infirmier soit mieux jugée par les observateurs est dû au fait que le médecin aide d'abord l'ambulancier et ensuite l'infirmier.

Pour vérifier si notre hypothèse H3 est validée, nous avons appliqué des tests de Student sur (iii) la confiance perçue du médecin en l'ambulancier et (iv) la confiance perçue du médecin en l'infirmier. Les résultats ((iii) $p < 0.0001$ et (iv) $p = 0.004$) montrent que les différences d'appréciation entre les deux variantes sont significatives à $p < 0.01$. Notre hypothèse H3 est

TABLE 7.13 – Moyennes m et écarts types σ obtenus pour chacune des affirmations et chacune des variantes du scénario de sauvetage de victime

	(iii) A3	(iv) A4
<i>SD</i>	$m = 4.92, \sigma = 0.29$	$m = 4.92, \sigma = 0.29$
<i>SE</i>	$m = 1.5, \sigma = 0.91$	$m = 3.08, \sigma = 1.73$

donc vérifiée.

7.3.4 Conclusion

Ces deux évaluations sont préliminaires : elles sont trop simples pour constituer une évaluation complète de notre système qui permettrait de valider nos travaux. Elles nous ont néanmoins permis de mettre en place et de tester une méthode qui pourrait être appliquée pour une évaluation plus complète : il est ressorti de ces évaluations que les participants arrivent à lire correctement les comportements des agents tels que présentés sur les graphes.

Les résultats de ces évaluations préliminaires sont encourageants : les participants sont capables de percevoir le manque de confiance d'un agent en un autre au travers de ses comportements. De plus, les participants ont jugé dans le scénario *SA* que l'équipe était très efficace : dans ce scénario, les comportements des agents sont en effet optimaux du point de vue de l'équipe (i.e. réalisation des buts collectifs en priorité et bonne organisation des agents). À l'inverse, les participants ont jugé l'équipe comme peu efficace dès lors qu'un agent n'est pas intègre et ne participe pas à l'activité collective.

Il serait donc intéressant de poursuivre dans cette direction en proposant une évaluation plus poussée de notre système.

7.4 Bilan du chapitre

Nous nous sommes intéressés dans la section 7.1 de ce chapitre au fonctionnement pratique de notre moteur décisionnel. Nous avons dans un premier temps donné une brève description de l'architecture de notre système, puis nous avons abordé la question de la performance de notre système. Si cette performance dépend de nombreux facteurs, nos objectifs sont globalement atteints : notre système permet de générer en temps réel les comportements d'une dizaine d'agents sur un arbre de tâches de taille conséquente (i.e. plusieurs centaines de tâches). En effet, l'objectif final est d'intégrer un apprenant à une petite équipe d'agents virtuels dont les comportements seraient générés par notre système, et ce afin d'entraîner l'apprenant à prêter attention à chacun de ses coéquipiers lors d'activités collectives.

Nous avons ensuite présenté plusieurs scénarios d'exemple dans lesquels les comportements des agents sont générés avec notre système. Au travers de ces scénarios nous avons souligné l'importance de l'influence de plusieurs facteurs. (1) Nous avons vu en 7.2.1 que l'intégrité et la bienveillance d'un agent agissent sur l'importance que cet agent accorde à ses propres buts. Les agents étant dotés de théorie de l'esprit et tenant compte de chacun des autres agents

pour calculer l'importance qu'ils accordent à leurs buts, les variations de l'intégrité et de la bienveillance d'un seul agent influencent les comportements de toute l'équipe. Par ailleurs ces deux facteurs ne sont pas indépendants. (2) L'importance qu'un agent accorde à ses buts et les compétences des agents sont déterminantes pour le calcul de l'utilité qu'un agent accorde à une tâche, et par conséquent pour le choix de l'agent concernant la tâche à effectuer. Néanmoins les agents ne sélectionnent pas la tâche ayant la plus forte utilité pour eux : ils sélectionnent la tâche correspondant à la meilleure répartition des tâches selon eux. Pour prendre cette décision, les agents utilisent encore une fois leur théorie de l'esprit pour imaginer l'utilité que les tâches ont pour les autres. L'exemple donné en 7.2.3 montre que lorsque les agents accordent plus d'importance aux buts collectifs qu'à leurs buts personnels et lorsque leurs croyances sur les autres sont justes, notre moteur décisionnel leur permet d'obtenir des comportements optimaux du point de vue de l'équipe (i.e. réalisation des buts collectifs en priorité et bonne répartition des tâches). (3) Enfin, nous avons vu notamment au travers des scénarios *S2a* et *S5* que notre système permet de générer des comportements non-optimaux du point de vue de l'équipe mais cohérents et explicables du point de vue de chaque agent. De tels comportements peuvent être générés notamment en faisant varier l'intégrité des agents mais aussi en donnant aux agents des croyances fausses sur les autres : dans le premier cas, un agent non intègre choisit de manière volontaire de privilégier ses buts personnels, alors que dans le second cas, un agent ayant des croyances fausses sur les autres fait une erreur involontaire en prédisant les comportements des autres.

Finalement nous avons terminé ce chapitre en décrivant en section 7.3 deux évaluations préliminaires que nous avons conduites. Si ces évaluations sont insuffisantes pour valider les comportements générés par notre système, elles ont néanmoins permis de vérifier que l'efficacité perçue de l'équipe dépend de l'intégrité des agents de cette équipe, et un agent non intègre privilégiant ses buts personnels est perçu comme ne participant pas au bon fonctionnement de l'équipe. Les participants ont par ailleurs été capables de percevoir le manque de confiance d'un agent en les compétences des autres de par les comportements de celui-ci.

Le chapitre suivant est consacré à une conclusion générale sur les travaux menés au cours de cette thèse.

Chapitre 8

Conclusion générale

Nous avons présenté dans ce manuscrit les travaux menés au cours de cette thèse qui porte sur la génération de comportements d’agents virtuels dans des activités collaboratives en environnement virtuel de formation. En particulier, nous avons travaillé sur des activités collectives où les rôles des intervenants ne sont pas préétablis et où l’équipe doit s’auto-organiser dynamiquement. De plus, nous proposons de produire non pas des comportements d’équipe idéaux (i.e. réalisation des objectifs collectifs en priorité et bonne organisation des membres de l’équipe), mais des comportements non idéaux, de manière à préparer l’apprenant à faire face à ce type de situation et à l’entraîner à prêter attention à ses coéquipiers. Ce chapitre est consacré à une conclusion générale sur ces travaux.

Nous présentons d’abord un résumé de nos contributions dans la section 8.1 avant d’en aborder des limites dans la section 8.2. Enfin, nous terminons en présentant les perspectives de nos travaux en section 8.3.

8.1 Résumé des travaux

Nous avons présenté dans le chapitre 5 nos contributions portant sur l’ajout d’éléments au langage ACTIVITY-DL permettant la description d’activités collectives et sur la proposition de mécanismes de traitement des arbres de tâches ACTIVITY-DL de manière à générer des contraintes et informations sur lesquelles les agents pourront raisonner. Nous revenons sur ces contributions dans la section 8.1.1. Nous avons ensuite présenté dans le chapitre 6 nos contributions portant sur la proposition d’un moteur décisionnel permettant aux agents de raisonner sur les descriptions d’activités collectives et de générer, pour une équipe d’agents, des comportements cohérents, explicables et représentatifs de ceux observés sur le terrain. Nous résumons ces contributions section 8.1.2.

8.1.1 Description de l’activité et traitement des arbres de tâches

Nos travaux de thèse portent sur la génération de comportements collaboratifs pour des personnages virtuels autonomes. Ces comportements devant être représentatifs des comportements humains observés sur le terrain, nous nous sommes intéressés aux formalismes de description de l’activité. Parce que c’est un langage très expressif et parce qu’il permet une description hiérarchique de l’activité, nous avons choisi de travailler avec ACTIVITY-DL. Néanmoins ce langage

ne supportant la description des activités collectives, nous avons proposé d'ajouter la possibilité de spécifier, au niveau des actions, le nombre de personnes nécessaires et les compétences nécessaires pour faire ces actions.

Afin d'éviter que les agents aient à prendre en compte plusieurs centaines de tâches à la fois lorsqu'ils doivent en sélectionner une et de manière à ce que les agents puissent raisonner sur les tâches abstraites des arbres de tâches, nous avons proposé de propager certains éléments depuis les tâches feuilles de l'arbre jusqu'à la tâche racine. Ces éléments sont divisés en deux catégories : des contraintes et des informations. Les *contraintes* devront être vérifiées par l'équipe dans son ensemble pour que la tâche puisse être réalisée correctement. Ces contraintes comprennent (1) le nombre minimal de personnes nécessaires pour faire une tâche et (2) les compétences nécessaires pour réaliser une tâche ainsi que le nombre de personnes devant avoir chaque compétence. Les *informations* serviront à chaque agent pour déterminer s'il peut participer à une tâche : elles concernent (3) le nombre maximal de personnes pouvant participer, qui est utilisé par les agents pour générer des répartitions de tâches valides et (4) les compétences nécessaires pour participer à une tâche. La propagation de ces contraintes depuis les tâches feuilles jusqu'aux tâches racines n'est pas triviale, puisqu'elle est conditionnée à la fois par les contraintes logiques et temporelles attachées à chaque tâche abstraite. Par ailleurs, les contraintes et informations portent sur des éléments différents, n'ont pas le même sens et ne sont pas utilisées de la même manière par les agents. Les mécanismes permettant leur génération ne sont donc pas les mêmes.

Toutes nos contributions portant sur la description des activités collectives et le traitement des arbres de tâches ont été présentées dans le chapitre 5.

8.1.2 Moteur décisionnel

Nous avons proposé dans le chapitre 6 un moteur décisionnel reposant sur le modèle de confiance proposé par Mayer et al. (1995). Nous avons opérationnalisé ce modèle de confiance afin de définir les dimensions personnelles d'intégrité, de bienveillance et de compétence de chaque agent ainsi que les croyances de chaque agent sur les autres agents de l'environnement. Nous avons ensuite défini les mécanismes de prise de décision en nous appuyant sur ces dimensions personnelles et ces croyances.

Nous distinguons deux phases différentes lorsqu'un agent prend une décision : (1) dans la première phase de décision, les agents raisonnent sur leurs buts ainsi que sur les buts des autres agents alors que (2) dans la seconde phase de décision, les agents raisonnent sur les tâches de manière à en sélectionner une.

(1) Pour permettre aux agents de faire un choix informé quant aux objectifs à privilégier entre leurs buts personnels et les buts collectifs, nous avons proposé un mécanisme de calcul de l'importance des buts. Ce mécanisme permet de reproduire les effets d'influence du groupe sur la motivation de chaque individu : en se projetant à la place des autres, l'agent imagine l'importance que chaque autre agent accorde à ses buts et, en fonction de ce paramètre, module l'importance que lui-même accorde à ses buts. De cette manière, les agents peuvent adopter les buts des autres agents envers lesquels ils sont bienveillants, ce qui constitue une opérationnalisation du mécanisme d'adoption altruiste proposé par Castelfranchi and Falcone (2010).

(2) Les agents raisonnent ensuite sur les tâches de manière à en sélectionner une. Pour cela, l'utilité d'une tâche est calculée comme le produit de l'importance du but auquel la tâche

contribue et de la compétence de l'agent sur la tâche en question. L'agent ne sélectionne pas la tâche ayant la plus forte utilité : il calcule l'utilité qu'ont les tâches pour chaque autre agent selon lui, et sélectionne la répartition de tâches dont l'utilité est la plus élevée. Ce faisant, l'agent anticipe le comportement des autres, permettant ainsi à l'équipe de s'auto-organiser et aux comportements collaboratifs d'émerger.

Que ce soit pour le calcul de l'importance que les autres agents accordent à leurs buts ou celui de l'utilité des tâches pour les autres agents, un agent raisonne sur les croyances qu'il a vis-à-vis des autres. Si ses croyances sont justes, ses prédictions sur les comportements des autres seront justes, mais si ses croyances ne sont pas justes, alors ses prédictions seront fausses, ce qui peut nuire à l'organisation de l'équipe et entraîner des comportements d'équipe non idéaux du point de vue de l'équipe mais cohérents et explicables du point de vue de l'agent. Une autre manière de générer des comportements non idéaux du point de vue de l'équipe est de faire en sorte qu'un agent (ou plusieurs) privilégie ses buts personnels : ce sera sûrement le cas si l'agent n'est pas intègre.

Nous avons par ailleurs décrit l'implémentation de notre système dans le chapitre 7 avant de présenter des scénarios illustrant les comportements qu'il est possible de produire avec notre moteur décisionnel. Nous avons présenté dans ce même chapitre deux évaluations perceptives préliminaires insuffisantes pour valider nos travaux mais dont les résultats sont encourageants : en jouant sur l'intégrité des agents nous avons pu produire des équipes jugées comme plus ou moins efficaces par des observateurs extérieurs, et les participants ont été capables de percevoir le manque de confiance d'un agent en les compétences d'un autre d'après les comportements adoptés.

La section suivante est consacrée aux limites de nos travaux.

8.2 Limites

Nos travaux ne sont pas exempts de limites. Nous présentons dans la section 8.2.1 les limites liées à la description de l'activité ainsi qu'aux mécanismes de traitement de l'activité proposés. La section 8.2.2 est consacrée aux limites liées à notre modèle d'agent et à notre moteur décisionnel. Enfin nous décrirons dans la section 8.2.3 les limites liées à l'implémentation proposée.

8.2.1 Limites liées à la description et au traitement de l'activité

La description de l'activité est un problème difficile sur lequel de nombreux travaux se sont penchés. Bien que nos propositions aient permis de répondre à notre problème, nous pouvons mettre en évidence certaines limites que nous développons dans les paragraphes suivants.

Activités collectives et actions individuelles

Nous nous sommes intéressés aux activités collectives, et nous avons proposé qu'une action puisse être collective : l'action nécessite alors plus d'une personne pour être réalisée, et les personnes qui réalisent l'action doivent être synchronisées. Dans les activités collectives, certaines actions doivent ou peuvent être faites de manière individuelle (e.g. *installer l'ordinateur* dans le scénario développé chapitre 6 pour ne citer que cet exemple) : elles nécessitent au minimum une

personne pour être exécutées. La personne qui les exécute le fait alors *pour l'équipe* (i.e. une fois l'action terminée, elle l'est pour toute l'équipe).

Cependant, nous avons constaté lors de notre travail sur différents cas d'usage et scénarios d'exemple que certaines actions sont intrinsèquement individuelles bien qu'elles puissent être faites par plusieurs personnes : c'est le cas par exemple de l'action *boire un café*, que nous avons utilisée dans la première évaluation préliminaire présentée en 7.3.2. En effet, lorsque Rose boit un café, le café n'est pas bu par l'équipe, mais uniquement par Rose. On peut imaginer un autre exemple dans le domaine professionnel : si une équipe médicale doit intervenir sur un patient, tous les membres de l'équipe doivent se *laver les mains*. Pourtant cette action n'est pas collective au sens des actions collectives de ACTIVITY-DL : aucune synchronisation n'est nécessaire entre les membres de l'équipe pour se laver les mains.

Pour traiter de telles actions, il faudrait pouvoir spécifier dans la description de l'activité si une action se rapporte à l'équipe ou un à individu uniquement. Ajouter cette possibilité nécessiterait alors de modifier les règles de propagations de contraintes et d'informations présentées dans le chapitre 5.

Compétences et description de l'activité collective

Nous avons dans nos travaux imposé une contrainte forte sur la possibilité de spécifier des compétences nécessaires à une action : une seule compétence peut être associée à une action, et nous considérons alors cette compétence comme obligatoire. Or, on pourrait distinguer les compétences qui sont nécessaires de celles qui sont facultatives et qui faciliteraient l'exécution de l'action. On pourrait pour cela exploiter au mieux la richesse du langage ACTIVITY-DL qui propose différents types de pré-conditions : les compétences nécessaires pourraient être précisées en tant que *pré-conditions règlementaires* (i.e. pré-conditions qui correspondent aux états du monde nécessaires à la bonne réalisation de la tâche selon la procédure (Barot, 2014, p. 126)). Toutefois aucun des quatre types de pré-conditions déjà existants et définis dans ACTIVITY-DL ne permet d'exprimer le caractère facultatif d'une compétence pour l'exécution d'une tâche.

De même, dans notre proposition, chaque compétence est associée directement à une action. On pourrait adopter un formalisme de description plus permissif où il serait possible de préciser que pour l'action *intuber la victime*, il faut *une personne ayant la compétence "médecin" OU deux personnes ayant la compétence "infirmier"*.

Par ailleurs, nous avons également ignoré dans nos travaux le fait que certaines tâches peuvent être facultatives (marqueur `optional`) prédéfini dans ACTIVITY-DL).

Tous ces éléments pourraient être ajoutés pour étendre le modèle et être pris en compte lors de la génération des contraintes de faisabilité collective et des informations dynamiques associées à chaque tâche.

Spécifications structurelle et déontique

Nous avons considéré des équipes d'agents dans lesquelles chaque agent a autant de pouvoir que les autres (i.e. pas de "chef") et où il n'y a pas de rôles prédéfinis. Cependant, pour reproduire les comportements humains lors de certaines activités comme celles du sauvetage de victimes, il peut être intéressant d'intégrer ces éléments. Cela pourrait être fait au travers de l'ajout de

spécifications structurelle et déontique (Hubner et al., 2007). On pourrait alors définir pour les agents des droits et des obligations, comme cela est proposé par Hubner et al. (2007) dans le domaine des SMA ou par Gerbaud et al. (2007) dans le domaine des environnements virtuels de formation.

Il serait également intéressant d'étudier la notion de responsabilité : par exemple, lors de l'intervention d'une équipe de pompiers pour le sauvetage de la victime, normalement le chef de groupe a la responsabilité de *couper le compteur*. Cette personne a la possibilité de déléguer cette action, mais elle devrait alors vérifier qu'elle a été faite correctement (e.g. demander *As-tu bien coupé le compteur ?* ou aller voir si le compteur est bien coupé). En outre, la confiance étant fortement liée à la délégation (voir 4.1.2), l'ajout de cet élément paraîtrait particulièrement pertinent par rapport à nos travaux.

8.2.2 Limites liées à notre moteur décisionnel

Notre moteur décisionnel permet à chaque agent de sélectionner une action à faire en fonction de son intégrité, sa bienveillance, ses compétences et ses croyances sur les autres. Néanmoins, nous n'avons pas proposé de mécanismes de mise à jour des croyances, et nous n'avons pas intégré de notion de degré de croyance. Ces limites sont détaillées dans les paragraphes suivants.

Mise à jour des croyances

Nous avons dans ces travaux proposé un moteur décisionnel pour des personnages virtuels autonomes qui s'appuie en grande partie sur les croyances qu'ont chacun des agents sur les autres. Ces croyances doivent être dynamiques et évoluer au cours du temps : nous avons vu dans l'un des scénarios qui nous ont servi à proposer une évaluation préliminaire de notre moteur décisionnel que l'agent médecin n'a pas confiance en la compétence des agents infirmier et ambulancier, alors que ces agents sont compétents. Lorsque le médecin voit que l'infirmier s'occupe correctement du malade, sa croyance sur la compétence de l'infirmier devrait changer.

Or nous n'avons pas proposé dans cette thèse de mécanisme de mise à jour des croyances, ce qui est une limite non négligeable à nos travaux. En effet, la non mise à jour des croyances entrainera des comportements qui pourront paraître incohérents sur des scénarios qui dureront plus longtemps dans le temps : le médecin continuera à aider l'infirmier alors que ce dernier n'a, de toute évidence, pas besoin d'aide.

Degré de croyances

Nous n'avons pas pris en compte dans nos travaux les degrés de croyances : un agent est toujours sûr de son jugement sur l'autre. Or en réalité, et comme proposé dans les modèles de confiance cognitifs BDI+Repage, C&F et ForTrust, les croyances doivent être associées à des degrés de croyances. En effet, on est parfois *peu sûr* de nos jugements ou, à l'inverse, on est *certain* de ce que l'on pense. Lorsque les croyances sont considérées comme des certitudes, comme nous l'avons fait dans nos travaux, on devrait parler de *connaissances*.

La non prise en compte des degrés de croyance constitue donc une seconde limite à notre proposition de moteur décisionnel reposant sur la confiance. Cependant, cette seconde limite est fortement liée à la première : ces degrés de croyances devraient être mis à jour lorsqu'un agent

observe un autre faire une action dans l'environnement. Sans cette mise à jour il n'y a pas de sens à ajouter ces degrés de croyances au modèle d'agent.

8.2.3 Limites liées à l'implémentation proposée

Nous avons présenté dans le chapitre 7 l'implémentation de notre système. Si cette implémentation nous a permis de vérifier le bon fonctionnement de notre moteur décisionnel ainsi que de conduire une évaluation préliminaire, elle présente également des limites, relatives notamment au fonctionnement des agents. Nous présentons ici ces limites.

Impossibilité pour les agents de changer d'avis

Il est impossible pour un agent dans notre système de changer d'avis, aussi bien sur l'action choisie que sur l'importance qu'il accorde à ses buts. En effet, l'importance des buts est calculée une seule fois en début de simulation. Pendant la simulation, les agents ne reconsidèrent jamais l'importance qu'ils accordent à leurs buts. Cela peut être gênant principalement pour deux raisons. (1) Nous avons précisé dans la section 6.3 que les formules permettant de calculer l'importance des buts permettaient, pour des différents niveaux de motivation de plusieurs agents pour un même but, de converger. Or en ne calculant qu'une seule fois l'importance des buts, l'écart entre les niveaux de motivation est réduit mais les niveaux de motivation ne convergent pas. (2) Les agents peuvent avoir en début de simulation des croyances sur les autres qui s'avèrent fausses. L'importance accordée aux buts devrait alors être recalculée. Ce second point est cependant très fortement lié à une autre limite de notre moteur décisionnel dont nous avons déjà discuté : cela suppose l'existence d'un mécanisme de mise à jour des croyances. Si un tel mécanisme était mis en place, il serait alors intéressant de recalculer l'importance des buts à intervalles de temps réguliers en cours de simulation.

Dans le même ordre d'idées, une fois une action sélectionnée, un agent ne peut pas changer d'avis et décider de faire autre chose tant que l'action n'est pas terminée (à moins qu'il n'y ait échec de l'exécution ou que le seuil d'attente soit dépassé – voir 7.1.1). Cela peut être irréaliste surtout si l'action est longue. Il faudrait qu'en cours d'exécution l'agent puisse reconsidérer son choix (e.g. par rappel de la fonction `decide`) : l'agent pourrait alors commencer une action, changer d'avis et faire autre chose puis, éventuellement, revenir sur l'action commencée et la terminer.

"Blocage" des agents lorsqu'ils attendent

Lorsque les agents choisissent une action collective et qu'ils doivent attendre que d'autres la choisissent également pour être assez nombreux et ainsi pouvoir exécuter l'action, les agents n'ont pas la possibilité, dans l'implémentation actuelle, de faire autre chose en attendant (voir 7.1.1) et ils sont "bloqués" en attente. Cela peut nuire à la crédibilité des comportements produits puisque cela revient à avoir des agents qui ne font rien pendant un certain temps même si d'autres actions sont à faire.

D'autres types de comportements pourraient être mis en place pour éviter ces situations : les agents pourraient faire autre chose en attendant, puis dès que suffisamment de personnes

choisissent l'action collective, interrompre ce qu'ils ont commencé pour exécuter l'action collective. Les agents pourraient également avoir la possibilité de demander de l'aide aux autres de manière à signaler qu'ils attendent pour qu'éventuellement d'autres agents interrompent leurs actions pour participer à l'action collective.

La mise en place de ces mécanismes supposerait cependant que les agents aient la possibilité d'interrompre leurs actions, aient la possibilité de communiquer voire même de négocier à propos du meilleur moment pour faire l'action collective.

8.3 Perspectives

Nous présentons ici des perspectives de travaux futures liées aux propositions développées dans cette thèse. Certaines de ces perspectives sont liées aux limites dont nous venons de discuter. D'autres sont des questions de recherche plus générales qui pourraient être abordées en lien avec nos travaux.

8.3.1 Perspectives à court terme

Intégration de degrés de croyances et de mécanismes de mise à jour des croyances et des degrés de croyances associés

Il pourrait être relativement simple d'étendre notre modèle pour intégrer la notion de degrés de croyances : on pourrait s'inspirer de la littérature et utiliser des valeurs floues pour représenter les croyances sur l'intégrité, la bienveillance et les compétences comme cela est fait dans BDI+Repage, ou l'on pourrait simplement associer chaque croyance à une valeur de certitude comme cela est fait dans ForTrust. Les mécanismes de prise de décision seraient alors modifiés. Par exemple, si l'on utilise une valeur de certitude comme proposé dans ForTrust, dans chaque calcul où apparaissent des valeurs de croyance d'un agent sur l'intégrité, la bienveillance ou les compétences d'un autre, ces valeurs seraient pondérées par le degré de certitude qui leur serait associé.

De même, il serait possible de faire en sorte que la croyance du médecin sur la compétence de l'infirmier soit mise à jour de manière directe lorsque le médecin observe l'infirmier s'occuper du blessé. Cependant la mise à jour des croyances n'est pas toujours si simple. Par exemple, si l'on considère le scénario illustratif présenté dans le chapitre 6, l'agent Béa choisit de faire l'action *écrire un papier* qui correspond à l'un de ses buts personnels plutôt que d'œuvrer pour la réalisation du but collectif. Pour un observateur extérieur, ce comportement peut être interprété de plusieurs façons différentes : (i) Béa n'est pas intègre et accorde donc plus d'importance à son but personnel qu'au but collectif, (ii) Béa n'est pas compétente en informatique et ne peut donc pas participer à la réalisation du but collectif ou (iii) Béa est moyennement intègre mais elle pense que les autres ne le sont pas et ne veut donc pas être la seule à participer à l'activité collective.

Ces différentes interprétations possibles pourraient alors justement être traitées avec les degrés de croyance : on peut associer un degré de croyance de 60% à l'explication (i) (i.e. on considère que la probabilité que le comportement de Béa s'explique par l'hypothèse (i) est de 60%), de 30% à l'explication (ii) et de 10% à l'hypothèse (iii). Il faudrait alors définir la manière

dont sont mises à jour les croyances et le degré de certitude qui leur est associé à la lumière de ces explications et de leur degré de certitude.

Par ailleurs, si les agents disposent d'un mécanisme de mise à jour des croyances, l'importance qu'ils accordent à leurs buts devrait être recalculée à chaque prise de décision, ce qui n'est pas le cas dans l'implémentation actuelle.

Enfin, nous avons précisé en 4.3.1 que le modèle de confiance proposé par Mayer et al. (1995) comprend une dimension propre au trustor : celle de la disposition à faire confiance. Nous n'avons pas intégré cette dimension lors de la définition de nos agents et de leurs croyances. En effet, la disposition du trustor à faire confiance influence sa décision particulièrement lorsqu'il ne connaît pas ou connaît peu le trustee. Nos agents ayant des certitudes sur les autres agents de l'environnement, il n'était pas pertinent de tenir compte de la disposition à faire confiance. Intégrer des degrés de croyances permettrait de modéliser le fait qu'un agent connaît peu ou ne connaît pas un autre agent, et il serait alors intéressant d'intégrer la disposition à faire confiance dans les dimensions de nos agents.

8.3.2 Perspectives à long terme

Planification et prise en compte de la temporalité

Nous avons dans nos travaux complètement ignoré les aspects temporels. Nos agents ne prennent pas en compte les aspects liés à la durée des actions : les actions sont toujours considérées de la même manière quelque soit le temps qu'elles peuvent nécessiter. Il en va de même pour les tâches composées qui sont toujours considérées de la même manière quelque soit le nombre de sous-tâches qu'elles contiennent. Cette prise en compte pourrait cependant se faire de façon relativement simple : à la manière des compétences et des nombres de personnes nécessaires, la durée des actions pourrait être précisée sur les tâches feuilles de l'arbre de tâches et propagée sous forme d'information dynamique aux tâches composées. La durée de chaque tâche entrerait alors en compte dans le calcul de l'utilité des tâches.

Nous avons également omis dans ces travaux les problèmes liés à la planification. En effet, un agent n'a pas de vue d'ensemble des séquences d'actions qu'il va exécuter. Nous avons en partie contourné ce problème par l'utilisation de contraintes et d'informations qui sont propagées depuis les tâches feuilles jusqu'aux tâches racines des arbres de tâches, assurant ainsi que les agents ne se retrouvent pas dans une situation de blocage. Cependant cette solution n'est pas complètement satisfaisante, en particulier car elle ne permet pas un partage équitable des tâches entre les agents sur le long terme, notamment car toutes choses égales par ailleurs, c'est toujours l'agent le plus compétent qui fait la tâche. Par exemple si la compétence κ est nécessaire pour faire chacune des actions de la séquence s , et que deux agents Anna et Béa, tels que le niveau de compétence d'Anna sur κ est meilleur que celui de Béa, doivent se répartir les tâches, c'est Anna qui fera entièrement la séquence d'actions s puisque c'est elle la plus compétente. En revanche, si s n'avait pas été une séquence et que les actions avaient pu être faites en parallèle, Anna et Béa se seraient réparti les tâches. Des mécanismes de planification permettraient notamment de mieux gérer la répartition des tâches.

Néanmoins, un autre problème se pose en lien avec l'intégration de mécanismes de planification : notre moteur décisionnel permet aux agents de se projeter à la place de chacun des autres

pour imaginer la meilleure répartition possible de tâches à l'instant t . Si les agents planifient à l'avance non plus une seule action mais une séquence d'actions, pour que le moteur décisionnel fonctionne en temps réel, il n'est pas réaliste que les agents se projettent à la place de tous les autres pour imaginer toutes les séquences d'actions possibles. En effet, nous sommes déjà limités à un petit nombre d'agents du fait que le nombre de calculs à faire augmente de manière exponentielle lorsqu'on ajoute un agent à l'environnement. Ce nombre de calcul augmenterait également de manière exponentielle si les agents se projetaient sur une séquence de deux actions plutôt que sur une seule action.

Néanmoins ce procédé correspondrait de manière plus juste à la définition de la théorie de l'esprit, et une architecture comme PsychSim (Marsella et al., 2004) pourrait être utilisée.

Intégration de mécanismes de communication et de négociation

Nous nous sommes concentrés dans cette thèse sur les comportements liés à l'organisation implicite d'une équipe. À l'inverse des systèmes tels que STEAM qui sont conçus pour assurer une organisation optimale de l'équipe et qui reposent en grande partie sur la communication entre les agents, nous avons souhaité ici reproduire des comportements humains et non optimaux, où chacun prend ses décisions en faisant des suppositions sur les comportements des autres, ces suppositions n'étant pas toujours justes. Si une organisation toujours parfaitement optimale n'est pas réaliste, une équipe dans laquelle les agents ne communiquent jamais ne l'est pas non plus.

Il serait donc intéressant d'intégrer à notre système des mécanismes de communication et plus encore des mécanismes de négociation qui permettraient aux agents de négocier sur une répartition des tâches à adopter. De plus, nous avons considéré les buts comme statiques : lorsqu'un agent a un but, il le conserve et ne peut pas changer d'avis sur le but à accomplir. De la même manière que pour la répartition des tâches à adopter, définir en amont de la réalisation de l'activité une phase de discussion et de négociation des buts à atteindre par l'équipe (et/ou par des sous-équipes) serait une amélioration possible de notre système. On pourrait par exemple s'inspirer des nombreux travaux qui portent sur la négociation, aussi bien en SHS (Brett, 2000; Eden and Ackermann, 2001; Thompson et al., 2010) qu'en informatique (Chu-Carroll and Carberry, 1994, 2000; Core et al., 2006; Traum et al., 2003, 2005, 2008).

L'une des principales difficultés liée à l'intégration de tels mécanismes serait de savoir *quand* il faudrait les utiliser, puisque pour obtenir des comportements proches de ceux des humains, il ne faudrait pas remplacer complètement le processus d'organisation implicite par une organisation explicite au travers de la communication et de la négociation mais pouvoir faire coexister les deux.

Par ailleurs, Butler (1999) a démontré l'influence de la confiance sur la négociation : les mécanismes de négociation définis pourraient s'appuyer sur notre modèle de confiance (i.e. les comportements de négociation générés seraient influencés par la confiance qu'a chaque agent en l'agent avec lequel il négocie) et une dynamique de la confiance pourrait être définie en fonction des propositions de chaque agent.

Intégration à un environnement virtuel de formation et évaluation sur un scénario réel avec un apprenant

Nos travaux n'ont pas pu être intégrés à un environnement virtuel de formation, bien que ce soit l'objectif final visé : par exemple pour l'évaluation préliminaire proposée nous avons simplement présenté aux sujets des graphes dépeignant le comportement des agents. Il serait donc intéressant de poursuivre dans cette direction en utilisant notre moteur décisionnel pour diriger les comportements de personnages virtuels autonomes dans un environnement virtuel de formation.

Si le couplage de notre système à un environnement virtuel de formation demanderait un travail non négligeable, la problématique principale serait celle de la prise en compte de l'apprenant par nos agents. Par exemple, l'apprenant pourrait être considéré par notre système comme un autre agent quelconque sur lequel nos agents auraient des croyances. La difficulté viendrait cependant du fait que, si dans notre système les agents partagent tous une même représentation de l'activité, l'apprenant n'a pas nécessairement la même représentation de cette activité. Les comportements de l'apprenant qui ne correspondraient pas exactement à l'arbre de tâches sur lequel raisonnent les agents ne seraient alors pas compris par ces derniers. Si l'on dispose d'un mécanisme de mise à jour des croyances des agents, leurs croyances sur l'apprenant risquent d'être biaisées par des différences de conception quant à la manière de réaliser la tâche.

Par ailleurs, dans un contexte de formation, le paramétrage des agents en amont de la session de formation demande un certain effort : il peut être difficile de trouver les valeurs d'intégrité, de bienveillance et de compétences ainsi que les valeurs correspondant aux croyances des agents sur les autres qui permettront d'obtenir les comportements souhaités en vue de produire un scénario qui répondra à certains objectifs pédagogiques précis.

L'utilisation de notre moteur décisionnel dans un environnement virtuel de formation serait cependant l'opportunité d'évaluer les comportements des agents dans le contexte d'utilisation finale et sur un scénario réel, plus complexe et plus complet que celui proposé dans l'évaluation préliminaire que nous avons menée.

Annexes

Annexe A

Publications

- *From Individual Cognition to Collective Intelligence* – Lucile Callebert, 14th International Conference on Intelligent Virtual Agents (IVA), 2014 (doctoral consortium).
- *Relations de confiance dans un groupe d’agents* – Lucile Callebert, Domitile Lourdeaux, Jean-Paul Barthès, 23ème édition des Journées Francophones des Systèmes Multi-Agents, 2015 (poster).
- *A Trust-Based Decision-Making Approach Applied to Agents in Collaborative Environments* – Lucile Callebert, Domitile Lourdeaux, Jean-Paul Barthès, 8th Conference on Agents and Artificial Intelligence (ICAART), 2016.
- *Trust-Based Decision-Making System for Action Selection by Autonomous Agents* – Lucile Callebert, Domitile Lourdeaux, Jean-Paul Barthès, IEEE 20th International Conference on Computer Supported Cooperative Work in Design (CSCWD), 2016.
- *Moteur décisionnel reposant sur un modèle de confiance pour des agents autonomes* – Lucile Callebert, Domitile Lourdeaux, Jean-Paul Barthès, Rencontre des Jeunes Chercheurs en Intelligence Artificielle (RJCIA), 2016.
- *Priority-based contextual local decision making in multi-agent systems* – Jean-Paul Barthès, Lucile Callebert, Domitile Lourdeaux, IEEE 20th International Conference on Computer Supported Cooperative Work in Design (CSCWD), 2016.

Annexe B

Représentation de l'activité

B.1 Agrandissements

Les figures B.1, B.2, B.4, B.5, B.6 et B.7 sont des agrandissements de celles présentées dans le chapitre 5.

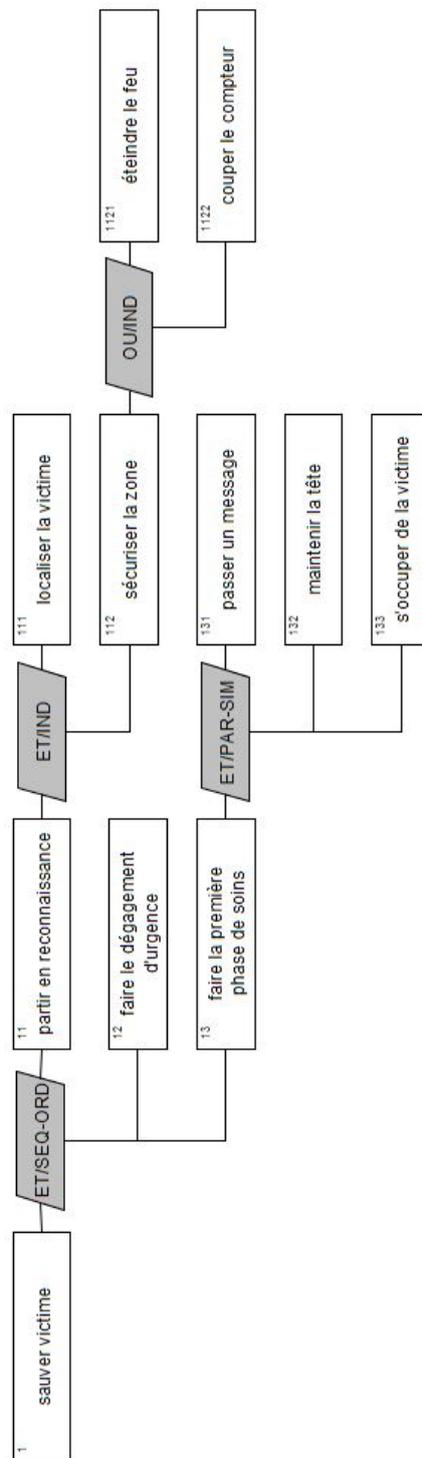


FIGURE B.1 – Exemple d’un arbre de tâches décrit en ACTIVITY-DL. Cette figure est un agrandissement de la figure 5.1 présentée page 76.

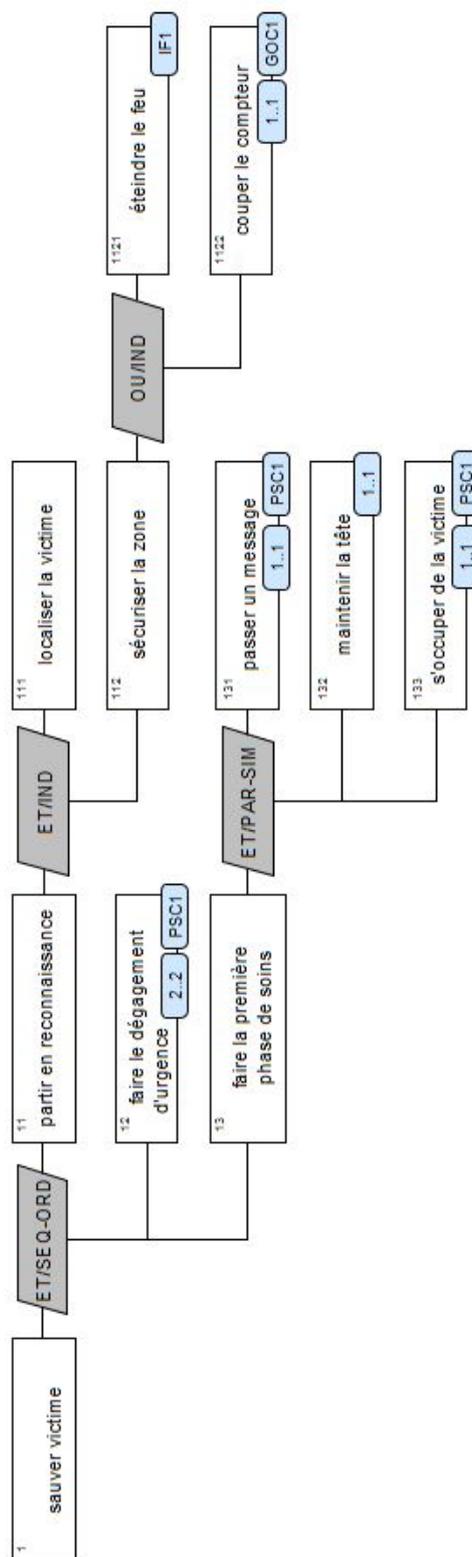


FIGURE B.2 – Même exemple que dans la figure 5.1. Ici l'arbre de tâches décrit en ACTIVITY-DL comporte, dans les bulles bleues en bas à droite de chaque tâche feuille, les éléments de nombre de personnes et de compétences renseignés par les experts du domaine. Cette figure est un agrandissement de la figure 5.2 présentée page 80.

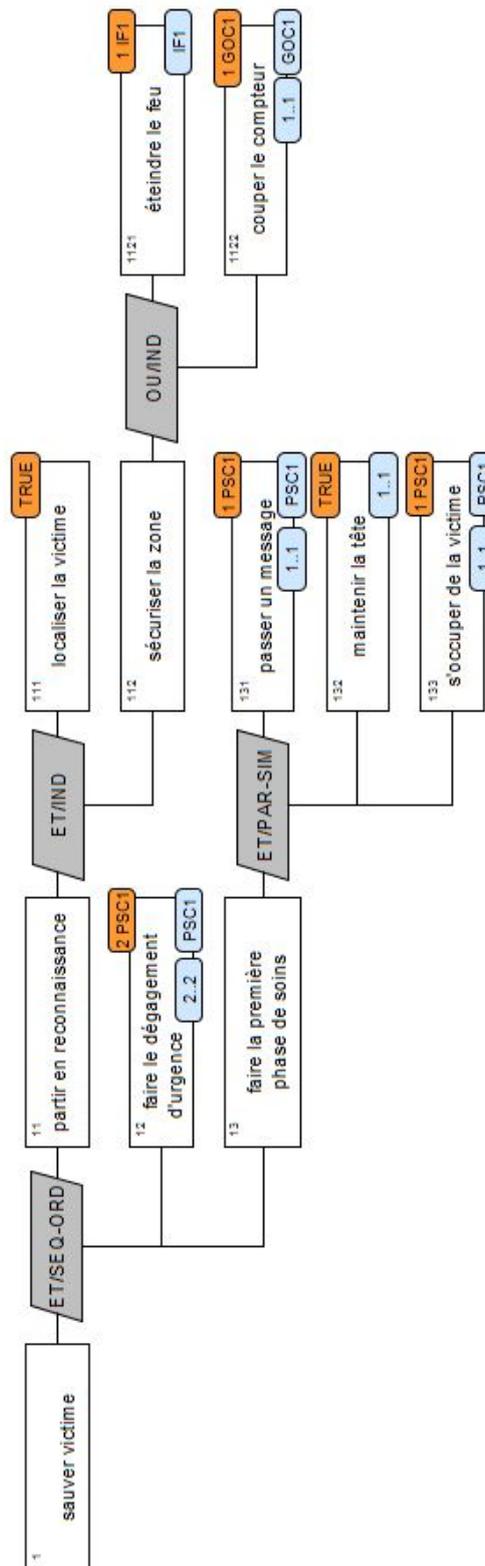


FIGURE B.3 – Même arbre de tâches que celui présenté en figure 5.2 comportant (i) dans les bulles bleues en bas à droite de chaque tâche feuille, les éléments de nombre de personnes et de compétences renseignés par les experts du domaine; et (ii) dans les bulles orange en haut à droite de chaque tâche feuille, les contraintes de faisabilité collective en termes de compétences associées aux tâches feuilles. *Cette figure est un agrandissement de la figure 5.3 présentée page 84.*

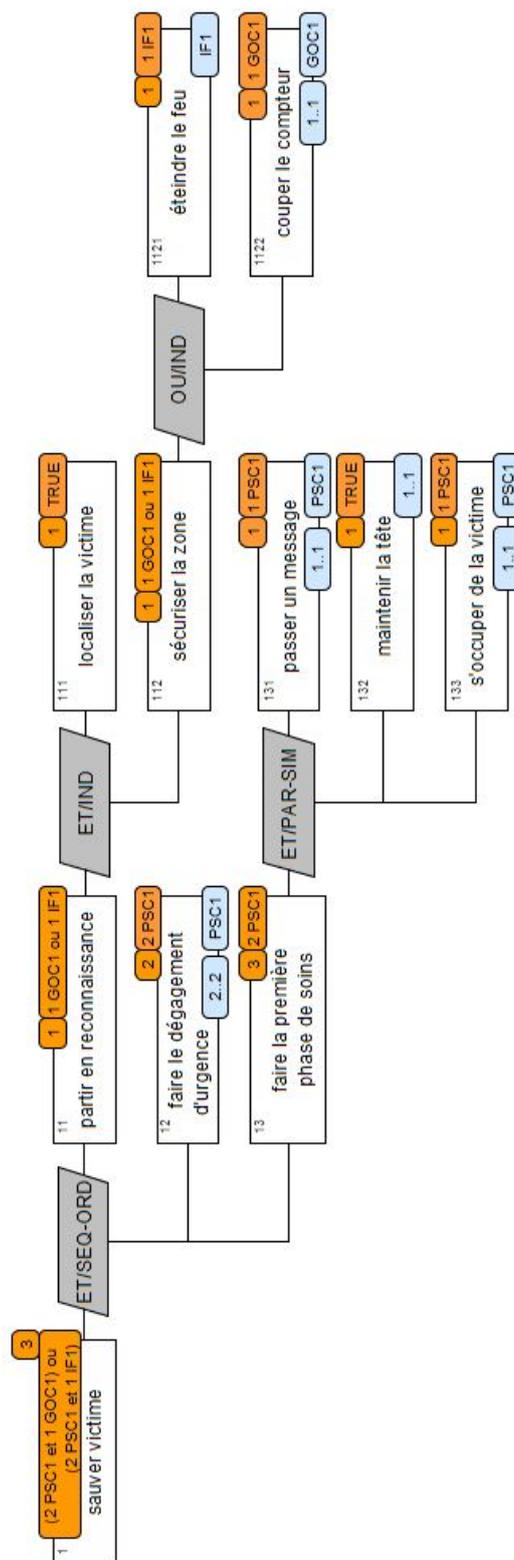


FIGURE B.4 – Même arbre de tâches que celui présenté en figure 5.2 sur lequel apparaissent, dans les bulles orange en haut à droite de chaque tâche, les contraintes de faisabilité collective en termes de nombre minimal de personnes et de compétences nécessaires. Cette figure est un agrandissement de la figure 5.4 présentée page 90.

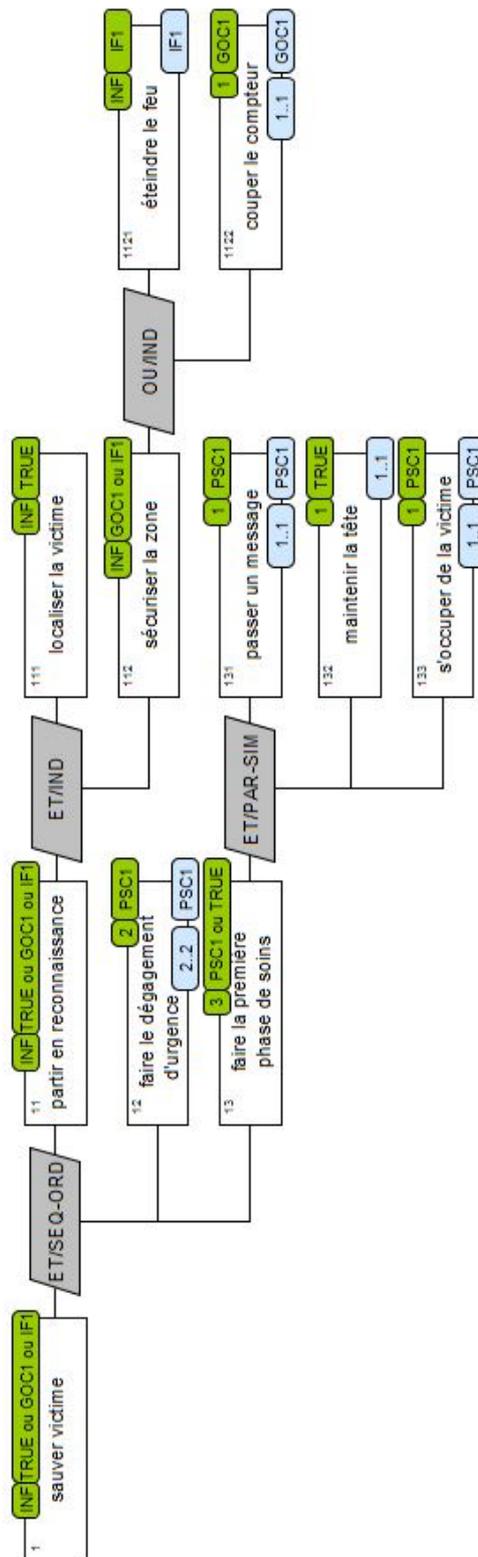


FIGURE B.5 – Même arbre de tâches que celui présenté en figure 5.2 sur lequel apparaissent, dans les bulles vertes en haut à droite de chaque tâche, les informations concernant le nombre maximal de personnes pouvant participer à chaque tâche à t_0 et les informations sur les compétences nécessaires, pour un agent, pour pouvoir participer à la tâche à t_0 . Cette figure est un agrandissement de la figure 5.5 présentée page 95.

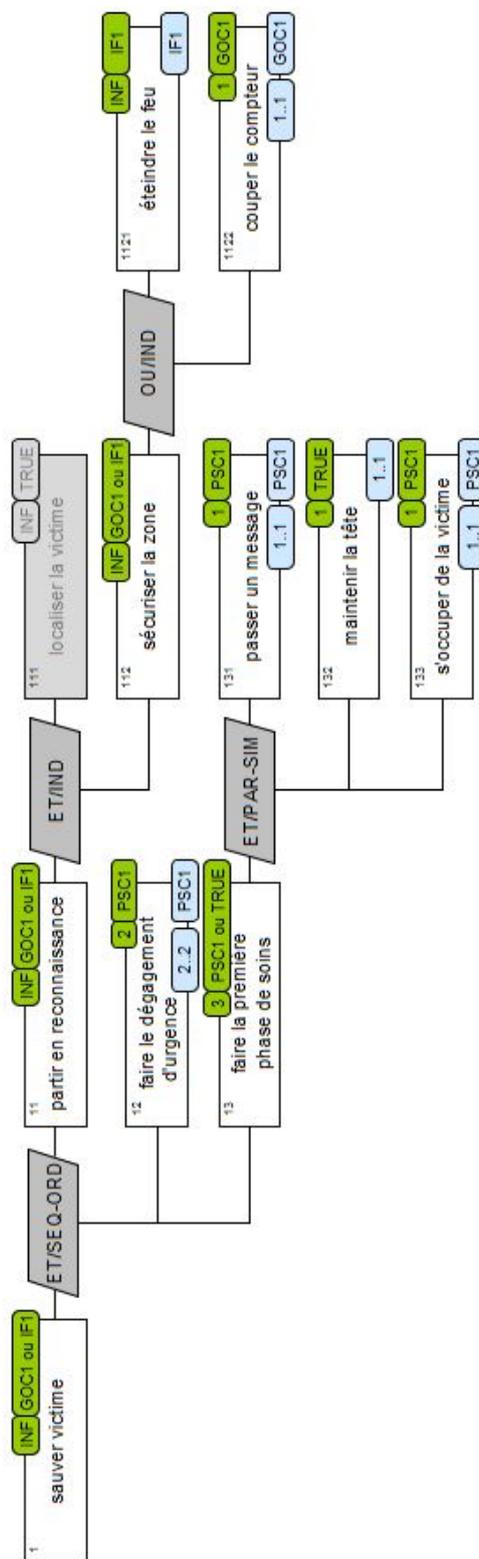


FIGURE B.6 – Même arbre de tâches que celui présenté en figure 5.2 sur lequel apparaissent, dans les bulles vertes en haut à droite de chaque tâche, les informations concernant le nombre maximal de personnes pouvant participer à chaque tâche à t_1 et les informations sur les compétences nécessaires, pour un agent, pour pouvoir participer à la tâche à t_1 . La tâche grisée est celle déjà réalisée. Cette figure est un agrandissement de la figure 5.6 présentée page 95.

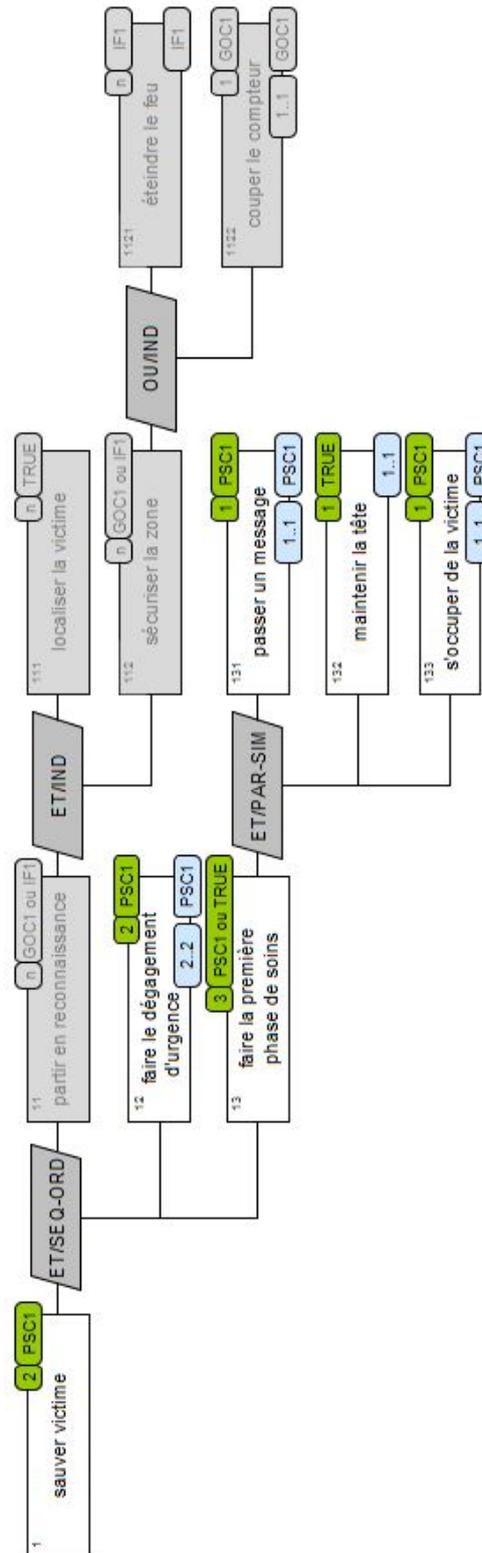


FIGURE B.7 – Même arbre de tâches que celui présenté en figure 5.2 sur lequel apparaissent, dans les bulles vertes en haut à droite de chaque tâche, les informations concernant le nombre maximal de personnes pouvant participer à chaque tâche à t_2 et les informations sur les compétences nécessaires, pour un agent, pour pouvoir participer à la tâche à t_2 . Les tâches grisées sont les tâches déjà réalisées. *Cette figure est un agrandissement de la figure 5.7 présentée page 96.*

B.2 ACTIVITY-DL : XML

Nous avons présenté dans le chapitre 5 un exemple d'activité décrit en ACTIVITY-DL s'appuyant sur un scénario de sauvetage de victime. Le XML correspondant à ce scénario est donné dans la section de code B.1 :

- La balise `tags` contient la description des marqueurs (balise `tag`) utilisés dans cette description d'activité : ici les marqueurs permettant de spécifier les compétences nécessaires (`ability`) et les nombres minimaux (`nbmin`) et maximaux (`nbmax`) de personnes pouvant participer à une action.
- La balise `task` contient toutes les tâches : les tâches abstraites sont décrites dans `task_m` et les tâches correspondant à des actions sont décrites dans `task_f`.
- Les balises `task` contiennent chacune la description d'une tâche.

Dans le cas des tâches abstraites :

- `subtasks` contient la liste de toutes les sous-tâches.
- `constructor` correspond à la contrainte temporelle associée à la tâche. Si le constructeur est `SEQ-ORD`, cette balise contient des balises `relation` qui permettent de spécifier l'ordre de réalisation des sous-tâches.
- Enfin l'élément `conditions:satisfaction` correspond à la contrainte logique associée à la tâche.

Dans le cas des tâches correspondant à des actions, la balise `reftags` contient des `tag` qui permettent de spécifier les compétences nécessaires et les nombres minimaux et maximaux de personnes pouvant participer à la réalisation de la tâche.

Code B.1 – XML de l'arbre de tâches ACTIVITY-DL présenté chapitre 5

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <model version="*">
3    <tags>
4      <tag id="ability" name="ability" valued="true" />
5      <tag id="nbmin" name="nbmin" valued="true" />
6      <tag id="nbmax" name="nbmax" valued="true" />
7    </tags>
8    <tasks>
9      <task_m>
10       <task id="t_sauver_victime" name="sauver victime" iterative="false" optional="false">
11         <subtasks>
12           <subtask id="t_partir_en_reconnaissance" />
13           <subtask id="t_faire_le_degagement_d_urgence" />
14           <subtask id="t_faire_la_premiere_phase_de_soins" />
15         </subtasks>
16         <constructor type="SEQ-ORD">
17           <relation lh="t_partir_en_reconnaissance" operator="&lt;" rh="
18             t_faire_le_degagement_d_urgence" />
19           <relation lh="t_faire_le_degagement_d_urgence" operator="&lt;" rh="
20             t_faire_la_premiere_phase_de_soins" />
21         </constructor>
22         <conditions>
23           <satisfaction id="AND" name="AND" />
24         </conditions>
25       </task_m>
26       <task id="t_partir_en_reconnaissance" name="partir en reconnaissance" iterative="false"
27         optional="false">
28         <subtasks>
29           <subtask id="t_localiser_la_victime" />
30           <subtask id="t_securiser_la_zone" />
31         </subtasks>
32         <constructor type="IND" />
33         <conditions>
34           <satisfaction id="AND" name="AND" />
35         </conditions>
36       </task>
37       <task id="t_faire_la_premiere_phase_de_soins" name="faire la premiere phase de soins"
38         iterative="false" optional="false">

```

```

35     <subtasks>
36         <subtask id="t_passer_un_message"/>
37         <subtask id="t_maintenir_la_tete"/>
38         <subtask id="t_s_occuper_de_la_victime"/>
39     </subtasks>
40     <constructor type="PAR-SIM"/>
41     <conditions>
42         <satisfaction id="AND" name="AND"/>
43     </conditions>
44 </task>
45 <task id="t_securiser_la_zone" name="securiser la zone" iterative="false" optional="false"
46     >
47     <subtasks>
48         <subtask id="t_eteindre_le_feu"/>
49         <subtask id="t_couper_le_compteur"/>
50     </subtasks>
51     <constructor type="IND"/>
52     <conditions>
53         <satisfaction id="OR" name="OR"/>
54     </conditions>
55 </task>
56 </task_m>
57 <task_f>
58 <task id="t_faire_le_degagement_d_urgence" name="faire le degagement d urgence" iterative
59     ="false" optional="false">
60     <reftags>
61         <reftag id="nbmin" value='2'/>
62         <reftag id="nbmax" value='2'/>
63         <reftag id="ability" value="PSC1" />
64     </reftags>
65 </task>
66 <task id="t_localiser_la_victime" name="localiser la victime" iterative="false" optional=
67     "false">
68 </task>
69 <task id="t_eteindre_le_feu" name="eteindre le feu" iterative="false" optional="false">
70     <reftags>
71         <reftag id="ability" value="IF1" />
72     </reftags>
73 </task>
74 <task id="t_couper_le_compteur" name="couper le compteur" iterative="false" optional="
75     false">
76     <reftags>
77         <reftag id="nbmin" value='1'/>
78         <reftag id="nbmax" value='1'/>
79         <reftag id="ability" value="GOCl" />
80     </reftags>
81 </task>
82 <task id="t_passer_un_message" name="passer un message" iterative="false" optional="false"
83     >
84     <reftags>
85         <reftag id="nbmin" value='1'/>
86         <reftag id="nbmax" value='1'/>
87         <reftag id="ability" value="PSC1" />
88     </reftags>
89 </task>
90 <task id="t_maintenir_la_tete" name="maintenir la tete" iterative="false" optional="false"
91     >
92     <reftags>
93         <reftag id="nbmin" value='1'/>
94         <reftag id="nbmax" value='1'/>
95         <reftag id="ability" value="PSC1" />
96     </reftags>
97 </task>
98 </task_f>
99 </tasks>
100 </model>

```

B.3 ACTIVITY-DL : workflow

Au cours de cette thèse, nous avons également été amenés, notamment avec l'étudiante Chloé Dejon encadrée lors d'un projet tutoré, à travailler sur différentes formes de représentation de l'activité, et en particulier sur une représentation sous forme de workflow. En effet, un workflow semble être plus facilement abordable pour les personnes non familières de ACTIVITY-DL, puisqu'il permet une représentation plus naturelle notamment des aspects temporels. Nous abordons ici ce format de représentation.

B.3.1 Représentation graphique

La représentation proposée est la suivante :

- Les tâches "feuilles" d'un arbre de tâches sont représentées sous forme d'un rectangle.
- Les tâches abstraites sont représentées par des barres verticales qui en délimitent le début et la fin. Pour ne pas surcharger le graphe produit, les noms de ces tâches n'apparaissent pas nécessairement.
- L'aspect temporel est ici représenté graphiquement (contrairement à la représentation sous forme d'arbre) :
 - Les séquences de tâches (contrainte temporelle SEQ-ORD) sont représentées à l'aide de flèches liant les tâches.
 - Les tâches qui doivent être effectuées en parallèle (contrainte temporelle PAR) sont groupées entre les barres verticales de couleur *gris clair* correspondant à leur tâche mère.
 - Les tâches pouvant être effectuées en parallèle ou séquentiellement (contrainte temporelle IND) sont groupées entre les barres verticales de couleur *gris foncé* correspondant à leur tâche mère.
- Les contraintes logiques associées à chaque tâche abstraite apparaissent sous chaque barre de début de tâche.

B.3.2 *Exemple*

La représentation graphique sous forme de workflow de l'activité décrite à titre d'exemple dans le chapitre 5 page 75 est donnée dans la figure B.8.

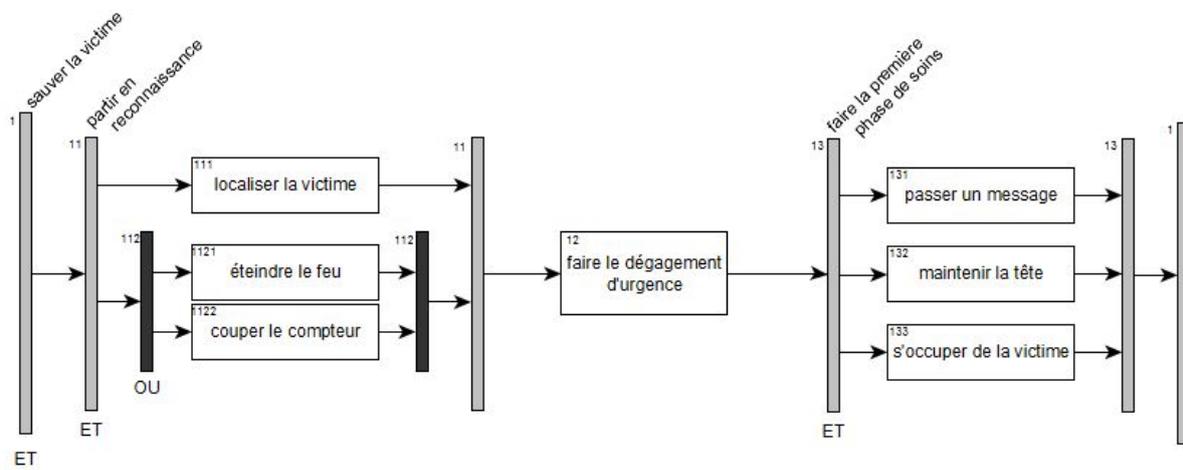


FIGURE B.8 – Représentation de l'activité sous forme de workflow : l'activité représentée est celle décrite dans le chapitre 5 page 75. Cette représentation peut être générée à partir d'un fichier XML décrivant une activité en ACTIVITY-DL tel que celui présenté dans la section précédente.

Annexe C

Notations

Le tableau C.1 récapitule les notations utilisées dans ce manuscrit pour manipuler les éléments liés à un arbre de tâches décrit en ACTIVITY-DL (chapitre 5) et pour définir les mécanismes régissant le fonctionnement de notre moteur décisionnel (chapitre 6).

TABLE C.1 – Récapitulatif des notations utilisées dans le manuscrit.

Notation	Signification
<i>Éléments associés à un arbre de tâches décrit en ACTIVITY-DL (voir chapitre 5).</i>	
τ_k	Désigne n'importe quelle tâche de l'arbre de tâches.
$T(\tau_k)$ $T(\tau_k) = \{\tau_{k1}, \dots, \tau_{kn}\}$ $T(\tau_k) = \emptyset$	Ensemble des sous-tâches de la tâche τ_k . → si τ_k est une tâche composée ayant n sous-tâches, on note $\{\tau_{k1}, \dots, \tau_{kn}\}$ l'ensemble $T(\tau_k)$; → si τ_k est une tâche feuille, elle n'a pas de sous-tâches et T_k est l'ensemble vide.
κ_i	Désigne n'importe quelle compétence.
$n_i(\tau_k) \cdot \kappa_i$	$n_i(\tau_k)$ personnes doivent avoir la compétence κ_i pour réaliser collectivement la tâche τ_k .
<i>Éléments liés à l'expression des contraintes de faisabilité collective (voir 5.4)</i>	
$n_{min}(\tau_k)$	Nombre minimal de personnes nécessaires pour faire la tâche τ_k .
$c_{col}(\tau_k)$ $c_{col}(\tau_k) = TRUE$	Contrainte de faisabilité collective en termes de compétences attachée à la tâche τ_k , de la forme : → si τ_k est une tâche feuille qui ne nécessite aucune compétence ;
<i>Continue sur la page suivante</i>	

Notation	Signification
$c_{col}(\tau_k) = n_i(\tau_k) \cdot \kappa_i$ $= n_{min}(\tau_k) \cdot \kappa_i$ $c_{col}(\tau_k) = C_1(\tau_k) \vee \dots \vee C_{n(k)}(\tau_k)$ $= \bigvee_{j=1}^{n(k)} C_j(\tau_k)$ $C_j(\tau_k) = n_1(C_j(\tau_k)) \cdot \kappa_1 \wedge$ $\dots \wedge n_{m(j)}(C_j(\tau_k)) \cdot \kappa_{m(j)}$	<p>→ si τ_k est une tâche feuille qui nécessite la compétence κ_i ;</p> <p>→ si τ_k est une tâche composée, $c_{col}(\tau_k)$ est une forme normale disjonctive, et</p> <p>chaque $C_j(\tau_k)$ est une conjonction d'éléments $n_x(C_j(\tau_k)) \cdot \kappa_x$.</p>
<i>Éléments liés à l'expression des informations dynamiques (voir 5.5)</i>	
$T_{NOTDONE}(\tau_k, t)$	Sous-ensemble des sous-tâches de τ_k qui ne sont pas encore réalisées à l'instant t .
$n_{max}(\tau_k, t)$	Nombre de personnes pouvant participer à la réalisation de τ_k à un instant t . Si τ_k est une tâche feuille, alors $n_{max}(\tau_k, t) = n_{max}(\tau_k)$.
$c(\tau_k, t)$ $c(\tau_k, t) = TRUE$ $c(\tau_k, t) = \kappa_i$ $c(\tau_k, t) = \kappa_1 \vee \dots \vee \kappa_m$	<p>Information sur les compétences nécessaires à un agent pour <i>participer</i> à la réalisation de τ_k à un instant t, de la forme :</p> <p>→ si τ_k est une tâche feuille qui ne nécessite aucune compétence ;</p> <p>→ si τ_k est une tâche feuille qui nécessite la compétence κ_i ;</p> <p>→ si τ_k est une tâche composée nécessitant les compétences $\{\kappa_1, \dots, \kappa_m\}$.</p>
<i>Éléments liés aux agents (voir chapitre 6)</i>	
<i>Éléments liés au modèle d'agent (voir 6.1)</i>	
$\Gamma_{x,self}$ $\Gamma_{x,team}$ $\Gamma_x = \Gamma_{x,self} \cup \Gamma_{x,team}$	<p><i>Buts de l'agent x :</i></p> <ul style="list-style-type: none"> – Ensemble des buts personnels de l'agent x. – Ensemble des buts collectifs de l'agent x. – Ensemble des buts de l'agent x.
i_x	<p><i>Dimensions de l'agent x :</i></p> <p>Valeur d'intégrité de l'agent x.</p>
<i>Continue sur la page suivante</i>	

Notation	Signification
b_{x,y_i}	Valeur de bienveillance de l'agent x envers l'agent y_i .
$a_{x,\kappa}$	Niveau de compétence de l'agent x sur la compétence κ .
$\Gamma_{y_i,self}^x$ $\Gamma_{y_i,team}^x$ $\Gamma_{y_i}^x = \Gamma_{y_i,self} \cup \Gamma_{y_i,team}^x$	<i>Croyances de l'agent x sur les buts de l'agent y_i :</i> – Ensemble des buts personnels de l'agent y_i selon x . – Ensemble des buts collectifs de l'agent y_i selon x . – Ensemble des buts de l'agent y_i selon x .
$i_{y_i}^x$ b_{y_i,y_j}^x $a_{y_i,\kappa}^x$	<i>Croyances de l'agent x sur les dimensions de l'agent y_i :</i> – Valeur d'intégrité de l'agent y_i selon x . – Valeur de bienveillance de l'agent y_i envers y_j selon x . – Niveau de compétence de l'agent y_i sur la compétence κ selon x .
<i>Éléments liés au processus décisionnel (voir 6.3, 6.4, 6.5 et 6.6)</i>	
$iImp_x(\gamma)$ (resp. $fImp_x(\gamma)$) $Imp_{y_i}^x(\gamma)$ (resp. $fImp_{y_i}^x(\gamma)$)	Importance initiale (resp. finale) pour x du but γ . Importance initiale (resp. finale) pour y_i du but γ selon x .
$\delta_{x,y_i}(\gamma)$ $\Delta_x(\gamma)$	Influence de y_i sur l'importance que x accorde au but γ . Influence globale de tous les autres agents de l'environnement sur l'importance que x accorde au but γ .
$a_x(\tau_k)$ $a_{y_i}^x(\tau_k)$	Niveau de compétence global de l'agent x sur la tâche τ_k . Niveau de compétence global de l'agent y_i sur la tâche τ_k selon x .
$U_x(\tau_k)$ $U_{y_i}^x(\tau_k)$	Utilité pour x de la tâche τ_k . Utilité pour y_i de la tâche τ_k selon x .
$(y_i, \tau_{k_{y_i,j}})$ $R_{x,j} = \{(x, \tau_{k_{x,j}}), (y_1, \tau_{k_{y_1,j}}), \dots, (y_{n-1}, \tau_{k_{y_{n-1},j}})\}$ $U_x(R_{x,j})$	L'agent y_i est assigné à la tâche $\tau_{k_{y_i,j}}$. Répartition de tâches $\#j$ générée par l'agent x . Utilité de la répartition $R_{x,j}$ selon x .
<i>Continue sur la page suivante</i>	

Notation	Signification
$R_x = \{R_{x,1}, \dots, R_{x,m}\}$ $R_{max_R, x} = \{R_{x,1}, \dots, R_{x,m(max_R)}\}$ $R_{max_\tau, x} = \{R_{x,1}, \dots, R_{x,m(max_\tau)}\}$	<p>Ensemble des m répartitions des tâches générées par l'agent x.</p> <p>Sous-ensemble des $m(max_R)$ répartitions de tâches pour lesquelles l'utilité $U_x(R_{x,j})$ de la répartition $R_{x,j}$ est maximisée.</p> <p>Sous-ensemble des $m(max_\tau)$ répartitions de tâches pour lesquelles l'utilité $U_x(\tau_{k_{x,j}})$ de la tâche $\tau_{k_{x,j}}$ à laquelle x est assigné dans la répartition $R_{x,j}$ est maximisée.</p>
$T_{x,n}$ $A_{x,n}$	<p>Ensemble des tâches sur lesquelles l'agent x raisonne à l'étape de récursivité n de son processus récursif de sélection de tâche.</p> <p>Ensemble des agents sur lesquels l'agent x raisonne à l'étape de récursivité n de son processus récursif de sélection de tâche.</p>
$A_{x,n}(\tau_{k_j})$	<p>Ensemble des agents étant assignés à la tâche τ_{k_j} dans la répartition $R_{x,j}$ de tâches sélectionnée par x; $A_{x,n}(\tau_{k_j})$ étant un sous-ensemble de $A_{x,n}$ et $A_{x,n}(\tau_{k_j})$ contenant au moins l'agent x.</p>
α_{ifl} α_{true}	<p><i>Coefficients à renseigner en fonction de l'application :</i></p> <ul style="list-style-type: none"> – Représente le pouvoir d'influence des agents les uns sur les autres. – Représente un niveau de compétence par défaut des agents pour les tâches ne nécessitant pas de compétences particulières.

Annexe D

Traitement des autres combinaisons de contraintes logiques et temporelles possibles avec Activity-DL

ACTIVITY-DL est le langage de description de l'activité collective avec lequel nous travaillons : c'est un langage très expressif, et nous avons précisé en 5.1.3 à quels éléments de ce langage nous nous intéressons en particulier dans le cadre de nos travaux; nous avons notamment détaillé les contraintes logiques et temporelles. Nous avons également précisé que par souci de simplicité nous ne développons pas toutes les combinaisons de contraintes logiques et temporelles qu'il est possible de rencontrer dans ACTIVITY-DL.

De manière à identifier l'ensemble des cas qu'il est possible de traiter avec notre systèmes, nous détaillons ici les autres contraintes logiques et temporelles qu'il est possible d'attacher à une tâche composée avec ACTIVITY-DL :

- La contrainte logique **OU-exclusif** permet de préciser qu'exactly une seule des sous-tâches doit être réalisée pour valider la tâche mère. Une fois cette sous-tâche effectuée, il n'est pas possible de faire les autres.
- La contrainte logique **NOT** permet de préciser quelles sous-tâches *ne doivent pas* être faites pour valider la tâche mère.
- La contrainte temporelle **SEQ** indique que les sous-tâches doivent être réalisées de manière séquentielle, mais l'ordre de réalisation n'a pas d'importance.
- La contrainte temporelle **PAR-START** permet de préciser que toutes les sous-tâches doivent être commencées en même temps.
- La contrainte temporelle **PAR-END** permet de préciser que toutes les sous-tâches doivent terminer en même temps.
- enfin la contrainte temporelle **PAR** indique que la réalisation de toutes les sous-tâches doit avoir un intervalle de temps commun.

Nous expliquons ici comment doivent être traitées les différentes combinaisons de contraintes logiques et temporelles que nous n'avons pas spécifiquement étudiées dans le chapitre 5.

D.1 Autres cas de génération des contraintes de faisabilité collective

Nous avons donc proposé en 5.4 un mécanisme de propagation de contrainte de faisabilité collective adapté à un nombre restreint de combinaisons (à savoir, les combinaisons ET/PAR-SIM, ET/SEQ-ORD, ET/IND, OU/SEQ-ORD et OU-IND). Néanmoins :

- Les combinaisons (i) ET/SEQ et (ii) ET/PAR-START et ET/PAR-END doivent être traitées respectivement de la même manière que (i) ET/SEQ-ORD et (ii) ET/PAR-SIM. En effet :
 - (i) Dans le cas ET/SEQ, toutes les tâches devront être effectuées de manière séquentielle. La seule différence avec la combinaison ET/SEQ-ORD est que dans ce cas, l'ordre des tâches n'est pas imposé. Cependant, le caractère imposé de l'ordre de réalisation des tâches n'a pas d'impact sur la génération de la contrainte de faisabilité collective.
 - (ii) Dans les cas ET/PAR-START et ET/PAR-END, comme dans le cas ET/PAR-SIM, à un instant t , toutes les tâches devront être exécutées en même temps (que ce soit commencées en même temps, terminées en même temps, ou commencées et terminées en même temps), et c'est ce caractère simultané qui est déterminant pour la génération de la contrainte de faisabilité collective.

De plus, les cas OU/PAR-START et OU/PAR-END seraient traités de la même manière que le cas OU/PAR-SIM pour les mêmes raisons qu'expliquées en (ii) ; le cas OU/PAR-SIM étant lui-même traité de la même manière que le cas ET/PAR-SIM (voir 5.1.3).

- La combinaison OU/SEQ doit être traitée de la même manière que OU/SEQ-ORD, pour les mêmes raisons qu'expliquées en (i) : le caractère imposé de l'ordre de réalisation des sous-tâches n'impacte pas la génération de la contrainte de faisabilité collective.
- De manière générale, pour la génération de la contrainte de faisabilité collective, la contrainte logique OU-exclusif doit être traitée de la même manière que le OU.
- Enfin nous n'avons pas du tout traité la contrainte logique NOT. Cependant, pour la contrainte de faisabilité collective, le NOT n'apporte pas d'information, puisqu'il permet de préciser qu'une tâche ne doit *pas* être réalisée. Or, le principe de la contrainte de faisabilité collective est de répertorier les compétences qui sont nécessaires pour exécuter les tâches qui doivent être réalisées.

La seule contrainte que nous ne pouvons pas traiter avec les mécanismes proposés dans ce manuscrit est la contrainte temporelle PAR : dans ce cas, les sous-tâches doivent être exécutées avec au moins un intervalle de temps commun. Cette contrainte est très imprécise dans le sens où de nombreux sous-cas peuvent être dérivés de cette contrainte (e.g. toutes les sous-tâches sont exécutées en même temps ; les sous-tâches sont exécutées en séquences, chaque sous-tâche à l'exception de la première débutant un peu avant que la précédente ne soit terminée ; etc). Pour gérer une telle contrainte, il faudrait considérer tous les sous-cas possibles.

D.2 Autres cas de propagation d'informations

Nous avons ensuite proposé en 5.5 un mécanisme de propagation d'informations dynamiques lui aussi adapté au même nombre restreint de combinaisons (à savoir, les combinaisons ET/PAR-SIM, ET/SEQ-ORD, ET/IND, OU/SEQ-ORD et OU-IND).

Nous avons vu que pour la propagation de ces informations, les contraintes logiques ne sont pas déterminantes car les agents doivent dans tous les cas choisir une sous-tâche à exécuter, alors que les contraintes temporelles le sont. Le seul cas où le choix des agents quant à la sous-tâche à effectuer est contraint est le cas **SEQ-ORD**, puisque les sous-tâches doivent être effectuées dans un ordre précis. Dans tous les autres cas, l'agent aura le choix entre toutes les sous-tâches restant à faire. Les autres cas de combinaisons de contraintes logiques et temporel doivent être traités comme suit :

- Le **OU-exclusif/SEQ-ORD** doit être traité de la même manière que le **OU/SEQ-ORD** et le **ET/SEQ-ORD** puisqu'ici aussi, l'ordre de réalisation des tâches est imposé ; les agents n'ont donc pas le choix.
- Pour les mêmes raisons qu'expliquées en [D.1](#), aucune information ne peut être propagée à partir d'une tâche à laquelle est attachée une contrainte logique **NOT**.
- Enfin, toutes combinaisons autres que **OU-exclusif/SEQ-ORD** et que celles que nous avons détaillées dans le chapitre 5 et ne comportant pas de contrainte logique **NOT** doivent être traitées de la même manière que défini pour les cas **ET/PAR-SIM**, **ET/IND** et **OU/IND**. On remarque qu'ici, la contrainte temporelle **PAR** ne pose pas de problème.

Annexe E

Prise de décision

Nous donnons ici les détails des calculs effectués par le moteur décisionnel de l'agent Anna dans le scénario *S2a* décrit dans la section 7.2.1 du chapitre 7.

E.1 Scénario *S2a*

E.1.1 Importance des buts

Les tableaux E.1 et E.2 donnent respectivement les valeurs calculées par Anna concernant l'importance initiale que les agents accordent aux buts et l'influence des agents sur chacun des autres pour le calcul de l'importance finale accordée à ces mêmes buts.

TABLE E.1 – *S2a* – Importance initiale des buts calculée par Anna. La valeur $iImp_a(\gamma)$ représente l'importance pour Anna du but γ . La valeur $iImp_{y_i}^a(\gamma)$ représente l'importance du but γ pour y_i selon Anna.

	<i>monter les meubles</i>	<i>installer info.</i>	<i>formaliser le modèle</i>	<i>écrire un papier</i>
Anna	$iImp_a(\gamma_m) = 0.25$	$iImp_a(\gamma_e) = 0.25$	$iImp_a(\gamma_f) = 0.75$	$iImp_a(\gamma_p) = 0$
Bea	$iImp_b^a(\gamma_m) = 0$	$iImp_b^a(\gamma_e) = 0.5$	$iImp_b^a(\gamma_f) = 0$	$iImp_b^a(\gamma_p) = 0.5$
Cara	$iImp_c^a(\gamma_m) = 0.5$	$iImp_c^a(\gamma_e) = 0.5$	$iImp_c^a(\gamma_f) = 0$	$iImp_c^a(\gamma_p) = 0$
Dora	$iImp_d^a(\gamma_m) = 0.75$	$iImp_d^a(\gamma_e) = 0.75$	$iImp_d^a(\gamma_f) = 0$	$iImp_d^a(\gamma_p) = 0$

E.1.2 Utilité des tâches

Le tableau E.3 donne l'utilité des tâches pour chacun des agents calculée par Anna.

E.1.3 Sélection d'une tâche

Dans le tableau E.3, on voit que pour chaque agent y_i , les ensembles de tâches T_{U_{max}, y_i}^x pour lesquelles l'utilité est maximisée ne contiennent qu'un seul élément :

- $T_{U_{max}, a} = \{\tau_f\}$,
- $T_{U_{max}, b}^a = \{\tau_p\}$,
- $T_{U_{max}, c}^a = \{\tau_e\}$, et

TABLE E.2 – S2a – Ce qu’Anna pense être l’influence des autres agents sur l’importance des buts pour chaque agent.

La valeur $\delta_{a,y_i}(\gamma)$ représente l’influence de l’agent y_i sur l’importance accordée au but γ par Anna. La valeur $\Delta_a(\gamma)$ représente l’influence de l’ensemble des agents sur l’importance accordée au but γ par Anna. La valeur $\delta_{y_i,y_j}^a(\gamma)$ représente l’influence de l’agent y_j sur l’importance accordée au but γ par y_i selon Anna. $\Delta_{y_i}^a(\gamma)$ représente l’influence de l’ensemble des agents sur l’importance accordée au but γ par y_i selon Anna.

	<i>monter les meubles</i>	<i>installer info.</i>	<i>formaliser le modèle</i>	<i>écrire un papier</i>
Pour Anna				
Bea	—	$\delta_{a,b}(\gamma_e) \simeq 0.062$	—	$\delta_{a,b}(\gamma_p) = 0.125$
Cara	$\delta_{a,c}(\gamma_m) \simeq 0.062$	$\delta_{a,c}(\gamma_e) \simeq 0.062$	—	—
Dora	$\delta_{a,d}(\gamma_m) = 0.125$	$\delta_{a,d}(\gamma_e) = 0.125$	—	—
	$\Delta_a(\gamma_m) \simeq 0.094$	$\Delta_a(\gamma_e) \simeq 0.083$	$\Delta_a(\gamma_f) = 0.0$	$\Delta_a(\gamma_p) = 0.125$
Pour Béa				
Anna	$\delta_{b,a}^a(\gamma_m) \simeq 0.062$	$\delta_{b,a}^a(\gamma_e) = -0.125$	$\delta_{b,a}^a(\gamma_f) \simeq 0.188$	—
Cara	$\delta_{b,c}^a(\gamma_m) = 0.125$	$\delta_{b,c}^a(\gamma_e) = 0.0$	—	—
Dora	$\delta_{b,d}^a(\gamma_m) \simeq 0.188$	$\delta_{b,d}^a(\gamma_e) \simeq 0.062$	—	—
	$\Delta_b^a(\gamma_m) = 0.125$	$\Delta_b^a(\gamma_e) \simeq -0.021$	$\Delta_b^a(\gamma_f) \simeq 0.188$	$\Delta_b^a(\gamma_p) = 0.0$
Pour Cara				
Anna	$\delta_{c,a}^a(\gamma_m) = -0.125$	$\delta_{c,a}^a(\gamma_e) = -0.125$	$\delta_{c,a}^a(\gamma_f) \simeq 0.188$	—
Bea	—	$\delta_{c,b}^a(\gamma_e) = 0.0$	—	$\delta_{c,b}^a(\gamma_p) = 0.125$
Dora	$\delta_{c,d}^a(\gamma_m) \simeq 0.062$	$\delta_{c,d}^a(\gamma_e) \simeq 0.062$	—	—
	$\Delta_c^a(\gamma_m) \simeq -0.031$	$\Delta_c^a(\gamma_e) \simeq -0.021$	$\Delta_c^a(\gamma_f) \simeq 0.188$	$\Delta_c^a(\gamma_p) = 0.125$
Pour Dora				
Anna	$\delta_{d,a}^a(\gamma_m) = -0.25$	$\delta_{d,a}^a(\gamma_e) = -0.25$	$\delta_{d,a}^a(\gamma_f) \simeq 0.188$	—
Bea	—	$\delta_{d,b}^a(\gamma_e) = -0.125$	—	$\delta_{d,b}^a(\gamma_p) = 0.125$
Cara	$\delta_{d,c}^a(\gamma_m) = -0.125$	$\delta_{d,c}^a(\gamma_e) = -0.125$	—	—
	$\Delta_d^a(\gamma_m) \simeq -0.188$	$\Delta_d^a(\gamma_e) \simeq -0.167$	$\Delta_d^a(\gamma_f) \simeq 0.188$	$\Delta_d^a(\gamma_p) = 0.125$

TABLE E.3 – S2a – Valeurs d’utilité pour toutes les tâches et pour tous les agents calculées par Anna : $U_a(\tau)$ représente l’utilité pour Anna de la tâche τ et $U_{y_i}^a(\tau)$ représente l’utilité pour y_i de faire la tâche τ selon Anna.

	<i>monter les meubles</i>	<i>installer info.</i>	<i>formaliser le modèle</i>	<i>écrire un papier</i>
Anna	$U_a(\tau_m) \simeq 0.172$	$U_a(\tau_e) \simeq 0.083$	$U_a(\tau_f) = 0.375$	$U_a(\tau_p) \simeq 0.062$
Bea	—	$U_b^a(\tau_e) \simeq 0.24$	$U_b^a(\tau_f) \simeq 0.094$	$U_b^a(\tau_p) = 0.25$
Cara	—	$U_c^a(\tau_e) \simeq 0.24$	$U_c^a(\tau_f) \simeq 0.094$	$U_c^a(\tau_p) \simeq 0.062$
Dora	$U_d^a(\tau_m) \simeq 0.281$	$U_d^a(\tau_e) \simeq 0.146$	$U_d^a(\tau_f) \simeq 0.094$	$U_d^a(\tau_p) \simeq 0.062$

— $T_{U_{max,d}^a} = \{\tau_m\}$.

La répartition de tâches formée à partir de ces éléments est donc :

$$R_{a,1} = \{(a, \tau_f), (b, \tau_p), (c, \tau_e), (d, \tau_m)\} \quad .$$

Cette répartition est valide (pour chaque tâche τ_k , on a bien $n(\tau_k) \leq n_{max}(\tau_k)$). C'est donc cette répartition qui est choisie par l'agent Anna.

E.2 Scénario S2b

Nous donnons ici les détails des calculs effectués par le moteur décisionnel de l'agent Anna dans le scénario S2b décrit dans la section 7.2.1 du chapitre 7.

E.2.1 Importance des buts

Les tableaux E.4 et E.5 donnent respectivement les valeurs calculées par Anna concernant l'importance initiale que les agents accordent aux buts et l'influence des agents sur chacun des autres pour le calcul de l'importance finale accordée à ces mêmes buts.

TABLE E.4 – S2b – Importance initiale des buts calculée par Anna. La valeur $iImp_a(\gamma)$ représente l'importance pour Anna du but γ . La valeur $iImp_{y_i}^a(\gamma)$ représente l'importance du but γ pour y_i selon Anna.

	<i>monter les meubles</i>	<i>installer info.</i>	<i>formaliser le modèle</i>	<i>écrire un papier</i>
Anna	$iImp_a(\gamma_m) = 0.5$	$iImp_a(\gamma_e) = 0.5$	$iImp_a(\gamma_f) = 0.5$	$iImp_a(\gamma_p) = 0$
Bea	$iImp_b^a(\gamma_m) = 0$	$iImp_b^a(\gamma_e) = 0.5$	$iImp_b^a(\gamma_f) = 0$	$iImp_b^a(\gamma_p) = 0.5$
Cara	$iImp_c^a(\gamma_m) = 0.5$	$iImp_c^a(\gamma_e) = 0.5$	$iImp_c^a(\gamma_f) = 0$	$iImp_c^a(\gamma_p) = 0$
Dora	$iImp_d^a(\gamma_m) = 0.75$	$iImp_d^a(\gamma_e) = 0.75$	$iImp_d^a(\gamma_f) = 0$	$iImp_d^a(\gamma_p) = 0$

E.2.2 Utilité des tâches

Le tableau E.6 donne l'utilité des tâches pour chacun des agents calculée par Anna.

E.2.3 Sélection d'une tâche

D'après le tableau E.6, on voit que pour chaque agent y_i , les ensembles de tâches T_{U_{max}, y_i}^x pour lesquelles l'utilité est maximisée ne contiennent qu'un seul élément :

- $T_{U_{max}, a} = \{\tau_m\}$,
- $T_{U_{max}, b}^a = \{\tau_e\}$,
- $T_{U_{max}, c}^a = \{\tau_e\}$, et
- $T_{U_{max}, d}^a = \{\tau_m\}$.

La répartition formée à partir de ces éléments est donc :

$$R_{a,1} = \{(a, \tau_m), (b, \tau_e), (c, \tau_e), (d, \tau_m)\} \quad .$$

Cette répartition est valide (pour chaque tâche τ_k , on a bien $n(\tau_k) \leq n_{max}(\tau_k)$). C'est donc cette répartition qui est choisie par l'agent Anna.

TABLE E.5 – S2b – Ce qu’Anna pense être l’influence des autres agents sur l’importance des buts pour chaque agent.

La valeur $\delta_{a,y_i}(\gamma)$ représente l’influence de l’agent y_i sur l’importance accordée au but γ par Anna. La valeur $\Delta_a(\gamma)$ représente l’influence de l’ensemble des agents sur l’importance accordée au but γ par Anna. La valeur $\delta_{y_i,y_j}^a(\gamma)$ représente l’influence de l’agent y_j sur l’importance accordée au but γ par y_i selon Anna. $\Delta_{y_i}^a(\gamma)$ représente l’influence de l’ensemble des agents sur l’importance accordée au but γ par y_i selon Anna.

	<i>monter les meubles</i>	<i>installer info.</i>	<i>formaliser le modèle</i>	<i>écrire un papier</i>
Pour Anna				
Bea	—	$\delta_{a,b}(\gamma_e) = 0.0$	—	$\delta_{a,b}(\gamma_p) = 0.125$
Cara	$\delta_{a,c}(\gamma_m) = 0.0$	$\delta_{a,c}(\gamma_e) = 0.0$	—	—
Dora	$\delta_{a,d}(\gamma_m) \simeq 0.062$	$\delta_{a,d}(\gamma_e) \simeq 0.062$	—	—
	$\Delta_a(\gamma_m) \simeq 0.031$	$\Delta_a(\gamma_e) \simeq 0.021$	$\Delta_a(\gamma_f) = 0.0$	$\Delta_a(\gamma_p) = 0.125$
Pour Béa				
Anna	$\delta_{b,a}^a(\gamma_m) = 0.125$	$\delta_{b,a}^a(\gamma_e) = 0.0$	$\delta_{b,a}^a(\gamma_f) = 0.125$	—
Cara	$\delta_{b,c}^a(\gamma_m) = 0.125$	$\delta_{b,c}^a(\gamma_e) = 0.0$	—	—
Dora	$\delta_{b,d}^a(\gamma_m) \simeq 0.188$	$\delta_{b,d}^a(\gamma_e) \simeq 0.062$	—	—
	$\Delta_b^a(\gamma_m) \simeq 0.146$	$\Delta_b^a(\gamma_e) \simeq 0.021$	$\Delta_b^a(\gamma_f) = 0.125$	$\Delta_b^a(\gamma_p) = 0.0$
Pour Cara				
Anna	$\delta_{c,a}^a(\gamma_m) = 0.0$	$\delta_{c,a}^a(\gamma_e) = 0.0$	$\delta_{c,a}^a(\gamma_f) = 0.125$	—
Bea	—	$\delta_{c,b}^a(\gamma_e) = 0.0$	—	$\delta_{c,b}^a(\gamma_p) = 0.125$
Dora	$\delta_{c,d}^a(\gamma_m) \simeq 0.062$	$\delta_{c,d}^a(\gamma_e) \simeq 0.062$	—	—
	$\Delta_c^a(\gamma_m) \simeq 0.031$	$\Delta_c^a(\gamma_e) \simeq 0.021$	$\Delta_c^a(\gamma_f) = 0.125$	$\Delta_c^a(\gamma_p) = 0.125$
Pour Dora				
Anna	$\delta_{d,a}^a(\gamma_m) = -0.125$	$\delta_{d,a}^a(\gamma_e) = -0.125$	$\delta_{d,a}^a(\gamma_f) = 0.125$	—
Bea	—	$\delta_{d,b}^a(\gamma_e) = -0.125$	—	$\delta_{d,b}^a(\gamma_p) = 0.125$
Cara	$\delta_{d,c}^a(\gamma_m) = -0.125$	$\delta_{d,c}^a(\gamma_e) = -0.125$	—	—
	$\Delta_d^a(\gamma_m) = -0.125$	$\Delta_d^a(\gamma_e) = -0.125$	$\Delta_d^a(\gamma_f) = 0.125$	$\Delta_d^a(\gamma_p) = 0.125$

TABLE E.6 – S2b – Valeurs d’utilité pour toutes les tâches et pour tous les agents calculées par Anna : $U_a(\tau)$ représente l’utilité pour Anna de la tâche τ et $U_{y_i}^a(\tau)$ représente l’utilité pour y_i de faire la tâche τ selon Anna.

	<i>monter les meubles</i>	<i>installer info.</i>	<i>formaliser le modèle</i>	<i>écrire un papier</i>
Anna	$U_a(\tau_m) \simeq 0.266$	$U_a(\tau_e) \simeq 0.13$	$U_a(\tau_f) = 0.25$	$U_a(\tau_p) \simeq 0.062$
Bea	—	$U_b^a(\tau_e) \simeq 0.26$	$U_b^a(\tau_f) \simeq 0.062$	$U_b^a(\tau_p) = 0.25$
Cara	—	$U_c^a(\tau_e) \simeq 0.26$	$U_c^a(\tau_f) \simeq 0.062$	$U_c^a(\tau_p) \simeq 0.062$
Dora	$U_d^a(\tau_m) \simeq 0.312$	$U_d^a(\tau_e) \simeq 0.156$	$U_d^a(\tau_f) \simeq 0.062$	$U_d^a(\tau_p) \simeq 0.062$

E.3 Scénario S3

Nous donnons ici les détails des calculs effectués par le moteur décisionnel de l'agent Anna dans le scénario S3 décrit dans la section 7.2.2 du chapitre 7.

E.3.1 Importance des buts

Les tableaux E.7 et E.8 donnent respectivement les valeurs calculées par Anna concernant l'importance initiale que les agents accordent aux buts et l'influence des agents sur chacun des autres pour le calcul de l'importance finale accordée à ces mêmes buts.

TABLE E.7 – S3 – Importance initiale des buts calculée par Anna. La valeur $iImp_a(\gamma)$ représente l'importance pour Anna du but γ . La valeur $iImp_{y_i}^a(\gamma)$ représente l'importance du but γ pour y_i selon Anna.

	<i>monter les meubles</i>	<i>installer info.</i>	<i>formaliser le modèle</i>	<i>écrire un papier</i>
Anna	$iImp_a(\gamma_m) = 0.5$	$iImp_a(\gamma_e) = 0.5$	$iImp_a(\gamma_f) = 0.5$	$iImp_a(\gamma_p) = 0$
Bea	$iImp_b^a(\gamma_m) = 0$	$iImp_b^a(\gamma_e) = 0.5$	$iImp_b^a(\gamma_f) = 0$	$iImp_b^a(\gamma_p) = 0.5$
Cara	$iImp_c^a(\gamma_m) = 0.5$	$iImp_c^a(\gamma_e) = 0.5$	$iImp_c^a(\gamma_f) = 0$	$iImp_c^a(\gamma_p) = 0$
Dora	$iImp_d^a(\gamma_m) = 0.75$	$iImp_d^a(\gamma_e) = 0.75$	$iImp_d^a(\gamma_f) = 0$	$iImp_d^a(\gamma_p) = 0$

E.3.2 Utilité des tâches

Le tableau E.9 donne l'utilité des tâches pour chacun des agents calculée par Anna.

E.3.3 Sélection d'une tâche

Dans le tableau E.9, on peut déduire, pour chaque agent y_i , les ensembles T_{U_{max},y_i}^x de tâches pour lesquelles l'utilité est maximisée :

- $T_{U_{max},a} = \{\tau_m, \tau_e, \tau_f\}$,
- $T_{U_{max},b}^a = \{\tau_p\}$,
- $T_{U_{max},c}^a = \{\tau_e\}$, et
- $T_{U_{max},d}^a = \{\tau_m\}$.

Ici, l'ensemble $T_{U_{max},a}$ contient plus d'un élément. Anna génère donc plusieurs répartitions de tâches qui sont les suivantes :

$$R_{a,1} = \{(a, \tau_m), (b, \tau_e), (c, \tau_e), (d, \tau_m)\}$$

$$R_{a,2} = \{(a, \tau_e), (b, \tau_e), (c, \tau_e), (d, \tau_m)\}$$

$$R_{a,3} = \{(a, \tau_f), (b, \tau_e), (c, \tau_e), (d, \tau_m)\}$$

Parmi ces répartitions, $R_{a,2}$ n'est pas valide car trois agents sont assignés à la tâche *installer l'équipement informatique*, alors que deux personnes au maximum peuvent participer à cette tâche (i.e. voir figure 7.1 – informations dynamiques en termes de nombre maximal de personnes pouvant participer à une tâche). Anna doit donc faire un choix entre les répartitions $R_{a,1}$ et $R_{a,3}$.

TABLE E.8 – S3 – Ce qu’Anna pense être l’influence des autres agents sur l’importance des buts pour chaque agent.

La valeur $\delta_{a,y_i}(\gamma)$ représente l’influence de l’agent y_i sur l’importance accordée au but γ par Anna. La valeur $\Delta_a(\gamma)$ représente l’influence de l’ensemble des agents sur l’importance accordée au but γ par Anna. La valeur $\delta_{y_i,y_j}^a(\gamma)$ représente l’influence de l’agent y_j sur l’importance accordée au but γ par y_i selon Anna. $\Delta_{y_i}^a(\gamma)$ représente l’influence de l’ensemble des agents sur l’importance accordée au but γ par y_i selon Anna.

	<i>monter les meubles</i>	<i>installer info.</i>	<i>formaliser le modèle</i>	<i>écrire un papier</i>
Pour Anna				
Bea	—	$\delta_{a,b}(\gamma_e) = 0.0$	—	$\delta_{a,b}(\gamma_p) = 0.125$
Cara	$\delta_{a,c}(\gamma_m) = 0.0$	$\delta_{a,c}(\gamma_e) = 0.0$	—	—
Dora	$\delta_{a,d}(\gamma_m) = 0.0$	$\delta_{a,d}(\gamma_e) = 0.0$	—	—
	$\Delta_a(\gamma_m) = 0.0$	$\Delta_a(\gamma_e) = 0.0$	$\Delta_a(\gamma_f) = 0.0$	$\Delta_a(\gamma_p) = 0.125$
Pour Béa				
Anna	$\delta_{b,a}^a(\gamma_m) = 0.125$	$\delta_{b,a}^a(\gamma_e) = 0.0$	$\delta_{b,a}^a(\gamma_f) = 0.125$	—
Cara	$\delta_{b,c}^a(\gamma_m) = 0.125$	$\delta_{b,c}^a(\gamma_e) = 0.0$	—	—
Dora	$\delta_{b,d}^a(\gamma_m) \simeq 0.188$	$\delta_{b,d}^a(\gamma_e) \simeq 0.062$	—	—
	$\Delta_b^a(\gamma_m) \simeq 0.146$	$\Delta_b^a(\gamma_e) \simeq 0.021$	$\Delta_b^a(\gamma_f) = 0.125$	$\Delta_b^a(\gamma_p) = 0.0$
Pour Cara				
Anna	$\delta_{c,a}^a(\gamma_m) = 0.0$	$\delta_{c,a}^a(\gamma_e) = 0.0$	$\delta_{c,a}^a(\gamma_f) = 0.125$	—
Bea	—	$\delta_{c,b}^a(\gamma_e) = 0.0$	—	$\delta_{c,b}^a(\gamma_p) = 0.125$
Dora	$\delta_{c,d}^a(\gamma_m) \simeq 0.062$	$\delta_{c,d}^a(\gamma_e) \simeq 0.062$	—	—
	$\Delta_c^a(\gamma_m) \simeq 0.031$	$\Delta_c^a(\gamma_e) \simeq 0.021$	$\Delta_c^a(\gamma_f) = 0.125$	$\Delta_c^a(\gamma_p) = 0.125$
Pour Dora				
Anna	$\delta_{d,a}^a(\gamma_m) = -0.125$	$\delta_{d,a}^a(\gamma_e) = -0.125$	$\delta_{d,a}^a(\gamma_f) = 0.0$	—
Bea	—	$\delta_{d,b}^a(\gamma_e) = -0.125$	—	$\delta_{d,b}^a(\gamma_p) = 0.125$
Cara	$\delta_{d,c}^a(\gamma_m) = -0.125$	$\delta_{d,c}^a(\gamma_e) = -0.125$	—	—
	$\Delta_d^a(\gamma_m) = -0.125$	$\Delta_d^a(\gamma_e) = -0.125$	$\Delta_d^a(\gamma_f) = 0.0$	$\Delta_d^a(\gamma_p) = 0.125$

TABLE E.9 – S3 – Valeurs d’utilité pour toutes les tâches et pour tous les agents calculées par Anna : $U_a(\tau)$ représente l’utilité pour Anna de la tâche τ et $U_{y_i}^a(\tau)$ représente l’utilité pour y_i de faire la tâche τ selon Anna.

	<i>monter les meubles</i>	<i>installer info.</i>	<i>formaliser le modèle</i>	<i>écrire un papier</i>
Anna	$U_a(\tau_m) = 0.25$	$U_a(\tau_e) = 0.125$	$U_a(\tau_f) = 0.25$	$U_a(\tau_p) \simeq 0.062$
Bea	—	$U_b^a(\tau_e) \simeq 0.26$	$U_b^a(\tau_f) \simeq 0.062$	$U_b^a(\tau_p) = 0.25$
Cara	—	$U_c^a(\tau_e) \simeq 0.26$	$U_c^a(\tau_f) \simeq 0.062$	$U_c^a(\tau_p) \simeq 0.062$
Dora	$U_d^a(\tau_m) \simeq 0.312$	$U_d^a(\tau_e) \simeq 0.156$	$U_d^a(\tau_f) = 0.0$	$U_d^a(\tau_p) \simeq 0.062$

Pour cela, Anna calcule l’utilité de ces répartitions. Elle trouve $U_a(R_{a,1}) = U_a(R_{a,3}) \simeq 0.203$.

Anna doit maintenant sélectionner la meilleure répartition. Pour cela elle cherche tout d'abord $R_{max_R,a}$, l'ensemble des répartitions dont l'utilité est la plus élevée. Les répartitions $R_{a,1}$ et $R_{a,3}$ ayant la même utilité, Anna trouve :

$$R_{max_R,a} = \{R_{a,1}, R_{a,3}\} \quad .$$

L'ensemble $R_{max_R,a}$ contenant plus d'un élément, Anna cherche le sous-ensemble $R_{max_\tau,a} \subseteq R_{max_R,a}$ de répartitions pour lesquelles l'utilité de la tâche à laquelle elle est assignée est maximisée :

$$R_{max_\tau,a} = \{R_{a,1}, R_{a,3}\} \quad .$$

L'ensemble $R_{max_\tau,a}$ contenant toujours plus d'un élément, Anna choisira au hasard l'une des deux répartitions de tâches : Anna choisira donc au hasard de faire *formaliser le modèle* ou de faire *monter les meubles*. Dans ces deux répartitions les autres agents sont assignés aux mêmes tâches : Béa et Cara feront *installer l'équipement informatique* et Dora fera *monter les meubles*.

E.4 Scénario S_4

Nous donnons ici les détails des calculs effectués par le moteur décisionnel de l'agent Anna dans le scénario S_4 décrit dans la section 7.2.3 du chapitre 7.

E.4.1 Importance des buts

Les tableaux E.10, E.11 et E.12 donnent respectivement les valeurs calculées par Anna concernant l'importance initiale que les agents accordent aux buts, l'influence des agents sur chacun des autres pour le calcul de l'importance finale accordée à ces mêmes buts et les valeurs calculées par Anna concernant l'importance finale que les agents accordent aux buts.

TABLE E.10 – S_4 – Importance initiale des buts calculée par Anna. La valeur $iImp_a(\gamma)$ représente l'importance pour Anna du but γ . La valeur $iImp_{y_i}^a(\gamma)$ représente l'importance du but γ pour y_i selon Anna.

	<i>monter les meubles</i>	<i>installer info.</i>	<i>formaliser le modèle</i>	<i>écrire un papier</i>
Anna	$iImp_a(\gamma_m) = 0.75$	$iImp_a(\gamma_e) = 0.75$	$iImp_a(\gamma_f) = 0.25$	$iImp_a(\gamma_p) = 0$
Bea	$iImp_b^a(\gamma_m) = 0$	$iImp_b^a(\gamma_e) = 0.75$	$iImp_b^a(\gamma_f) = 0$	$iImp_b^a(\gamma_p) = 0.25$
Cara	$iImp_c^a(\gamma_m) = 0.5$	$iImp_c^a(\gamma_e) = 0.5$	$iImp_c^a(\gamma_f) = 0$	$iImp_c^a(\gamma_p) = 0$
Dora	$iImp_d^a(\gamma_m) = 0.75$	$iImp_d^a(\gamma_e) = 0.75$	$iImp_d^a(\gamma_f) = 0$	$iImp_d^a(\gamma_p) = 0$

E.4.2 Sélection d'une tâche

Le tableau E.13 présente les différentes répartitions de tâches générées par Anna.

TABLE E.13 – S_4 – Tableau des répartitions de tâches générées par Anna.

Les combinaisons où des agents ne faisaient rien ont été éliminées pour limiter la taille du tableau (ces combinaisons ayant une utilité moindre puisque l'utilité de ne rien faire est nulle).

La ligne 21 (répartition $R_{a,21}$) correspond à la répartition pour laquelle l'utilité est maximale.

j	Anna		Béa		Cara		$U_a(R_{a,j})$
	$\tau_{k_a,j}$	$U_a(\tau_{k_a,j})$	$\tau_{k_b,j}$	$U_b^a(\tau_{k_b,j})$	$\tau_{k_c,j}$	$U_c^a(\tau_{k_c,j})$	
1	τ_f	0.125	τ_f	0.031	τ_e	0.281	0.146
2	τ_f	0.125	τ_f	0.031	τ_p	0.031	0.062
3	τ_f	0.125	τ_e	0.354	τ_f	0.031	0.17
4	τ_f	0.125	τ_e	0.354	τ_e	0.281	0.253
5	τ_f	0.125	τ_e	0.354	τ_p	0.031	0.17
6	τ_f	0.125	τ_p	0.125	τ_f	0.031	0.094
7	τ_f	0.125	τ_p	0.125	τ_e	0.281	0.177
8	τ_f	0.125	τ_p	0.125	τ_p	0.031	0.094

Continue sur la page suivante

j	Anna		Béa		Cara		$U_a(R_{a,j})$
	$\tau_{k_{a,j}}$	$U_a(\tau_{k_{a,j}})$	$\tau_{k_{b,j}}$	$U_b^a(\tau_{k_{b,j}})$	$\tau_{k_{c,j}}$	$U_c^a(\tau_{k_{c,j}})$	
9	τ_e	0.531	τ_f	0.031	τ_f	0.031	0.198
10	τ_e	0.531	τ_f	0.031	τ_e	0.281	0.281
11	τ_e	0.531	τ_f	0.031	τ_p	0.031	0.198
12	τ_e	0.531	τ_e	0.354	τ_f	0.031	0.306
13	τ_e	0.531	τ_e	0.354	τ_p	0.031	0.306
14	τ_e	0.531	τ_p	0.125	τ_f	0.031	0.229
15	τ_e	0.531	τ_p	0.125	τ_e	0.281	0.312
16	τ_e	0.531	τ_p	0.125	τ_p	0.031	0.229
17	τ_m	0.344	τ_f	0.031	τ_f	0.031	0.135
18	τ_m	0.344	τ_f	0.031	τ_e	0.281	0.219
19	τ_m	0.344	τ_f	0.031	τ_p	0.031	0.135
20	τ_m	0.344	τ_e	0.354	τ_f	0.031	0.243
21	τ_m	0.344	τ_e	0.354	τ_e	0.281	0.326
22	τ_m	0.344	τ_e	0.354	τ_p	0.031	0.243
23	τ_m	0.344	τ_p	0.125	τ_f	0.031	0.167
24	τ_m	0.344	τ_p	0.125	τ_e	0.281	0.25
25	τ_m	0.344	τ_p	0.125	τ_p	0.031	0.167
26	τ_p	0.031	τ_f	0.031	τ_f	0.031	0.031
27	τ_p	0.031	τ_f	0.031	τ_e	0.281	0.115
28	τ_p	0.031	τ_f	0.031	τ_p	0.031	0.031
29	τ_p	0.031	τ_e	0.354	τ_f	0.031	0.139
30	τ_p	0.031	τ_e	0.354	τ_e	0.281	0.222
31	τ_p	0.031	τ_e	0.354	τ_p	0.031	0.139
32	τ_p	0.031	τ_p	0.125	τ_f	0.031	0.062
33	τ_p	0.031	τ_p	0.125	τ_e	0.281	0.146

TABLE E.11 – S_4 – Ce qu’Anna pense être l’influence des autres agents sur l’importance des buts pour chaque agent.

La valeur $\delta_{a,y_i}(\gamma)$ représente l’influence de l’agent y_i sur l’importance accordée au but γ par Anna. La valeur $\Delta_a(\gamma)$ représente l’influence de l’ensemble des agents sur l’importance accordée au but γ par Anna. La valeur $\delta_{y_i,y_j}^a(\gamma)$ représente l’influence de l’agent y_j sur l’importance accordée au but γ par y_i selon Anna. $\Delta_{y_i}^a(\gamma)$ représente l’influence de l’ensemble des agents sur l’importance accordée au but γ par y_i selon Anna.

	<i>monter les meubles</i>	<i>installer info.</i>	<i>formaliser le modèle</i>	<i>écrire un papier</i>
Pour Anna				
Bea	—	$\delta_{a,b}(\gamma_e) = 0.0$	—	$\delta_{a,b}(\gamma_p) \simeq 0.062$
Cara	$\delta_{a,c}(\gamma_m) = -0.125$	$\delta_{a,c}(\gamma_e) = -0.125$	—	—
Dora	$\delta_{a,d}(\gamma_m) = 0.0$	$\delta_{a,d}(\gamma_e) = 0.0$	—	—
	$\Delta_a(\gamma_m) \simeq -0.062$	$\Delta_a(\gamma_e) \simeq -0.042$	$\Delta_a(\gamma_f) = 0.0$	$\Delta_a(\gamma_p) \simeq 0.062$
Pour Béa				
Anna	$\delta_{b,a}^a(\gamma_m) \simeq 0.188$	$\delta_{b,a}^a(\gamma_e) = 0.0$	$\delta_{b,a}^a(\gamma_f) \simeq 0.062$	—
Cara	$\delta_{b,c}^a(\gamma_m) = 0.125$	$\delta_{b,c}^a(\gamma_e) = -0.125$	—	—
Dora	$\delta_{b,d}^a(\gamma_m) \simeq 0.188$	$\delta_{b,d}^a(\gamma_e) = 0.0$	—	—
	$\Delta_b^a(\gamma_m) \simeq 0.167$	$\Delta_b^a(\gamma_e) \simeq -0.042$	$\Delta_b^a(\gamma_f) \simeq 0.062$	$\Delta_b^a(\gamma_p) = 0.0$
Pour Cara				
Anna	$\delta_{c,a}^a(\gamma_m) \simeq 0.062$	$\delta_{c,a}^a(\gamma_e) \simeq 0.062$	$\delta_{c,a}^a(\gamma_f) \simeq 0.062$	—
Bea	—	$\delta_{c,b}^a(\gamma_e) \simeq 0.062$	—	$\delta_{c,b}^a(\gamma_p) \simeq 0.062$
Dora	$\delta_{c,d}^a(\gamma_m) \simeq 0.062$	$\delta_{c,d}^a(\gamma_e) \simeq 0.062$	—	—
	$\Delta_c^a(\gamma_m) \simeq 0.062$	$\Delta_c^a(\gamma_e) \simeq 0.062$	$\Delta_c^a(\gamma_f) \simeq 0.062$	$\Delta_c^a(\gamma_p) \simeq 0.062$
Pour Dora				
Anna	$\delta_{d,a}^a(\gamma_m) = 0.0$	$\delta_{d,a}^a(\gamma_e) = 0.0$	$\delta_{d,a}^a(\gamma_f) \simeq 0.062$	—
Bea	—	$\delta_{d,b}^a(\gamma_e) = 0.0$	—	$\delta_{d,b}^a(\gamma_p) \simeq 0.062$
Cara	$\delta_{d,c}^a(\gamma_m) = -0.125$	$\delta_{d,c}^a(\gamma_e) = -0.125$	—	—
	$\Delta_d^a(\gamma_m) \simeq -0.062$	$\Delta_d^a(\gamma_e) \simeq -0.042$	$\Delta_d^a(\gamma_f) \simeq 0.062$	$\Delta_d^a(\gamma_p) \simeq 0.062$

TABLE E.12 – S_4 – Valeurs d’importance finale accordée par Béa, Cara et Dora aux buts *monter les meubles*, *installer l’équipement informatique*, *écrire un papier* et *formaliser le modèle* selon Anna.

	<i>monter les meubles</i>	<i>installer info.</i>	<i>formaliser le modèle</i>	<i>écrire un papier</i>
Anna	$fImp_a(\gamma_m) \simeq 0.688$	$fImp_a(\gamma_e) \simeq 0.708$	$fImp_a(\gamma_f) = 0.25$	$fImp_a(\gamma_p) \simeq 0.062$
Bea	$fImp_b^a(\gamma_m) \simeq 0.167$	$fImp_b^a(\gamma_e) \simeq 0.708$	$fImp_b^a(\gamma_f) \simeq 0.062$	$fImp_b^a(\gamma_p) = 0.25$
Cara	$fImp_c^a(\gamma_m) \simeq 0.562$	$fImp_c^a(\gamma_e) \simeq 0.562$	$fImp_c^a(\gamma_f) \simeq 0.062$	$fImp_c^a(\gamma_p) \simeq 0.062$
Dora	$fImp_d^a(\gamma_m) \simeq 0.688$	$fImp_d^a(\gamma_e) \simeq 0.708$	$fImp_d^a(\gamma_f) \simeq 0.062$	$fImp_d^a(\gamma_p) \simeq 0.062$

E.5 Scénario S5

Nous donnons ici les détails des calculs effectués par le moteur décisionnel de l'agent Anna dans le scénario S5 décrit dans la section 7.2.4 du chapitre 7.

E.5.1 Sélection d'une tâche

Le tableau E.14 présente les différentes répartitions de tâches générées par Anna.

TABLE E.14 – S5 – Tableau des répartitions de tâches générées par Anna.

Les combinaisons où des agents ne faisaient rien ont été éliminées pour limiter la taille du tableau (ces combinaisons ayant une utilité moindre puisque l'utilité de ne rien faire est nulle).

La ligne 34(répartition $R_{a,34}$) correspond à la répartition pour laquelle l'utilité est maximale.

j	Anna		Cara		Dora		$U_a(R_{a,j})$
	$\tau_{k_{a,j}}$	$U_a(\tau_{k_{a,j}})$	$\tau_{k_{c,j}}$	$U_c^a(\tau_{k_{c,j}})$	$\tau_{k_{d,j}}$	$U_d^a(\tau_{k_{d,j}})$	
1	τ_f	0.125	τ_f	0.031	τ_p	0.062	0.073
2	τ_f	0.125	τ_f	0.031	τ_e	0.167	0.108
3	τ_f	0.125	τ_f	0.031	τ_m	0.344	0.167
4	τ_f	0.125	τ_p	0.062	τ_f	0.031	0.073
5	τ_f	0.125	τ_p	0.062	τ_p	0.062	0.083
6	τ_f	0.125	τ_p	0.062	τ_e	0.167	0.118
7	τ_f	0.125	τ_p	0.062	τ_m	0.344	0.177
8	τ_f	0.125	τ_m	0.281	τ_f	0.031	0.146
9	τ_f	0.125	τ_m	0.281	τ_p	0.062	0.156
10	τ_f	0.125	τ_m	0.281	τ_e	0.167	0.191
11	τ_f	0.125	τ_m	0.281	τ_m	0.344	0.25
12	τ_p	0.062	τ_f	0.031	τ_f	0.031	0.042
13	τ_p	0.062	τ_f	0.031	τ_p	0.062	0.052
14	τ_p	0.062	τ_f	0.031	τ_e	0.167	0.087
15	τ_p	0.062	τ_f	0.031	τ_m	0.344	0.146
16	τ_p	0.062	τ_p	0.062	τ_f	0.031	0.052
17	τ_p	0.062	τ_p	0.062	τ_e	0.167	0.097
18	τ_p	0.062	τ_p	0.062	τ_m	0.344	0.156
19	τ_p	0.062	τ_m	0.281	τ_f	0.031	0.125
20	τ_p	0.062	τ_m	0.281	τ_p	0.062	0.135
21	τ_p	0.062	τ_m	0.281	τ_e	0.167	0.17
22	τ_p	0.062	τ_m	0.281	τ_m	0.344	0.229
23	τ_e	0.333	τ_f	0.031	τ_f	0.031	0.132
24	τ_e	0.333	τ_f	0.031	τ_p	0.062	0.142

Continue sur la page suivante

	Anna		Cara		Dora		
j	$\tau_{k_a,j}$	$U_a(\tau_{k_a,j})$	$\tau_{k_c,j}$	$U_c^a(\tau_{k_c,j})$	$\tau_{k_d,j}$	$U_d^a(\tau_{k_d,j})$	$U_a(R_{a,j})$
25	τ_e	0.333	τ_f	0.031	τ_e	0.167	0.177
26	τ_e	0.333	τ_f	0.031	τ_m	0.344	0.236
27	τ_e	0.333	τ_p	0.062	τ_f	0.031	0.142
28	τ_e	0.333	τ_p	0.062	τ_p	0.062	0.153
29	τ_e	0.333	τ_p	0.062	τ_e	0.167	0.188
30	τ_e	0.333	τ_p	0.062	τ_m	0.344	0.247
31	τ_e	0.333	τ_m	0.281	τ_f	0.031	0.215
32	τ_e	0.333	τ_m	0.281	τ_p	0.062	0.226
33	τ_e	0.333	τ_m	0.281	τ_e	0.167	0.26
34	τ_e	0.333	τ_m	0.281	τ_m	0.344	0.319
35	τ_m	0.344	τ_f	0.031	τ_f	0.031	0.135
36	τ_m	0.344	τ_f	0.031	τ_p	0.062	0.146
37	τ_m	0.344	τ_f	0.031	τ_e	0.167	0.181
38	τ_m	0.344	τ_f	0.031	τ_m	0.344	0.24
39	τ_m	0.344	τ_p	0.062	τ_f	0.031	0.146
40	τ_m	0.344	τ_p	0.062	τ_p	0.062	0.156
41	τ_m	0.344	τ_p	0.062	τ_e	0.167	0.191
42	τ_m	0.344	τ_p	0.062	τ_m	0.344	0.25
43	τ_m	0.344	τ_m	0.281	τ_f	0.031	0.219
44	τ_m	0.344	τ_m	0.281	τ_p	0.062	0.229
45	τ_m	0.344	τ_m	0.281	τ_e	0.167	0.264

Annexe F

Évaluations préliminaires

Nous avons décrit dans le chapitre 7 les évaluation préliminaires perceptives que nous avons conduites. Nous présentons ici les consignes qui ont été données aux participants.

Le but de cette étude est d'évaluer les comportements d'agents autonomes. Vous allez devoir regarder des vidéos vous présentant, sous forme de graphe, une séquence d'actions effectuées par les agents. Des questions vous seront ensuite posées sur les comportements des agents observés dans la vidéo. L'étude prend environ 10 minutes.

Lecture du graphe représentant les comportements des agents

Dans le graphe que vous serez amenés à observer sur la vidéo, une ligne est associée à chaque action. Les agents sont représentés par des points de couleur, la ligne de couleur représente l'historique des actions des agents. Le temps est représenté en abscisse du graphe. Un exemple de graphe et l'explication de lecture associée sont présentés ci-dessous.

Exemple

Vous observez dans cet exemple le comportement de trois agents, Rose, Marine et Jade, qui peuvent effectuer des tâches ménagères pour que la maison soit en ordre. Ces tâches sont les suivantes : faire les comptes, étendre le linge, préparer le repas, mettre la table, passer l'aspirateur. Préparer le repas est une tâche compliquée, c'est pourquoi il vaut mieux le faire à deux. En plus de ces tâches, les agents peuvent également choisir de se détendre.

Sur l'exemple présenté dans le graphe ci-dessus : *l'agent Rose a d'abord passé l'aspirateur, puis elle a mis la table, pendant que l'agent Marine faisait les comptes. Marine et Rose ont ensuite commencé à préparer le repas ensemble, et sont encore en train de préparer le repas. L'agent Jade a étend le linge.*

Consignes

Vous allez observer trois vidéos et répondre à des questions sur chacune d'elles. Vous pouvez rejouer au temps de fois que vous le désirez chaque vidéo.

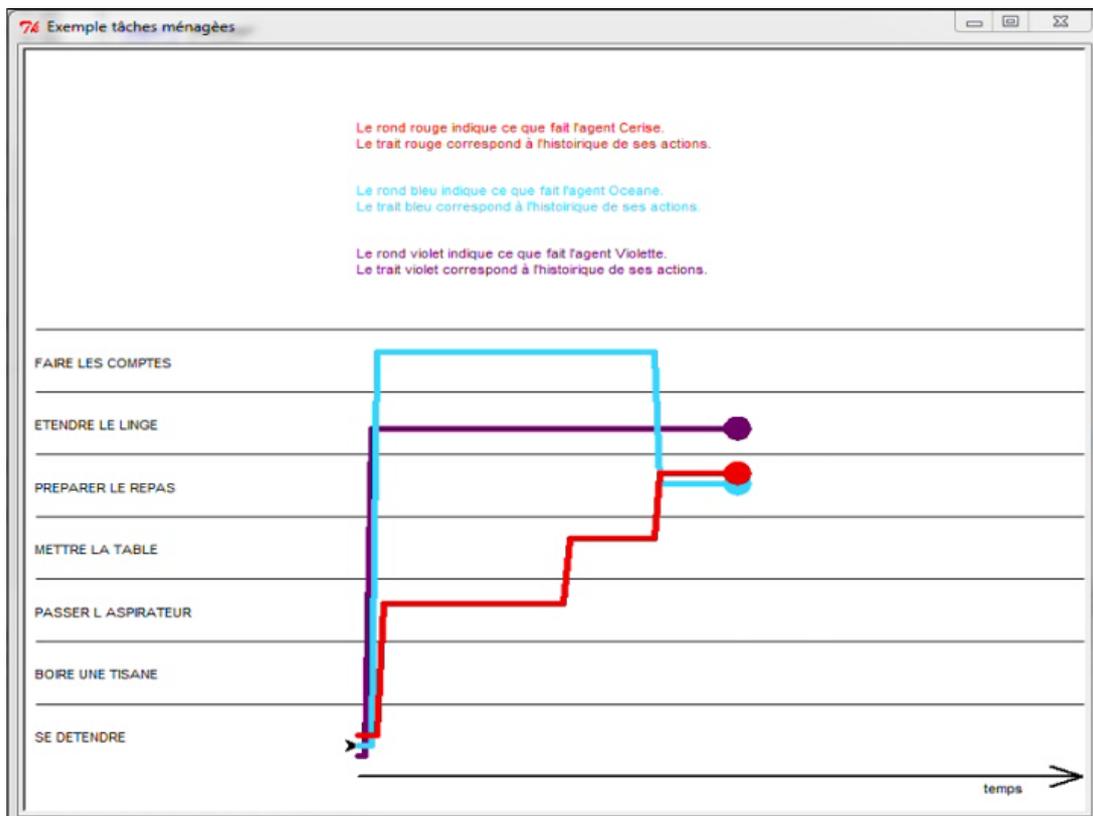


FIGURE F.1 – Exemple d'une répartition de tâche entre agents.

Bibliographie

- Kahina Amokrane. *Suivi de l'apprenant en environnement virtuel pour la formation à la prévention des risques sur des sites Seveso*. Thèse de Doctorat, Université de technologie de Compiègne, 2010. 76
- Carl Anderson and Nigel R Franks. Teamwork in animals, robots, and humans. *Advances in the Study of Behavior*, 33 :1–48, 2003. 2, 3
- Keith Anderson, Elisabeth André, Tobias Baur, Sara Bernardini, Mathieu Chollet, Evi Chrysafidou, Ionut Damian, Cathy Ennis, Arjan Egges, Patrick Gebhard, et al. The TARDIS framework : intelligent virtual agents for social coaching in job interviews. In *Advances in computer entertainment*, pages 476–491. Springer, 2013. 1, 28
- Dimitrios Antos, Celso De Melo, Jonathan Gratch, and Barbara J Grosz. The Influence of Emotion Expression on Perceptions of Trustworthiness in Negotiation. In *AAAI Conference on Artificial Intelligence*. AAAI Press, 2011. 28
- Ruth Aylett, Sandy Louchart, Joao Dias, Ana Paiva, Marco Vala, Sarah Woods, and Lynne Hall. Unscripted narrative for affectively driven characters. *Computer Graphics and Applications, IEEE*, 26(3) :42–52, 2006. 9
- Annette Baier. Trust and antitrust. *Ethics*, 96(2) :231–260, 1986. 58
- Mukesh Barange. *Task-Oriented Communicative Capabilities of Agents in Collaborative Virtual Environments for Training*. Thèse de Doctorat, Université de Bretagne Occidentale (UBO), Brest, 2015. 72
- Mukesh Barange, Alexandre Kabil, and Pierre Chevaillier. The C2BDI agent architecture for teamwork coordination using spoken dialogues between virtual agents and users. In *Advances in Practical Applications of Heterogeneous Multi-Agent Systems*, pages 315–318. Springer, 2014. 11
- Simon Baron-Cohen, Alan M Leslie, and Uta Frith. Does the autistic child have a “theory of mind”? *Cognition*, 21(1) :37–46, 1985. 59
- Camille Barot. *Scénarisation d'environnements virtuels*. Thèse de Doctorat, Université de technologie de Compiègne, 2014. 8, 11, 71, 74, 160
- Camille Barot, Domitile Lourdeaux, J-M Burkhardt, Kahina Amokrane, and Dominique Lenne. V3S : A virtual environment for risk-management training based on human-activity models. *Presence : Teleoperators and Virtual Environments*, 22(1) :1–19, 2013. 1

- Jean-Paul Barthès, Lucile Callebert, and Domitile Lourdeaux. Priority-based contextual local decision making in multi-agent systems. In *Computer Supported Cooperative Work in Design*, 2016. 10
- Timothy Bickmore and Justine Cassell. Relational agents : a model and implementation of building user trust. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 396–403. ACM, 2001. 28
- Timothy Bickmore and Justine Cassell. Social dialogue with embodied conversational agents. In *Advances in natural multimodal dialogue systems*, pages 23–54. Springer, 2005. 28, 29, 30
- Jeanne M Brett. Culture and negotiation. *International Journal of Psychology*, 35(2) :97–104, 2000. 165
- Russell G Brown and James S Jennings. A pusher/steerer model for strongly cooperative mobile robot manipulation. In *International Conference on Intelligent Robots and Systems*, volume 3, pages 562–568. IEEE, 1995. 3
- John K Butler. Trust expectations, information sharing, climate of trust, and negotiation effectiveness and efficiency. *Group & Organization Management*, 24(2) :217–238, 1999. 165
- Janis A Cannon-Bowers, Eduardo Salas, and Sharolyn Converse. Shared mental models in expert team decision making. In *Individual and Group Decision Making : Current Issues*, pages 221–246. N. John Castellan, 1993. 19
- Y Uny Cao, Alex S Fukunaga, and Andrew Kahng. Cooperative mobile robotics : Antecedents and directions. *Autonomous robots*, 4(1) :7–27, 1997. 3
- Javier Carbo, Jose M Molina, and Jorge Davila. Trust management through fuzzy reputation. *International Journal of Cooperative Information Systems*, 12(01) :135–155, 2003. 33
- Jeffrey P Carpenter, Amrita G Daniere, and Lois M Takahashi. Cooperation, trust, and social capital in southeast asian urban slums. *Journal of Economic Behavior & Organization*, 55(4) :533–551, 2004. 53
- Kevin Carpentier. *Scénarisation personnalisée dynamique dans les environnements virtuels pour la formation*. Thèse de Doctorat, Université de technologie de Compiègne, 2015. 8
- Peter Carruthers and Peter K Smith. *Theories of theories of mind*. Cambridge Univ Press, 1996. 59
- Sandro Castaldo. Meanings of trust : a meta-analysis of trust definitions. In *European Academy of Management Conference*, 2002. 56, 57, 59
- Christiano Castelfranchi and Rino Falcone. *Trust theory : A socio-cognitive and computational model*, volume 18. John Wiley & Sons, 2010. xvii, 3, 25, 31, 37, 41, 42, 43, 44, 47, 48, 51, 54, 55, 56, 58, 60, 61, 62, 65, 158
- Cristiano Castelfranchi. Modelling social action for AI agents. *Artificial Intelligence*, 103(1) : 157–182, 1998. 17, 53, 54

- Cristiano Castelfranchi and Rino Falcone. Principles of trust for MAS : Cognitive anatomy, social importance, and quantification. In *Conference on Multi Agent Systems*, pages 72–79. IEEE, 1998. 45, 47
- MO Cavazza, Fred Charles, and Steven J Mead. Character-based interactive storytelling. *IEEE Intelligent systems*, 2002. 8, 73
- Ananish Chaudhuri, Barry Sopher, and Paul Strand. Cooperation in social dilemmas, trust and reciprocity. *Journal of Economic Psychology*, 23(2) :231–249, 2002. 53
- Brian F Chellas. *Modal logic : an introduction*, volume 316. Cambridge Univ Press, 1980. 47
- Pierre Chevaillier, Thanh-hai Trinh, Mukesh Barange, Pierre De Loor, Frédéric Devillers, Julien Soler, and Ronan Querrec. Semantic modeling of virtual environments using MASCARET. In *Software Engineering and Architectures for Realtime Interactive Systems*, pages 1–8. IEEE, 2012. 2, 72
- Jennifer Chu-Carroll and Sandra Carberry. A plan-based model for response generation in collaborative task-oriented dialogues. *arXiv preprint cmp-lg/9405011*, 1994. 165
- Jennifer Chu-Carroll and Sandra Carberry. Conflict resolution in collaborative planning dialogs. *International Journal of Human-Computer Studies*, 53(6) :969–1015, 2000. 165
- Roy YJ Chua, Michael W Morris, and Shira Mor. Collaborating across cultures : Cultural metacognition and affect-based trust in creative collaboration. *Organizational Behavior and Human Decision Processes*, 118(2) :116–131, 2012. 52
- Guillaume Claude, Valérie Gouranton, and Bruno Arnaldi. Roles in collaborative virtual environments for training. In *International Conference on Artificial Reality and Telexistence Eurographics Symposium on Virtual Environments*, pages 1–8, 2015. 2, 10
- Philip R Cohen and Hector J Levesque. Intention is choice with commitment. *Artificial intelligence*, 42(2-3) :213–261, 1990. 12
- Philip R. Cohen and Hector J. Levesque. Teamwork. *Noûs*, 25(4) :487–512, 1991. 12
- Rosaria Conte and Mario Paolucci. *Reputation in artificial societies : Social beliefs for social order*, volume 6. Springer Science & Business Media, 2002. 38
- Mark Core, David Traum, H Chad Lane, William Swartout, Jonathan Gratch, Michael Van Lent, and Stacy Marsella. Teaching negotiation skills through practice and reflection with virtual humans. *Simulation*, 82(11) :685–701, 2006. 165
- Stanislas Couix and Jean-Marie Burkhardt. Task descriptions using academic oriented modelling languages : a survey of actual practices across the SIGCHI community. In *Conference on Human-Computer Interaction*, pages 555–570. Springer, 2011. 73
- Andrew J Cowell and Kay M Stanney. Manipulation of non-verbal interaction style and demographic embodiment to increase anthropomorphic computer character credibility. *International journal of human-computer studies*, 62(2) :281–306, 2005. 28

- James H Davis, F David Schoorman, Roger C Mayer, and Hwee Hoon Tan. The trusted general manager and business unit performance : Empirical evidence of a competitive advantage. *Strategic Management Journal*, 21(5) :563–576, 2000. 52
- Celso M. de Melo, Peter J. Carnevale, and Jonathan Gratch. People’s biased decisions to trust and cooperate with agents that express emotions. In *Autonomous Agents and Multi-Agents Systems*, 2013. 28
- Morton Deutsch. Trust and suspicion. *Journal of conflict resolution*, 2(4) :265–279, 1958. 53, 56, 58, 62, 64, 65
- Joana Dimas and Rui Prada. Dynamic identity model for agents. In *Multi-Agent-Based Simulation*, pages 37–52. Springer, 2013. 20
- Kurt T Dirks. The effects of interpersonal trust on work group performance. *Journal of applied psychology*, 84(3) :445, 1999. 52
- Colin Eden and Fran Ackermann. Group decision and negotiation in strategy making. *Group Decision and Negotiation*, 10(2) :119–140, 2001. 165
- Lydie Edward. *Modélisation décisionnelle de personnages virtuels autonomes évoluant dans un environnement pour la présentation des risques sur les sites SEVESO*. Thèse de Doctorat, Université de Technologie de Compiègne, 2011. 76
- Ferda Erdem. Optimal trust and teamwork : From groupthink to teamthink. *Work Study*, 52(5) :229–233, 2003. 4, 52, 54
- Rino Falcone and Cristiano Castelfranchi. Social trust : A cognitive approach. In *Trust and deception in virtual societies*, pages 55–90. Springer, 2001. 56
- Rino Falcone, Giovanni Pezzulo, and Cristiano Castelfranchi. A fuzzy approach to a belief-based trust computation. In *Workshop on Deception, Fraud and Trust in Agent Societies*, pages 73–86. Springer, 2002. 43
- Xiaocong Fan and John Yen. Modeling and simulating human teamwork behaviors using intelligent agents. *Physics of life reviews*, 1(3) :173–201, 2004. 2, 13, 71
- Matthias Fink and Alexander Kessler. Cooperation, trust and performance – empirical results from three countries. *British Journal of Management*, 21(2) :469–483, 2010. 52
- Christopher D Frith and Rhiannon Corcoran. Exploring ‘theory of mind’ in people with schizophrenia. *Psychological medicine*, 26(03) :521–530, 1996. 59
- Simon Gächter, Benedikt Herrmann, and Christian Thöni. Trust, voluntary cooperation, and socio-economic background : survey and experimental evidence. *Journal of Economic Behavior & Organization*, 55(4) :505–531, 2004. 53
- Diego Gambetta. Can we trust trust? *Trust : Making and breaking cooperative relations*, 13 : 213–237, 2000. 3, 25, 58, 65, 66, 101, 102

- Stéphanie Gerbaud. *Contribution à la formation en réalité virtuelle : scénarios collaboratifs et intégration d'humains virtuels collaborant avec des utilisateurs réels*. Thèse de Doctorat, INSA de Rennes, 2008. 72, 76
- Stéphanie Gerbaud and Bruno Arnaldi. Humains virtuels et collaboration dans un environnement virtuel de formation. *Technique et science informatiques*, 28(6-7) :741–766, 2009. 2, 7
- Stéphanie Gerbaud, Nicolas Mollet, and Bruno Arnaldi. Virtual environments for training : from individual learning to collaboration with humanoids. In *International Conference on Technologies for E-Learning and Digital Entertainment*, pages 116–127. Springer, 2007. 161
- Stéphanie Gerbaud, Nicolas Mollet, Franck Ganier, Bruno Arnaldi, and Jacques Tisseau. GVT : a platform to create virtual environments for procedural training. In *Virtual Reality Conference*, pages 225–232. IEEE, 2008. 1
- Stéphanie Gerbaud, Valérie Gouranton, and Bruno Arnaldi. Adaptation in collaborative virtual environments for training. In *Learning by Playing. Game-based Education System Design and Development*, pages 316–327. Springer, 2009. 10, 11
- Joseph A Giampapa and Katia Sycara. Team-oriented agent coordination in the RETSINA multi-agent system. Technical report, Robotics Institute, Pittsburgh, PA, 2002. 13, 73
- Harjinder Gill, Kathleen Boies, Joan E Finegan, and Jeffrey McNally. Antecedents of trust : Establishing a boundary condition for the relation between propensity to trust and intention to trust. *Journal of business and psychology*, 19(3) :287–302, 2005. 63
- Philip D Gillis and Steven R Hursh. Using behavior moderators to influence cgf command entity effectiveness and performance. In *8th Conference on Computer Generated Forces and Behavioral Representation*, 1999. 21, 22
- Arthur Graesser and Danielle McNamara. Self-regulated learning in learning environments with pedagogical agents that interact in natural language. *Educational Psychologist*, 45(4) :234–244, 2010. 7
- Jones Granatyr, Vanderson Botelho, Otto Robert Lessing, Edson Emílio Scalabrin, Jean-Paul Barthès, and Fabrício Enembreck. Trust and reputation models for multiagent systems. *ACM Computing Surveys*, 48(2) :27, 2015. 31, 32, 35, 37
- Jonathan Gratch and Stacy Marsella. A domain-independent framework for modeling emotion. *Cognitive Systems Research*, 5(4) :269–306, 2004. 23
- Barbara J Grosz and Sarit Kraus. Collaborative plans for complex group action. *Artificial Intelligence*, 86(2) :269–357, 1996. 12, 16
- Barbara J Grosz and Candace L Sidner. Plans for discourse. Technical report, DTIC Document, 1988. 12, 16

- Nicolas Habonneau, Urs Richle, Nicolas Szilas, and Jean E Dumas. 3D simulated interactive drama for teenagers coping with a traumatic brain injury in a parent. In *Interactive Storytelling*, pages 174–182. Springer, 2012. 9
- Mahdi Hannoun, Olivier Boissier, Jaime S Sichman, and Claudette Sayettat. MOISE : An organizational model for multi-agent systems. In *Advances in Artificial Intelligence*, pages 156–165. Springer, 2000. 14
- Maaïke Harbers, Birna van van Riemsdijk, and Catholijn Jonker. Measuring sharedness of mental models and its relation to team performance. In *International Workshop on Coordination, Organisations, Institutions and Norms*, pages 106–120, 2012. 19
- Russell Hardin. *Trust and trustworthiness*. Russell Sage Foundation, 2002. 57, 58, 61
- David Harel, Dexter Kozen, and Jerzy Tiuryn. *Dynamic logic*. MIT press, 2000. 47
- Andreas Herzig, Emiliano Lorini, Jomi Fred Hübner, Jonathan Ben-Naim, Cristiano Castelfranchi, Robert Demolombe, Dominique Longin, Laurent Vercouter, and Olivier Boissier. Prolegomena for a logic of trust and reputation. In *International Workshop on Normative Multiagent Systems*, pages 143–157. Citeseer, 2008. 45
- Andreas Herzig, Emiliano Lorini, Jomi F Hübner, and Laurent Vercouter. A logic of trust and reputation. *Logic Journal of IGPL*, 18(1) :214–244, 2010. 45, 46, 47
- Jean Claude Heudin. A connexionist model for emotions in digital agents. In *International Conference on Agents and Artificial Intelligence*, pages 272–279, 2016. 20
- Randall W Hill Jr, Jonathan Gratch, Stacy Marsella, Jeff Rickel, William R Swartout, and David R Traum. Virtual humans in the mission rehearsal exercise system. *KI*, 17(4) :5, 2003. 9, 11, 23
- Jomi F Hubner, Jaime S Sichman, and Olivier Boissier. Developing organised multiagent systems using the MOISE+ model : Programming issues at the system and agent levels. *International Journal of Agent-Oriented Software Engineering*, 1(3-4) :370–395, 2007. xvii, 2, 14, 15, 161
- Jomi F Hübner, Emiliano Lorini, Andreas Herzig, and Laurent Vercouter. From cognitive trust theories to computational trust. In *International Workshop on Trust in Agent Societies*, pages 2009–11, 2009. 46
- Jomi Fred Hübner, Jaime Simao Sichman, and Olivier Boissier. A model for the structural, functional, and deontic specification of organizations in multiagent systems. In *Advances in artificial intelligence*, pages 118–128. Springer, 2002. 14, 15, 18
- Lauriane Huguet, Domitile Lourdeaux, Nicolas Sabouret, and Marie-Hélène Ferrer. Perturbed communication in a virtual environment to train medical team leaders. In *Medicine Meets Virtual Reality Conference*, pages 146–149. IOS Press, 2016. 1
- Trung Dong Huynh, Nicholas R Jennings, and Nigel R Shadbolt. An integrated trust and reputation model for open multi-agent systems. *Autonomous Agents and Multiagent Systems*, 13(2) :119–154, 2006. 34

- Irving L Janis and Leon Mann. *Decision making : A psychological analysis of conflict, choice, and commitment*. Free Press, 1977. 21, 22
- Sirkka L Jarvenpaa, Kathleen Knoll, and Dorothy E Leidner. Is anybody out there? Antecedents of trust in global virtual teams. *Journal of management information systems*, 14(4) :29–64, 1998. 63
- Nicholas R Jennings. Controlling cooperative problem solving in industrial multi-agent systems using joint intentions. *Artificial intelligence*, 75(2) :195–240, 1995. 13
- W Lewis Johnson. Pedagogical agent research at CARTE. *AI Magazine*, 22(4) :85, 2001. 7
- W Lewis Johnson and Jeff Rickel. Steve : An animated pedagogical agent for procedural training in virtual environments. *ACM SIGART Bulletin*, 8(1-4) :16–21, 1997. 7, 10
- W Lewis Johnson, Erin Shaw, Andrew Marshall, and Catherine LaBore. Evolution of user interaction : The case of agent Adele. In *International Conference on Intelligent user interfaces*, pages 93–100. ACM, 2003. 7
- Gareth R Jones and Jennifer M George. The experience and evolution of trust : Implications for cooperation and teamwork. *Academy of management review*, 23(3) :531–546, 1998. 25
- Karen Jones. Trust as an affective attitude. *Ethics*, 107(1) :4–25, 1996. 56, 60, 61
- Laurent Karsenty. Comment faire confiance dans les situations à risques. *Société d’Ergonomie de Langue Française*, pages 205–212, 2010. 63, 64, 102
- Herbert W Kee and Robert E Knox. Conceptual and methodological considerations in the study of trust and suspicion. *Journal of Conflict Resolution*, 14(3) :357–366, 1970. 53
- Patrick Kenny, Arno Hartholt, Jonathan Gratch, William Swartout, David Traum, Stacy Marsella, and Diane Piepol. Building interactive virtual humans for training environments. In *Interservice/Industry Training, Simulation and Education Conference*, 2007. 4, 8
- Norbert L Kerr. Motivation losses in small groups : A social dilemma analysis. *Journal of Personality and Social Psychology*, 45(4) :819, 1983. 3, 108, 111, 141
- Vincent Lanquepin, Kevin Carpentier, Domitile Lourdeaux, Margaux Lhommet, Camille Barot, and Kahina Amokrane. HUMANS : a HUMAN models based Artificial eNvironments Software platform. In *Virtual Reality International Conference : Laval Virtual*, pages 59–68. ACM, 2013. 10
- Hector J Levesque, Philip R Cohen, and José HT Nunes. On acting together. In *AAAI National conference on Artificial intelligence*, pages 94–99. AAAI Press, 1990. 12
- Roy J Lewicki, Daniel J McAllister, and Robert J Bies. Trust and distrust : New relationships and realities. *Academy of management Review*, 23(3) :438–458, 1998. 58, 65, 66
- Margaux Lhommet. *Replicants : humains virtuels cognitifs, émotionnels et sociaux. De l’empathie cognitive à l’empathie affective*. Thèse de Doctorat, Université de technologie de Compiègne, 2012. 10, 20

- Karen E Lochbaum, Barbara J Grosz, and Candace L Sidner. Models of plans to support communication : An initial report. In *AAAI National Conference on Artificial Intelligence*, pages 485–490. AAAI Press / MIT Press, 1990. 12, 119
- Emiliano Lorini and Robert Demolombe. From binary trust to graded trust in information sources : a logical perspective. In *Trust in agent societies*, pages 205–225. Springer, 2008a. 45, 46, 47
- Emiliano Lorini and Robert Demolombe. Trust and norms in the context of computer security : A logical formalization. In *Deontic logic in computer science*, pages 50–64. Springer, 2008b. 45
- Gehao Lu, Joan Lu, Shaowen Yao, and Yau Jim Yip. A review on computational trust models for multi-agent systems. *The open information science journal*, 2 :18–25, 2009. 31
- Niklas Luhmann. Familiarity, confidence, trust : Problems and alternatives. *Trust : Making and breaking cooperative relations*, 6 :94–107, 2000. 65
- Andrés Saraos Luna, Valérie Gouranton, and Bruno Arnaldi. Collaborative virtual environments for training : A unified interaction model for real humans and virtual humans. In *E-Learning and Games for Training, Education, Health and Sports*, pages 1–12. Springer, 2012. 7
- Lyuba Mancheva and Julie Dugdale. Understanding communications in medical emergency situations. In *Hawaii International Conference on System Sciences*, pages 198–206, 2016. 19
- Jacques Marc and René Amalberti. Contribution individuelle à la sécurité du collectif : l'exemple de la régulation du samu. *Le travail humain*, 65(3) :217–242, 2002. 2
- Stacy Marsella, Jonathan Gratch, Ning Wang, and Brooke Stankovic. Assessing the validity of a computational model of emotional coping. In *International Conference on Affective Computing and Intelligent Interaction*, pages 1–8. IEEE, 2009. 23
- Stacy C Marsella and Jonathan Gratch. EMA : A process model of appraisal dynamics. *Cognitive Systems Research*, 10(1) :70–90, 2009. 23, 25
- Stacy C Marsella, David V Pynadath, and Stephen J Read. PsychSim : Agent-based modeling of social interactions and influence. In *International Conference on Cognitive Modeling*, volume 36, pages 243–248. Citeseer, 2004. 29, 165
- Stephen Marsh and Pamela Briggs. Examining trust, forgiveness and regret as computational concepts. In *Computing with social trust*, pages 9–43. Springer, 2009. 55, 65
- Stephen Paul Marsh. *Formalising trust as a computational concept*. Thèse de Doctorat, University of Stirling, 1994. xi, 32, 33, 34, 35, 36
- John E Mathieu, Tonia S Heffner, Gerald F Goodwin, Eduardo Salas, and Janis A Cannon-Bowers. The influence of shared mental models on team process and performance. *Journal of applied psychology*, 85(2) :273, 2000. 19

- Roger C Mayer and Mark B Gavin. Trust in management and performance : Who minds the shop while the employees watch the boss? *Academy of Management Journal*, 48(5) :874–888, 2005. 52
- Roger C Mayer, James H Davis, and F David Schoorman. An integrative model of organizational trust. *Academy of management review*, 20(3) :709–734, 1995. iii, v, 56, 57, 58, 59, 60, 62, 63, 64, 65, 66, 67, 77, 91, 99, 100, 101, 102, 158, 164
- Robert R McCrae and Oliver P John. An introduction to the five-factor model and its applications. *Journal of personality*, 60(2) :175–215, 1992. 20
- Lynne McFall. Integrity. *Ethics*, 98(1) :5–20, 1987. 102
- D Harrison McKnight and Norman L Chervany. Trust and distrust definitions : One bite at a time. In *Trust in Cyber-societies*, pages 27–54. Springer, 2001. 52, 55, 57, 58, 59, 62, 64, 65
- Maria Miceli and Cristiano Castelfranchi. *Expectancy and emotion*. OUP Oxford, 2014. 56
- Aneil K Mishra. Organizational responses to crisis. *Trust in Organizations : Frontiers of theory and research*, pages 261–287, 1996. 57, 62, 63, 64
- Nicolas Mollet and Bruno Arnaldi. Storytelling in virtual reality for training. In *Technologies for E-Learning and Digital Entertainment*, pages 334–347. Springer, 2006. 10
- Bradford W Mott and James C Lester. U-DIRECTOR : A decision-theoretic narrative planning architecture for storytelling environments. In *International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 977–984. ACM, 2006. 8
- Stefanos Mouzas, Stephan Henneberg, and Peter Naudé. Trust and reliance in business relationships. *European Journal of Marketing*, 41(9/10) :1016–1032, 2007. 57
- Guillaume Muller and Laurent Vercoeur. Decentralized monitoring of agent communications with a reputation model. In *Trusting Agents for Trusting Electronic Societies*, pages 144–161. Springer, 2005. 32, 35
- Zeinab Noorian, Stephen Marsh, and Michael Fleming. Multi-layer cognitive filtering by behavioral modeling. In *Conference on Autonomous Agents and Multiagent Systems*, pages 871–878. International Foundation for Autonomous Agents and Multiagent Systems, 2011. 36, 37
- Jay F Nunamaker, Douglas C Derrick, Aaron C Elkins, Judee K Burgoon, and Mark W Patton. Embodied conversational agent-based kiosk for automated interviewing. *Journal of Management Information Systems*, 28(1) :17–48, 2011. 28
- Benjamin D Nye and Barry G Silverman. Social learning and adoption of new behavior in a virtual agent society. *Presence*, 22(2) :110–140, 2013. 22
- Magalie Ochs, Nicolas Sabouret, and Vincent Corruble. Simulation de la dynamique des émotions et des relations sociales de personnages virtuels. *Revue d'intelligence artificielle*, 23(2-3) :327–357, 2009. 20

- Andrew Ortony, Gerald L Clore, and Allan Collins. *The cognitive structure of emotions*. Cambridge university press, 1990. 21
- Michael E Palanski, Surinder S Kahai, and Francis J Yammarino. Team virtues and performance : An examination of transparency, behavioral integrity, and trust. *Journal of Business Ethics*, 99(2) :201–216, 2011. 52
- Craig D Parks and Lorne G Hulbert. High and low trusters’ responses to fear in a payoff matrix. *Journal of conflict resolution*, 39(4) :718–730, 1995. 53
- Isaac Pinyol and Jordi Sabater-Mir. Pragmatic-strategic reputation-based decisions in BDI agents. In *International Conference on Autonomous Agents and Multiagent Systems*, pages 1001–1008. International Foundation for Autonomous Agents and Multiagent Systems, 2009. 41
- Isaac Pinyol and Jordi Sabater-Mir. Computational trust and reputation models for open multi-agent systems : a review. *Artificial Intelligence Review*, 40(1) :1–25, 2013. 31, 32, 36, 37, 38, 48, 49
- Isaac Pinyol, Jordi Sabater-Mir, Pilar Dellunde, and Mario Paolucci. Reputation-based decisions for logic-based cognitive agents. *Autonomous Agents and Multiagent Systems*, 24(1) :175–216, 2012. 37, 38, 41
- David Pizzi and Marc Cavazza. Affective storytelling based on characters’ feelings. In *AAAI Fall Symposium on Intelligent Narrative Technologies*, pages 111–118, 2007. 9, 73
- Mark Plovnick, Ronald Fry, and Irwin Rubin. New developments in OD [organisational development] technology : Programmed team development. *Training & Development Journal*, 29(4) :19–25, 1975. 2
- Julie Porteous, Fred Charles, and Marc Cavazza. Networking : using character relationships for interactive narrative generation. In *International Conference on Autonomous Agents and Multiagent Systems*, pages 595–602. International Foundation for Autonomous Agents and Multiagent Systems, 2013. 73
- David Premack and Guy Woodruff. Does the chimpanzee have a theory of mind? *Behavioral and brain sciences*, 1(04) :515–526, 1978. 59
- Ronan Querrec, Cédric Buche, Eric Maffre, and Pierre Chevaillier. SécuRéVi : virtual environments for fire-fighting training. In *Virtual Reality International Conference*, pages 169–175, 2003. 2
- Sarvapali D Ramchurn, Dong Huynh, and Nicholas R Jennings. Trust in multi-agent systems. *The Knowledge Engineering Review*, 19(01) :1–25, 2004. 31, 32, 36
- Anand S Rao and Michael P Georgeff. Modeling rational agents within a BDI-architecture. *KR*, 91 :473–484, 1991. 115

- Jeff Rickel and W Lewis Johnson. Animated agents for procedural training in virtual reality : Perception, cognition, and motor control. *Applied artificial intelligence*, 13(4-5) :343–382, 1999. 73
- Jeff Rickel, Jonathan Gratch, Randall Hill, Stacy Marsella, and William Swartout. Steve goes to Bosnia : Towards a new generation of virtual humans for interactive experiences. In *AAAI spring symposium on Artificial Intelligence and Interactive Entertainment*, volume 11, 2001. 7
- Albert Rizzo, J Galen Buckwalter, Bruce John, Brad Newman, Thomas Parsons, Patrick Kenny, and Josh Williams. STRIVE : Stress resilience in virtual environments : a pre-deployment VR system for training emotional coping skills and assessing chronic and acute stress responses. *Studies in health technology and informatics*, 173 :379–385, 2011. 1
- Julian B Rotter. A new scale for the measurement of interpersonal trust. *Journal of personality*, 35(4) :651–665, 1967. 57
- Denise M Rousseau, Sim B Sitkin, Ronald S Burt, and Colin Camerer. Not so different after all : A cross-discipline view of trust. *Academy of management review*, 23(3) :393–404, 1998. 57, 58, 60, 65
- Jordi Sabater and Carles Sierra. REGRET : A reputation model for gregarious societies. In *Fourth workshop on deception fraud and trust in agent societies*, volume 70, 2001a. 32, 34
- Jordi Sabater and Carles Sierra. Social regret, a reputation model based on social relations. *ACM SIGecom Exchanges*, 3(1) :44–56, 2001b. 34
- Jordi Sabater and Carles Sierra. Review on computational trust and reputation models. *Artificial intelligence review*, 24(1) :33–60, 2005. 31, 32, 37
- Jordi Sabater, Mario Paolucci, and Rosaria Conte. Repage : Reputation and image among limited autonomous partners. *Journal of artificial societies and social simulation*, 9(2), 2006. 37, 38, 40
- Maxime Sanselone, Santiago Sanchez, Cedric Sanza, David Panzoli, and Yves Duthen. Constrained control of non-playing characters using Monte Carlo Tree search. In *Computational Intelligence and Games*, pages 1–8. IEEE, 2014. 9, 72
- F David Schoorman, Roger C Mayer, and James H Davis. An integrative model of organizational trust : Past, present, and future. *Academy of Management review*, 32(2) :344–354, 2007. 65, 66
- Susan P Shapiro. The social control of impersonal trust. *American journal of Sociology*, pages 623–658, 1987. 56
- Sharad Sharma and Stephen Otunba. Collaborative virtual environment to study aircraft evacuation for training and education. In *International Conference on Collaboration Technologies and Systems*, pages 569–574. IEEE, 2012. 7, 8

- Mei Si and Joseph Dean McDaniel. Establish trust and express attitude for a non-humanoid robot. In *Conference of the Cognitive Science Society*, 2016. 28
- Mei Si, Stacy Marsella, and David Pynadath. Importance of well-motivated characters in interactive narratives : An empirical evaluation. In *Interactive Storytelling*, pages 16–25. Springer, 2010. 4, 7
- Jaime Simao Sichman, Yves Demazeau, Rosaria Conte, and Cristiano Castelfranchi. A social reasoning mechanism based on dependence networks. In *European Conference on Artificial Intelligence*, pages 416–420, 1998. 17, 18
- Barry G Silverman, Michael Johns, Ransom Weaver, Kevin O’Brien, and Rachel Silverman. Human behavior models for game-theoretic agents : Case of crowd tipping. *Cognitive Science Quarterly*, 2(3/4) :273–301, 2002. 22
- Barry G Silverman, Gnana Bharathy, Kevin O’Brien, and Jason Cornwell. Human behavior models for agents in simulators and games ; part 2 : gamebot engineering with PMFserv. *Presence : Teleoperators and Virtual Environments*, 15(2) :163–185, 2006a. 22
- Barry G Silverman, Michael Johns, Jason Cornwell, and Kevin O’Brien. Human behavior models for agents in simulators and games ; part 1 : enabling science with PMFserv. *Presence : Teleoperators and Virtual Environments*, 15(2) :139–162, 2006b. xvii, 20, 21, 22
- Barry G Silverman, David Pietrocola, Ben Nye, Nathan Weyer, Oleg Osin, Dan Johnson, and Ransom Weaver. Rich socio-cognitive agents for immersive training environments : case of NonKin Village. *Autonomous Agents and Multiagent Systems*, 24(2) :312–343, 2012. 22
- Barry G. Silverman, Nancy Hanrahan, Lina Huang, Emilia Flores Rabinowitz, and Samuel Lim. Chapter 7 : Artificial intelligence and human behavior modeling and simulation for mental health conditions. In David D. Luxton, editor, *Artificial Intelligence in Behavioral and Mental Health Care*, pages 163 – 183. Academic Press, 2016. 22
- Sim B Sitkin and Nancy L Roth. Explaining the limited effectiveness of legalistic “remedies” for trust/distrust. *Organization science*, 4(3) :367–392, 1993. 62
- Jeroen Stouten, David De Cremer, and Eric van Dijk. Behavioral (in)tolerance of equality violation in social dilemmas : When trust affects contribution decisions after violations of equality. *Group processes & intergroup relations*, 12(4) :517–531, 2009. 53
- William Swartout, David Traum, Ron Artstein, Dan Noren, Paul Debevec, Kerry Bronnenkant, Josh Williams, Anton Leuski, Shrikanth Narayanan, Diane Piepol, et al. Ada and Grace : Toward realistic and engaging virtual museum guides. In *Intelligent Virtual Agents*, pages 286–300. Springer, 2010. 28
- Katia Sycara and Gita Sukthankar. Literature review of teamwork models. *Robotics Institute, Carnegie Mellon University*, 2006. 13
- Nicolas Szilas. A computational model of an intelligent narrator for interactive narratives. *Applied Artificial Intelligence*, 21(8) :753–801, 2007. 9, 72

- Henri Tajfel and John C Turner. An integrative theory of intergroup conflict. *The social psychology of intergroup relations*, 33(47) :74, 1979. 20
- Milind Tambe. Towards flexible teamwork. *Journal of artificial intelligence research*, 7 :83–124, 1997. xvii, 2, 16, 17
- Leigh L Thompson, Jiunwen Wang, and Brian C Gunia. Negotiation. *Annual review of psychology*, 61 :491–515, 2010. 165
- David Traum, Jeff Rickel, Jonathan Gratch, and Stacy Marsella. Negotiation over tasks in hybrid human-agent teams for simulation-based training. In *International joint conference on Autonomous Agents and Multiagent Systems*, pages 441–448. ACM, 2003. 165
- David Traum, William Swartout, Stacy Marsella, and Jonathan Gratch. Fight, flight, or negotiate : Believable strategies for conversing under crisis. In *Intelligent Virtual Agents*, pages 52–64. Springer, 2005. 29, 165
- David Traum, Stacy C Marsella, Jonathan Gratch, Jina Lee, and Arno Hartholt. Multi-party, multi-issue, multi-strategy negotiation for multi-modal virtual agents. In *Intelligent Virtual Agents*, pages 117–130. Springer, 2008. 30, 165
- Andrew C Wicks, Shawn L Berman, and Thomas M Jones. The structure of optimal trust : Moral and strategic implications. *Academy of Management review*, 24(1) :99–116, 1999. 55
- Peter S Wong, Sai On Cheung, and Peter K Ho. Contractor as trust initiator in construction partnering—prisoner’s dilemma perspective. *Journal of Construction Engineering and Management*, 131(10) :1045–1053, 2005. 53
- Toshio Yamagishi and Midori Yamagishi. Trust and commitment in the United States and Japan. *Motivation and emotion*, 18(2) :129–166, 1994. 64
- John Yen, Jianwen Yin, Thomas R Ioerger, Michael S Miller, Dianxiang Xu, and Richard A Volz. CAST : Collaborative agents for simulating teamwork. In *International Joint Conference on Artificial Intelligence*, volume 17, pages 1135–1144, 2001. 13
- Han Yu, Zhiqi Shen, Clement Leung, Chunyan Miao, and Victor R Lesser. A survey of multi-agent trust management systems. *Access, IEEE*, 1 :35–50, 2013. 31, 32, 36, 37