Stage de recherche du Master Web Intelligence
effectué à l'École Nationale Supérieure des Mines de Saint-Étienne

Centre de recherche G2I
Génie Industriel et Informatique

Département SMA
Systèmes Multi-Agents

Intégration de normes dans un système multi-agent décentralisé et ouvert

Amandine Grizard
amandine.grizard@c2m.univ-st-etienne.fr

Encadrants
Guillaume Muller
Tiberiu Stratulat
Laurent Vercouter

Table des matières

1 Introduction **1**

1.1 Cadre d'étude 1

1.2 Objectifs 1

1.3 Plan du rapport 2

2 Les normes pour les systèmes multi-agents **3**

2.1 Introduction aux systèmes multi-agents 3

2.1.1 Les agents logiciels 3

2.1.2 Les systèmes multi-agents ouverts et décentralisés 3

2.2 Introduction aux normes 4

2.2.1 Les normes dans les sciences sociales 4

2.2.2 Les normes dans les sciences juridiques 5

2.2.3 Les normes en informatique 5

2.3 Les normes dans les systèmes multi-agents 6

2.3.1 Les normes sociales 6

2.3.2 Agents normatifs 7

2.4 Sanctions associées aux normes 8

2.4.1 La direction d'une sanction 8

2.4.2 Le type de sanctions 8

2.4.3 Définition des sanctions 9

2.5 Conclusion 9

3 Système normatif fondé sur la réputation permettant une régulation décentralisée 11

3.1 Modèle de normes 11

3.1.1 Modélisation 11

3.1.2 Les obligations 12

3.1.3 Les normes sociales 12

3.1.4 Processus de détection de respect et de non respect des obligations . 13

3.1.5 Application des sanctions en terme de réputation 13

3.2 Mise en oeuvre 14

3.2.1 Les agents applicatifs 14

3.2.2 Les agents contrôleurs 15

3.2.3 Interaction entre un agent applicatif et un agent contrôleur 15

3.2.4 Exemple 16

3.3 Conclusion 18

4 Adaptation et émergence de normes **19**
4.1 Émergence de normes . . . 19
4.1.1 Définition de l'émergence . . . 19
4.1.2 Types d'émergences . . . 19
4.2 Adaptation de normes . . . 21
4.2.1 Définition de l'adaptation . . . 21
4.2.2 Évolution en normes sociales . . . 21
4.2.3 Évolution des normes sociales en obligations . . . 22
4.3 Conclusion . . . 23

5 Application au partage de fichiers dans des réseaux P2P **25**
5.1 Cadre d'étude . . . 25
5.1.1 Réseau P2P et Gnutella . . . 25
5.1.2 Description du simulateur . . . 26
5.1.3 Intégration des normes dans un système de type Gnutella . . . 27
5.2 Fonctionnement de l'application et objectif . . . 27
5.3 Le langage de communication . . . 28
5.3.1 Syntaxe du langage . . . 28
5.3.2 Description du langage . . . 28
5.4 Définition des normes . . . 29
5.4.1 La recherche de fichier . . . 29
5.4.2 La modification du TTL et du TTL max . . . 30
5.4.3 La demande d'un fichier . . . 30
5.5 Comportement des agents . . . 32
5.6 Résultats obtenus . . . 33
5.6.1 Réseau à 6 agents . . . 33
5.6.2 Réseau à 35 agents . . . 34
5.6.3 Remarque . . . 34
5.7 Conclusion . . . 34

6 Conclusion **41**
6.1 Synthèse . . . 41
6.2 Limites . . . 42
6.3 Perspectives . . . 42

Table des figures

2.1 Les différentes approches des normes et leur sanction 10

3.1 Implémentation du processus de respect et de non respect des obligations . 13
3.2 Exemple de scénario 16
3.3 Scénario étapes par étapes 17

4.1 Processus d'évolution 22

5.1 Recherche de fichiers et propagation du TTL dans un réseau Gnutella . . . 26
5.2 Protocole de recherche d'un fichier 29
5.3 Protocole de demande de modification du TTL 31
5.4 Protocole de demande de modification du TTL seuil 31
5.5 Protocole de demande d'un fichier 32
5.6 Evolution d'un réseau à 6 noeuds 36
5.7 Résultats sur un réseau à 6 noeuds 37
5.8 Résultats sur un réseau à 35 noeuds 38
5.9 Réseau initial à 35 noeuds 39
5.10 Réseau final à 35 noeuds 40

Résumé

Ce rapport a été écrit à la suite de mon stage de Master en juillet 2005. Il va porter sur l'intégration et l'émergence de normes dans les systèmes multi-agents

À l'origine, ce stage devait rentrer dans les travaux de thèse de Guillaume Muller afin de trouver une formalisation des normes de bon comportement prenant en compte la notion de réputation des agents. Après de nombreuses discussions, les travaux de ce stage se sont plus tournés vers de l'adaptation de normes.

Le rapport est organisé de la façon suivante : le premier chapitre est une introduction générale et est suivi d'un état de l'art des travaux existants sur les normes et les sanctions. Les chapitres 3 et 4 présentent notre modèle, le chapitre 5 l'illustre avec une application. Le chapitre 6 conclut le rapport.

Remerciements : Je tiens à remercier Guillaume Muller, Tiberiu Stratulat et Laurent Vercouter pour leur patience et le temps qu'ils m'ont consacré, ainsi que toute l'équipe SMA de l'école de Mines des Saint-Étienne et l'équipe EURISE de l'université Jean Monnet de Saint-Étienne.

Chapitre 1

Introduction

Dans cette introduction, nous allons placer le cadre d'étude du stage ainsi que ces objectifs. Pour finir nous donnerons un rapide plan du rapport.

1.1 Cadre d'étude

Nous nous plaçons dans un système multi-agent décentralisé et ouvert où des agents hétérogènes peuvent rentrer et sortir à tout moment. L'hétérogénéité réside dans le fait que les agents peuvent avoir des comportements différents qui parfois peuvent nuire au bon fonctionnement du système, soit parce que les agents sont malicieux, soit parce qu'ils possèdent des moyens de communication différents. Pour détecter ces agents malicieux, il est nécessaire de pouvoir définir ce que sont de bons et de mauvais comportements. La solution est de définir des normes que les agents devraient respecter et dont les violations seraient sanctionnées. L'intégration de normes dans un tel système va permettre aux agents de connaître les lois qui le régissent et est un moyen de contrôle plus souple que les protocoles. En effet, si les agents sont capables d'interpréter les normes, il sera facile de les modifier afin de les adapter à l'évolution de l'environnement, c'est ce que nous appelons de l'émergence de normes. L'autre intérêt de la présence de normes dans un système ouvert et décentralisé est que leur violation est sanctionnée, c'est-à-dire que les agents, pour éviter les conséquences de trop nombreuses pénalités, auront tendance à respecter les normes. Si les sanctions influent sur la réputation des agents, ceux dotés d'intentions malveillantes en auraient une mauvaise. De ce fait, ces agents malicieux pourraient être perçue par les autres agents. De plus, les agents malicieux seraient reconnus par les autres.

1.2 Objectifs

Les objectifs du stage décrit dans ce rapport sont de trouver un modèle de normes et une architecture d'un système prenant en compte ces normes. Un processus d'émergence de normes est également présenté, plus particulièrement un processus d'adaptation de normes qui va permettre de modifier et de faire évoluer le système. Ces normes, et en particulier les sanctions qui y sont associées, vont utiliser la notion de réputation, liée à la confiance pour fiabiliser le système. Nous montrons l'intérêt d'intégrer les normes dans un système multi-agent ouvert et décentralisé au travers d'une petite application. Cette dernière illustre la reconfiguration d'un réseau suite à l'utilisation de normes.

1.3 Plan du rapport

Le deuxième chapitre explique rapidement de qu'est un système multi-agent ouvert et décentralisé et présente ensuite les différentes approches des normes dans de nombreux domaines et en particulier dans celui des systèmes multi-agents. Il donne également un rapide aperçu des différentes sanctions possibles. Ensuite, nous présentons dans le chapitre 3 notre modèle permettant d'intégrer les normes dans un système normatif et donnons une formalisation pour les normes et une architecture du système. Le chapitre suivant va expliquer la notion d'émergence de normes et comment elle est présente dans notre modèle. Nous finirons par une application illustrant le modèle que nous avons défini. Le dernier chapitre résume et conclut les chapitres précédents.

Chapitre 2

Les normes pour les systèmes multi-agents

Le chapitre présente comment la notion de normes peut être introduite et utilisée dans un système multi-agent. Après un bref résumé de ce qu'est un système multi-agent, nous nous intéressons au concept de norme tel qu'il est étudié en sciences humaines puis dans des travaux existants dans le domaine des systèmes multi-agents. Nous détaillons enfin la notion de sanction particulièrement utile à notre travail avant de conclure par une synthèse.

2.1 Introduction aux systèmes multi-agents

2.1.1 Les agents logiciels

"Un agent est une entité capable de contrôler ses activités de raisonnement et de décision, de perception et d'action sur l'environnement, de communication et de gestion des relations avec les autres agents" [Bois 05].

Les agents ont des propriétés particulières comme l'autonomie ou la sociabilité. Un agent est autonome s'il est capable de prendre des initiatives seul, sans intervention extérieur. L'autonomie signifie aussi qu'il n'est pas possible qu'une entité extérieure à l'agent exerce un quelconque contrôle direct sur celui-ci. Un agent est social s'il interagit avec les autres agents.

2.1.2 Les systèmes multi-agents ouverts et décentralisés

Un système multi-agent est constitué d'agents en interaction au sein d'un même environnement ou avec l'extérieur. C'est un ensemble organisé d'agents.

Ferber [Ferb 95] définit un système multi-agent de la façon suivante : "un agent dans un univers multi-agent peut communiquer avec d'autres agents, son comportement est la conséquence de ses observations, de ses connaissances et des interactions avec les autres agents. Nous considérons ici des systèmes multi-agents décentralisés et ouverts".

Un système décentralisé est un système où le contrôle n'est pas affecté à un unique point central, chaque agent a une connaissance partielle de son environnement et des autres agents. Un système multi-agent peut être également ouvert, ce qui signifie que des agents peuvent y rentrer et en sortir dynamiquement.

Associée à la propriété d'autonomie des agents, l'ouverture implique que des agents hétérogènes co-existent dans le système, ce qui peut inclure des agents défectueux, voire

malicieux. Ces agents malicieux pourraient, par exemple, ne pas respecter les protocoles d'interactions en vigueur dans le système.

Un système multi-agent, pour assurer son bon fonctionnement, doit pouvoir être en mesure de se protéger contre l'introduction d'agents malicieux. Dans la suite de ce chapitre, nous présentons comment des normes sociales peuvent être utilisées pour décrire les limites des comportements acceptables des agents, ce qui permet de détecter les violations de ces limites et donc des agents malicieux.

2.2 Introduction aux normes

"Une norme est une règle à laquelle nous devons nous conformer." [1]

Les normes sont présentes dans de nombreux domaines tels que les sciences sociales, les sciences juridiques, l'informatique... Chacun de ces domaines propose un concept de norme différent. En sociologie, une norme est équivalente à une règle ; en philosophie, elle sert de référence pour juger quelque chose ; en économie, c'est un modèle devant exister ou devant être suivi [Dign 02].

Dans sa thèse [Stra 02], STRATULAT donne des précisions sur les normes en fonction du contexte.

2.2.1 Les normes dans les sciences sociales

Les normes sociales sont liées à un certain type de comportement. Elles sont aussi appelées normes de communauté. TUOMELA [Tuom 95a] distingue les règles (r-normes) des normes sociales (s-normes). Il décrit également l'existence de normes potentiellement sociales comme les normes morales (m-normes) et les normes de prudence (p-normes).

Les règles ou r-normes

Une r-norme est une règle, pouvant prendre la forme de lois ou de chartes, à laquelle des individus doivent se conformer. Elle est exprimée explicitement.

Les r-normes sont fondées sur la notion d'autorité et d'acceptation, elle impose les r-normes à un ensemble d'agents qu'elle contrôle. Les r-normes sont créées par un groupe pour lui-même. Des accords peuvent en être à l'origine.

Ces règles entraînent des sanctions si elles ne sont pas respectées : les r-sanctions. Les règles formelles, donnant lieu à des sanctions formelles, sont connues de tous les membres du groupe. Les punitions peuvent être économiques ou physiques, ou des sanctions légales allant jusqu'au retrait de la liberté.

Les normes sociales ou s-normes

Une norme sociale représente un comportement global de la société.

Les s-normes sont fondées sur des croyances ou des acceptations mutuelles. Elles ne sont pas obligatoirement disponibles sous forme écrite. TUOMELA donne l'exemple d'une s-norme dans [Tuom 95a] qui est qu'un homme doit tenir la porte à une femme. Il donne aussi un exemple de norme spécifique à un groupe : dans certaines religions, il faut s'agenouiller pour prier.

Les s-normes sont de deux types : celles appliquées à toute la communauté et celles appliquées uniquement à un groupe d'individus en fonction de leur rôle dans la société. Elles sont enseignées aux nouveaux arrivants.

[1] Dictionnaire de la langue française

La violation de s-normes donne lieu à des s-sanctions : le violateur peut par exemple être expulsé du groupe.

Les normes morales ou m-normes

Une norme morale est une norme créée par la conscience personnelle d'un individu. Il se l'impose à lui-même. Un exemple de m-norme est que nous ne devons pas voler dans des circonstances normales.

Une m-norme peut devenir sociale (s-norme) si des membres du groupe demandent l'obéissance à cette norme. Une m-norme peut également se transformer en r-norme si elle est acceptée par un législateur ou une autorité pour la collectivité [Tuom 95b].

Les normes de prudence ou p-normes

Une norme de prudence est une norme créée par un individu pour son propre intérêt.

Les p-normes sont fondées sur les choses rationnelles à effectuer. Agir de façon à ne pas risquer notre vie est un exemple d'une p-norme.

De la même façon que les m-normes, les p-normes peuvent devenir des r-normes ou des s-normes.

TUOMELA fait des hypothèses empiriques sur ces différents types de normes :

- les r-normes sont plus importantes que les autres normes, elles sont "prioritaires" ;
- des instances de chaque sorte de normes peuvent être prioritaires sur les autres suivant certaines conditions, par exemple les p-normes dans des situations de conflits.

2.2.2 Les normes dans les sciences juridiques

Ces normes définissent les droits et les obligations des individus par rapport à leur rôle dans la société. Elles peuvent être similaires aux r-normes [Stra 02]. Les droits sont classés en deux groupes : déontiques et normatifs.

Les droits déontiques regroupent le droit, la liberté et le devoir. Les droits normatifs quant à eux, regroupent le pouvoir, l'immunité, l'incapacité et la responsabilité.

2.2.3 Les normes en informatique

Une norme indique les moyens pour atteindre un but [Stra 02].

Une norme informatique est un document établi par consensus et approuvé par un organisme de normalisation reconnu (ISO, CEI, UIT-T, ETSI...). Le contenu des normes est connu.

Les normes vont permettre un comportement cohérent du système. Elles sont déjà plus ou moins présentes à travers des règles de comportement, des protocoles, des lois, des politiques. Elles sont utilisées dans :

- les politiques de sécurité (pour garantir la confidentialité, l'intégrité, et la disponibilité des systèmes d'informations) ;
- le management des systèmes distribués ;
- le contrôle des activités des agents.

2.3 Les normes dans les systèmes multi-agents

Une des caractéristiques d'un agent est l'autonomie. De ce fait, il n'est pas possible d'avoir des garanties sur le comportement d'un agent. Par exemple, lorsqu'il interagit avec d'autres agents, un agent est habituellement supposé suivre un protocole d'interactions, mais il pourrait s'en écarter parce qu'il ne le connaît pas, ne le comprend pas ou a décidé rationnellement de ne pas le suivre. Plusieurs travaux proposent d'utiliser la notion de normes pour contrôler de manière flexible le comportement des agents.

Dans [Lope 02], López y López et al. définissent les normes comme un mécanisme de la société visant à influencer le comportement des agents. Elles représentent les actions d'un ensemble d'agents devant être bénéfiques à la société, au groupe. La construction de ces normes se fait par simple accord entre les agents ou avec des systèmes légaux plus complexes. Leur applicabilité dépend du contexte dans lequel les agents se trouvent. Ce sont les agents hétérogènes, pouvant appartenir à des sociétés différentes avec des rôles différents, qui sont capables d'adopter des normes.

Les agents hétérogènes ont des buts différents, leur comportement va être propre à chacun d'entre d'eux. Il est donc nécessaire de faire en sorte que le comportement de chacun converge vers un même comportement global. L'intégration de normes dans le système va permettre cette convergence de comportement. Pour Conte et al. [Cont 99], les normes permettent d'améliorer la coordination et la coopération entre les agents. Mais ces normes peuvent être en contradiction avec les buts d'un agent, et de part son autonomie, il peut choisir l'action qu'il va effectuer. La violation d'une norme serait une solution à la résolution de conflits.

Les normes peuvent être décrites explicitement à l'aide de la logique déontique [Wrig 51]. Cette logique est un outil de modélisation pour les notions d'obligation, d'interdiction et de permission. Elle va permettre également de définir la violation (et donc les sanctions), l'adoption ou l'adhésion à une norme.

Un agent autonome peut agir en fonction de deux choses. La première est la normativité : l'agent devra avoir un comportement conforme à une règle établie. La seconde est la rationalité : l'agent agira en fonction de ses connaissances et de ses intentions. Lorsqu'un agent se trouve dans une situation de conflits, c'est-à-dire que le but qu'il doit atteindre est en contradiction avec une règle, il doit choisir entre respecter la règle (agir pour la communauté) ou bien arriver à son but (agir pour lui-même). Il fait ce choix en sachant qu'une sanction négative lui sera appliquée en cas de violation de la règle. Elle peut être sous forme financière, une modification de la réputation, une exclusion du système... Il est difficile pour un agent de réagir aux violations des conventions des autres agents. Dans ce cas, la définition de normes et de sanctions est très utile.

Dans la suite nous allons décrire les normes sociales et donner la définition d'un agent normatif.

2.3.1 Les normes sociales

Les normes sociales sont des règles influençant le comportement social des agents. Dans le suite de la section, nous parlons de normes pour décriré des normes sociales. Elles sont partagées par tous les agents du système et permettent de standardiser le comportement des agents. Ainsi, la connaissance de ces normes améliore la coordination entre les agents pour anticiper les réactions des autres. Dans [Dign 00], Dignum fait la distinction entre les obligations et les normes. Les obligations définissent des règles à respecter, et à chacune d'entres elles est associée une conséquence. À l'inverse, les normes définissent des règles

de bon comportement, elles rendent la coordination entre les agents plus efficace. Les normes sociales restreignent donc l'autonomie de l'agent et permettent de contrôler son comportement.

Pour mieux expliciter la différence entre ces deux notions, prenons l'exemple se trouvant dans [Dign 00]. Dans une entreprise, la plupart des personnes arrivent à 9h le matin car elles accompagnent leurs enfants à l'école à 8h30. Si le directeur veut faire une réunion, il sait qu'elle devra commencer après 9h. La norme dans cette entreprise est de commencer à 9h. Si une personne arrive à 10h, elle ne sera pas pénalisée par son patron. Si cette personne n'assiste pas à la réunion, elle se pénalisera seule, mais ce n'est pas en fonction d'elle que se fera l'emploi du temps. Par contre, si le directeur imposait à ses employés d'arriver pour 9h, ce serait une obligation et toute personne en retard serait pénalisée. Par conséquent, les normes se mettent en place suite à des observations du comportement de la société, elles peuvent être modifiées en fonction de son changement de comportement. Les obligations sont des choses établies donnant lieu à des pénalités si elles ne sont pas respectées.

Dans [Dign 00], DIGNUM incorpore les influences sociales (les normes et les obligations) dans une architecture BDI [Rao 91] afin que les agents les prennent en compte lors du choix de leur comportement. Il sera donc déterminé en fonction des normes, des obligations et de ses buts propres. Les agents ont un raisonnement flexible et approprié à l'environnement grâce à la représentation explicite des normes et des obligations.

Un agent peut ne pas respecter une norme ou une obligation dans un but privé. Par exemple, dans [Dign 99], il est interdit d'effacer des fichiers, pourtant si l'agent en rencontre un infecté par un virus, il sera obligé de le supprimer. Dans ce même papier, Dignum détermine trois niveaux dans le comportement social d'un agent :

1. **Le niveau des conventions**
   Les conventions sont fixées au démarrage du système. Elles sont donc exprimées de manière explicite et peuvent être facilement modifiées ou changées. Il existe les conventions concrètes qui donnent la définition de certains termes (par exemple un objet à prix raisonnable voudra dire que le prix de cet objet ne dépassera pas plus de 20% son prix le plus bas) et les conventions abstraites qui définissent les obligations, les permissions et les interdictions.
2. **Le niveau des contrats**
   Le niveau des contrats est centré sur des obligations et des autorisations et issu d'interactions entre deux agents. Pour chacune d'elles, il est indiqué d'où elle vient, comment elle est accomplie et les conséquences d'une violation.
3. **Le niveau privé**
   Ce niveau va assurer l'autonomie de l'agent. Si l'agent est influencé uniquement par les deux niveaux précédents, son comportement sera pré-déterminé par les conventions et les contrats, et il ne sera plus autonome.

2.3.2 Agents normatifs

Un agent normatif [Lope 02] est un agent autonome dont le comportement est influencé par un ensemble de normes. Un agent décide d'adopter et de respecter des normes. Cependant il existe une différence entre ces deux notions : adopter signifie en avoir connaissance, respecter signifie en tenir compte. Un agent peut adopter une norme sans pour autant la respecter.

Un agent choisit de violer une norme si l'accomplissement de cette dernière n'est pas compatible avec ses buts ou s'il a des motivations internes à rejeter les ordres externes.

Il ne prend pas en compte une sanction si elle n'a aucun effet sur ses buts, ses intérêts. Offrir une récompense est un moyen de motiver l'agent au respect d'une norme seulement s'il peut en retirer un bénéficede cette récompense.

Un agent normatif peut adopter différentes stratégies :

- *sociale* ⇒ les agents sociaux respectent toujours les normes, au détriment de leurs buts personnels, ils ne sont jamais punis, ils obtiennent un bénéfice maximal suite aux récompenses qu'ils reçoivent ;
- *influençable* ⇒ les agents considèrent les effets des sanctions sur leurs propres buts ;
- *opportuniste* ⇒ les agents tiennent compte des effets des récompenses sur leurs propres buts ;
- *lâche* ⇒ les agents respectent toujours les normes par peur d'être sanctionnés ;
- *ambitieuse* ⇒ les agents veulent obtenir quelque chose même si leurs buts ne sont pas avantagés ;
- *rebelle* ⇒ les agents rejettent toutes les normes.

Pour Conte [Cont 99] un agent est normatif et autonome s'il :

- peut reconnaître ou non une norme comme étant une norme (formation de croyances normatives) ;
- décide si une norme lui convient et s'il l'accepte (dans ce cas la norme devient un but normatif) ;
- a le choix entre obéir ou violer une norme ;
- peut prendre l'initiative d'établir une norme.

Un agent aura un comportement normatif s'il accepte la norme en tant que norme et s'il décide de s'y soumettre.

Un agent, s'il ne respecte pas une norme sera réprimandé. Il existe donc des sanctions associées aux normes que nous détaillons dans la suite.

2.4 Sanctions associées aux normes

Dans les modèles de normes existants dans la littérature, la notion sanction existe mais n'est pas approfondie. Dans ses travaux, Pasquier [Pasq 04] explicite clairement les sanctions mais il les a étudiées pour les engagements sociaux du point de vue de la communication qui peuvent être vus comme des normes individuelles.

Il existe deux types de sanctions : individuelles (appliquées à un individu) et collectives (appliquées à la communauté). Pasquier n'a étudié que les sanctions individuelles. Ces dernières ont trois dimensions : la direction, le type et le style (définition).

2.4.1 La direction d'une sanction

Une sanction peut être positive ou négative.

Une sanction positive est une récompense qui encourage l'individu à garder le même comportement. Une sanction négative est utilisée pour empêcher un comportement qui a permis de violer une norme.

2.4.2 Le type de sanctions

Le premier type de sanctions est la sanction automatique, qui arrive lorsque l'individu se punit tout seul suite à une décision entraînant une violation. Pasquier donne l'exemple

d'un automobiliste, qui roulant en sens inverse, a toutes les chances d'avoir un accident. Les autres sanctions ne sont pas automatiques et il en existe trois sortes :

- *les sanctions matérielles*
  Elles peuvent être représentées par un acte violent ou par une demande d'indemnisation. Dans ce cas, ce pourrait être une action à effectuer ou bien une compensation financière.

- *les sanctions sociales*
  Elles peuvent affecter la réputation et la crédibilité d'un agent. La révélation implicite de certaines informations, où le violateur de l'action exprime une chose le concernant que les autres ne devraient pas savoir, peut donner lieu à cette forme de sanction.

- *les sanctions psychologiques*
  Elle peuvent s'exprimer sous forme de remords (culpabilité) ou de honte. Un violateur pourra culpabiliser à la suite de sa "mauvaise" action à cause de ses connaissances des normes sociales. Il pourra aussi avoir honte de ce qu'il a fait car son action l'a rabaissé, diminué vis à vis de lui même ou des autres.

La durée des sanctions indiquent si leurs effets sont durables ou non. Par exemple, les sanctions sur la réputation, la crédibilité et la confiance ont des conséquences dans le temps, mais ce n'est pas le cas pour les sanctions matérielles immédiates. Cette notion de temps dans les sanctions implique la possibilité d'"oublier" ou de "pardonner" une violation.

2.4.3 Définition des sanctions

Deux styles de sanctions sont distinguées : implicites, propre à chaque individu, et explicites, publiquement connues.

Les sanctions implicites sont décidées par les individus, qui les gèrent de façon autonome. En effet, si le comportement d'un agent ne correspond pas à ce qu'attend un autre agent, alors ce dernier va le pénaliser en fonction de l'importance qu'il accorde à cette violation. Ces sanctions ne sont pas publiques et toute la difficulté pour les agents est de les découvrir.

Les sanctions explicites permettent aux agents de connaître les conséquences d'une violation. Ils peuvent donc raisonner dessus.

2.5 Conclusion

Il est très difficile de donner une définition unique des normes. Comme nous l'avons vu précédemment, la terminologie et le sens donné à cette terminologie sont très variés. TUOMELA [Tuom 95a] distingue quatre types de normes : les règles (r-normes), les normes sociales (s-normes), les normes morales (m-normes) et les normes de prudence (p-normes). Dans le domaine des systèmes multi-agents, il n'y a que les deux premières qui sont reprises. De façon générale, les normes vont permettre d'améliorer le bon fonctionnement du système et d'assurer la coordination et la coopération entre les agents. DIGNUM [Dign 99] utilise plusieurs termes pour désigner les normes sociales : les obligations et les normes. Il propose aussi un modèle à trois niveaux. Il est possible de faire un rapprochement

entre les règles, les obligations et le niveau des conventions. Quelle que soit la définition donnée aux normes, elles sont toujours associées à des sanctions pouvant être implicites, explicites, sociales, matérielles... La principale différence entre les notions de normes, et donc les termes employés, est dans l'explicitation des sanctions. Le tableau 2.1 illustre les similitudes existantes entre les concepts de normes et de sanctions.

		Définition		Sanction	
		explicite	implicite	explicite	implicite
TUOMELA	r-norme	x		x	
	s-norme	x	x		x
	m-norme		x		
	p-norme		x		
DIGNUM	niveau convention	x			
	niveau contrat	x		x	
	obligation	x		x	
	norme		x		x

FIG. 2.1 – Les différentes approches des normes et leur sanction

Dans la suite nous nous intéressons plus particulièrement aux règles (r-normes) et aux normes sociales (s-normes) telles que TUOMELA les définit. Les sanctions que nous utilisons sont des sanctions sociales, explicites ou implicites. Pour cela, nous nous appuierons sur un modèle de réputation tel que décrit dans le chapitre suivant.

Chapitre 3

Système normatif fondé sur la réputation permettant une régulation décentralisée

Dans ce chapitre, nous présentons une proposition de système normatif s'appuyant sur une définition explicite des normes de bon comportement.

Nous nous plaçons dans un système décentralisé, ouvert et normatif où le comportement des agents est influencé par un ensemble de normes, ces dernières contribuant au bon fonctionnement du système. Les agents présents sont autonomes et hétérogènes et peuvent violer les normes, violation entraînant des sanctions sociales en terme de réputation.

Après avoir donné une modélisation des normes, nous verrons la composition et le fonctionnement du système.

3.1 Modèle de normes

Dans la suite, nous proposons une modélisation pour les normes. Celles-ci sont composées d'obligations et de normes sociales. Ces deux notions sont précisées, formalisées et utilisées par un processus de détections de violations.

3.1.1 Modélisation

De nombreux travaux utilisent la logique déontique pour modéliser les normes. Cette logique n'est utile que lorsque les trois conditions suivantes sont réunies :

1. il existe un besoin d'identifier les violations ;
2. les normes doivent être cohérentes entre elles ;
3. à partir d'un ensemble de normes, un autre ensemble peut être généré.

Dans notre système, nous avons seulement besoin d'identifier les violations. Nous faisons l'hypothèse que les normes ne se contredisent pas. Il n'est donc pas nécessaire d'utiliser la logique déontique. Nous formaliserons les normes de manière descriptive.

3.1.2 Les obligations

Définition

"Une obligation est un devoir, une contrainte imposée par des normes morales, des lois sociales." [1]

Une obligation est composée d'une condition, d'une contrainte et de deux sanctions explicites (une positive et une autre négative). Elle indique un fait attendu lorsqu'une condition est remplie et la sanction appliquée si le fait n'apparaît pas (violation). Les sanctions sont propres à chaque obligation.

Formalisation

Nous utilisons la notation suivante pour formaliser des obligations :

$$rule(Condition, Constraint, PositiveSanction, NegativeSanction)$$

Cette formalisation peut être interprétée comme "SI *Condition* ET NON *Constraint* ALORS *NegativeSanction* sinon *PositiveSanction*".
Condition, *Constraint*, *PositiveSanction* et *NegativeSanction* sont des formules logiques. Elles peuvent être atomiques (prédicats) ou complexes (avec des disjonctions, conjonctions). *Condition* est la condition qui doit être vérifiée pour que l'obligation soit vraie, *Constraint* est la contrainte qui doit être vérifiée si la condition est vraie et *Positive Sanction* (respectivement *NegativeSanction*) définie la sanction qui sera appliquée en cas de respect (respectivement non respect) de l'obligation.

$rule(asked\_time(Sender, Receiver), answered\_time(Receiver, Sender, Time), Positive\ Sanction, NegativeSanction)$ correspond par exemple à une obligation de la part de *Receiver* d'envoyer l'heure *Time* à *Sender* quand ce dernier la lui demande, sous peine de recevoir la sanction *NegativeSanction* (*PositiveSanction* s'il le fait). Les deux premiers prédicats correspondent à des actions qui devraient être faites. Des exemples de sanction positive et négative sont donnés à la section 3.1.5.

3.1.3 Les normes sociales

Définition

Une norme sociale est une règle régissant la conduite en société, précisant ce qui est normal ou non et propre à un groupe.

Une norme sociale est composée d'une condition et d'une contrainte. Elle indique l'action normale à effectuer lorsqu'une condition est remplie c'est-à-dire qu'elles représentent le comportement attendu dans une situation précise.

La violation des normes sociales n'entraîne pas nécessairement des sanctions. Si ces sanctions existent, elles sont implicites et déterminées par chaque individu et leur non respect n'a pas pour conséquence un dysfonctionnement du système.

Formalisation

La formalisation des normes sociales est équivalente à celle des obligations, seule l'explicitation de la sanction change :

[1] Définition de l'encyclopédie Larousse

$$norm(Condition, Constraint)$$

Cette formalisation peut être interprétée comme "SI *Condition* ALORS *Constraint*". *Condition* et *Constraint* sont des formules logiques povant être atomiques (prédicats) ou complexes (avec des disjonctions, conjonctions). *Condition* est la condition qui doit être vérifiée pour que la norme sociale soit vraie, *Constraint* est la contrainte qui doit être vérifiée si la condition est vraie.

La violations des normes sociales par les agents peut être sanctionnée pas les autres agents du système.

3.1.4 Processus de détection de respect et de non respect des obligations

Le processus de détection de respect et de non respect des obligations, implémenté à l'aide du langage Prolog (cf tableau 3.1), va permettre de détecter les violations des obligations en fonction des conditions et des actions effectuées qui sont représentées par de prédicats (faits). Il permet aussi de détecter les obligations respectées pour pouvoir ensuite appliquer des sanctions positives.

```
respects(Sanction) :-   findall(Sa, respect(Sa), Sanction).

respect(PositiveSanction) :-   rule(Condition, Constraint, PositiveSanction, NegativeSanction),
                               Condition, Constraint, ! .

respect(NegativeSanction) :-   rule(Condition, Constraint, PositiveSanction, NegativeSanction).
```

FIG. 3.1 – Implémentation du processus de respect et de non respect des obligations

Le prédicat *respects*(*Sanction*) permet de trouver toutes les sanctions (positives et négatives) qui seront appliquées à un agent et utilise un second prédicat *respect*(*Sa*) vérifiant dans un premier temps la véracité de *rule*(*Condition*,*Constraint*,*Positive Sanction*,*NegativeSanction*). Ensuite si la condition *Condition* est remplie, et la contrainte *Constraint* est vérifiée (respectivement ne l'est pas), alors la sanction appliquée sera positive (respectivement négative).

3.1.5 Application des sanctions en terme de réputation

Comme nous l'avons vu dans le chapitre précédent, les sanctions peuvent être positives si les obligations sont respectées et négatives dans le cas contraire. Elles sont appliquées par les agents et sont en terme de réputation permettant ainsi d'instaurer une notion de confiance dans le système. Des travaux sur la confiance existent et en particulier celui de MULLER et al. [Mull 04] sur lequel nous nous appuyons. Le modèle qu'il propose est fondé sur les engagements sociaux et leur respect ce qui permettra aux agents de construire les valeurs de réputations de ceux avec lesquels ils interagissent. Le calcul des valeurs de réputation se fait à partir des expériences directes entre les agents et par une moyenne pondérée du nombre d'engagements respectés ou violés.

Dans notre modèle, les sanctions définissent une valeur d'importance que va prendre le respect de l'obligation, valeur qui sera négative si une violation existe et positive sinon, et permettant de mettre à jour les valeurs de réputation. Une sanction (positive ou négative)

est définie par un prédicat de la forme $sanction(Applier, Sanctioned, Weight)$ où $Applier$ est l'agent appliquant la sanction, $Sanctioned$ l'agent sanctionné et $Weight$ l'importance de la violation de l'obligation. $sanction(ContSender, Receiver, -10)$ est un exemple de sanction, $(NegativeSanction)$ indiquant que $Receiver$ est sanctionné négativement par $ContSender$ avec une importance 10.

Les valeurs d'importance sont mises dans un ensemble $Valeurs_i$ correspondant à un agent i. À partir de cet ensemble, un taux représentant le respect des obligations ($Taux_i$) est calculé, $|Valeur_i|$ représente le cardinal de l'ensemble $Valeur_i$:

$$Taux_i = \frac{1}{|Valeur_i|} \times \sum_{p \in Valeur_i} p$$

Ce taux va être utilisé par un agent j pour modifier la valeur de réputation de l'agent i. Ce taux est positif si l'agent i a, de façon générale, respecté les obligations, et négatif dans le cas contraire. Par exemple, si l'agent j détient un ensemble $Valeur_i = \{-6, -9, 5, 6, -8, 6, 2, 3\}$, alors $Taux_i = -0.625$. L'agent i a commis plus de violations qu'il n'a respecté les obligations, il sera donc sanctionné négativement par l'agent j qui tiendra compte du taux -0.625 et choisira la valeur de réputation de i en fonction de ce taux. j pourra calculer la nouvelle réputation de i en y enlevant par exemple 0.625% s'il considère que le taux de violation n'est pas très important, ou bien $0.625 \times 20 = 12.5\%$ s'il pense le contraire.

Le calcul d'un taux d'importance de respect va permettre aux agents de mettre à jour les valeurs de réputation des autres agents, valeurs qui aideront les agents dans leurs interactions avec les autres en leur confiant des tâches de différentes importances. Si la réputation d'un agent est trop mauvaise, les autres agents ne voudront plus interagir avec lui et dans ce cas sera exclu ce qui permettra une régulation du système.

Les valeurs de réputation sont propres à chaque agent, chacun pouvant percevoir un même agent de différentes façons.

3.2 Mise en oeuvre

Ce système normatif doit assurer que le non respect des obligations est toujours détecté et sanctionné. Dans un système centralisé la présence d'une autorité assure le respect des obligations, mais comme notre système est décentralisé, nous créons des agents contrôleurs jouant le même rôle qu'un agent autorité (détection des violations et application des sanctions).

Nous faisons l'hypothèse que les agents partagent un même langage de communication et un même modèle de normes. Le système comportera des agents applicatifs et des agents contrôleurs.

Dans la suite, dire qu'un agent se trouve dans une **situation** revient à dire qu'une condition est remplie.

3.2.1 Les agents applicatifs

Ces agents ont accès aux normes qui régissent le système. Ils ont des buts à atteindre et doivent tenir compte de ces normes avant de réaliser leurs buts.

Les agents applicatifs étant autonomes, ils décident localement de respecter ou non les normes. Par exemple, ils pourront préférer agir pour eux-mêmes plutôt que pour les

autres. Il est donc nécessaire de détecter les violations des obligations que les agents pourraient commettre et de sanctionner ces derniers. Les agents contrôleurs vont appliquer des sanctions s'ils observent que des obligations n'ont pas été respectées.

3.2.2 Les agents contrôleurs

Les agents contrôleurs perçoivent les agents applicatifs comme des boîtes noires et se contentent d'observer les interactions entre ces agents applicatifs. Ils vont détecter les éventuelles violations et appliquer les sanctions, ils ont donc trois fonctions :

- observer ;
- détecter ;
- sanctionner.

Un agent contrôleur possède une base de connaissance contenant les normes, un processus de détection de respect et de non respect des obligations et un ensemble représentant les valeurs de réputation des agents applicatifs qui seront modifiées suite à l'application de sanctions.

Les sanctions issues du respect ou du non respect des obligations sont appliquées par les agents contrôleurs et non par les agents applicatifs qui, de part leur autonomie, n'administreraient pas forcément les sanctions. De plus un agent contrôleur ne sanctionne jamais son propre agent afin d'éviter que tous les agents du système n'aient la même perception d'un même agent car pour certains elle serait injustifiée.

Un contrôleur peut détecter deux types de violation :

- une violation de la part de son agent, au quel cas il préviendra le contrôleur de l'agent lesé en lui envoyant un message de la forme $inform(Sanction)$;
- une violation de la part d'un autre agent. Dans le cas d'un non respect d'obligation, il appliquera directement la sanction correspondante, et dans celui du non respect de normes sociales, il préviendra son agent.

Les agents contrôleurs peuvent également détecter le non respect des normes sociales en utilisant un processus similaire à celui des obligations puis prévenir leur agent des éventuelles violations.

3.2.3 Interaction entre un agent applicatif et un agent contrôleur

À chaque agent applicatif est assigné un agent contrôleur. Il lui tient à disposition des ressources telles que les obligations et les valeurs de réputation des autres agents et observe également les interactions de son agent avec ces derniers.

Fonctionnement d'un agent applicatif

Un agent applicatif doit atteindre ses buts. En fonction de la situation où il se trouve, il peut être soumis à des obligations. Il en a connaissance sous forme de faits logiques. Lorsqu'il possède une condition, notée $Condition$, il interroge le contrôleur en utilisant un message de la forme $request(Condition)$. Celui-ci lui répond avec un ou plusieurs messages de la forme $inform(Condition, Constraint, PositiveSanction, NegativeSanction)$. L'agent applicatif va alors chercher les actions permettant de rendre $Constraint$ vrai. En raisonnant sur ces actions et sur les sanctions, il va décider localement de son futur comportement, à savoir faire ou non les actions.

Fonctionnement d'un agent contrôleur

Un agent contrôleur observe les interactions de son agent avec les autres agents qui donnent lieu à des conditions que l'agent contrôleur transforme en prédicat. Il interroge alors la base de faits logiques pour connaître les obligations liées à la condition que doit respecter son agent applicatif et donc les actions que celui-ci doit effectuer.

Le contrôleur continue d'observer les actions de son agent et transforme chacune d'entres elles en prédicat. Il utilise ensuite ces prédicats et les prédicats correspondants à la condition et aux obligations afin de détecter d'éventuelles violations grâce au processus de détection de respect et de non respect des obligations.

Il y a un inconvénient et un avantage pour un agent applicatif d'accepter un agent contrôleur. En effet, le contrôleur va observer son agent et va donc détecter les éventuelles violations qu'il pourrait commettre. L'agent applicatif est alors contraint de respecter les normes plus souvent que s'il était seul s'il ne veut pas se faire exclure du système. En revanche, avoir un contrôleur lui permet de ne pas avoir à gérer l'ensemble des valeurs de réputation des autres agents et de ne pas avoir à détecter les éventuelles violations.

3.2.4 Exemple

Prenons deux agents applicatifs A et B, avec leur contrôleur respectif ContA et ContB. Définissons une obligation et le scénario qui devrait se produire (figure 3.2). Les obligations sont contenues dans une base de faits logiques. Chaque agent contrôleur possède cette base.

L'obligation que nous utilisons est la suivante :

$rule(ask\_time(Sender, Receiver), answer\_time(Receiver, Sender, Time),$
$sanction(ContSender, Receiver, 6), sanction(ContSender, Receiver, -10)).$

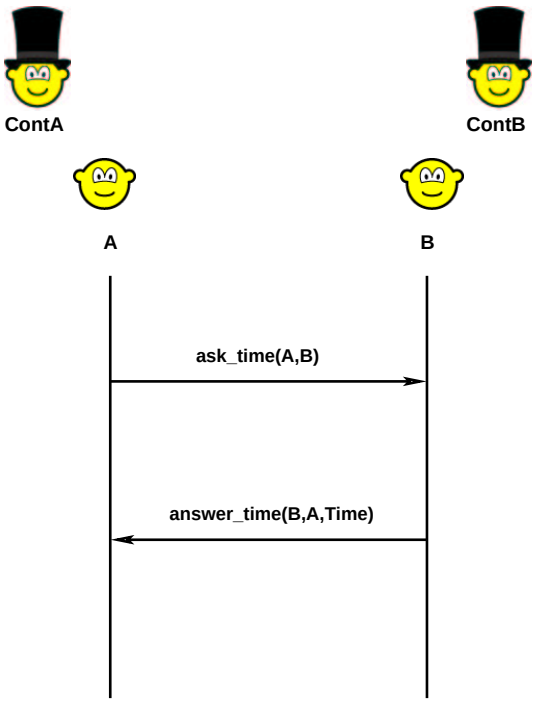

FIG. 3.2 – Exemple de scénario

L'agent A envoie un message $ask\_time(A, B)$ à l'agent B pour lui demander l'heure. Le contrôleur de B observe tous les messages reçus et envoyés. ContB ajoute donc le fait

$asked\_time(A, B)$ dans la base de faits et laisse du temps à B pour répondre.

Selon sa nature, B interrogera son contrôleur en lui envoyant ou non le message $request(ask\_time(A, B))$ et pourra ainsi connaître l'action à effectuer. S'il ne l'envoie pas, il est évident qu'il ne respectera pas l'obligation et sera sanctionné. Cependant, s'il interroge son contrôleur avec le message précédent, ce dernier lui répondra avec un message $inform(ask\_time(A, B), answer\_time(B, A, Time), 10)$.

Une fois que l'agent B possède ces deux nouvelles informations, il sait qu'il a l'obligation d'envoyer un message $answer\_name(B, A, Time)$ sous peine de recevoir une sanction. Comme B est autonome, il peut décider :

- soit de répondre, et dans ce cas l'agent ContB ajoutera dans la base $answered\_name(B, A, 12:00)$;
- soit de ne pas répondre, au quel cas l'agent ContB n'ajoutera rien dans la base.

Une fois le temps de réponse écoulé, ContB lance le processus de détection de violations. Si le fait $answered\_time(B, A, 12:00)$ est vrai, B ne sera pas sanctionné.

Faisons maintenant l'hypothèse que B n'a pas respecté l'obligation, il n'a donc pas donné de réponse à A et $answered\_time(B, A, 12:00)$ est faux. ContB détecte donc la violation et informe ContA de la sanction qui doit être appliquée.

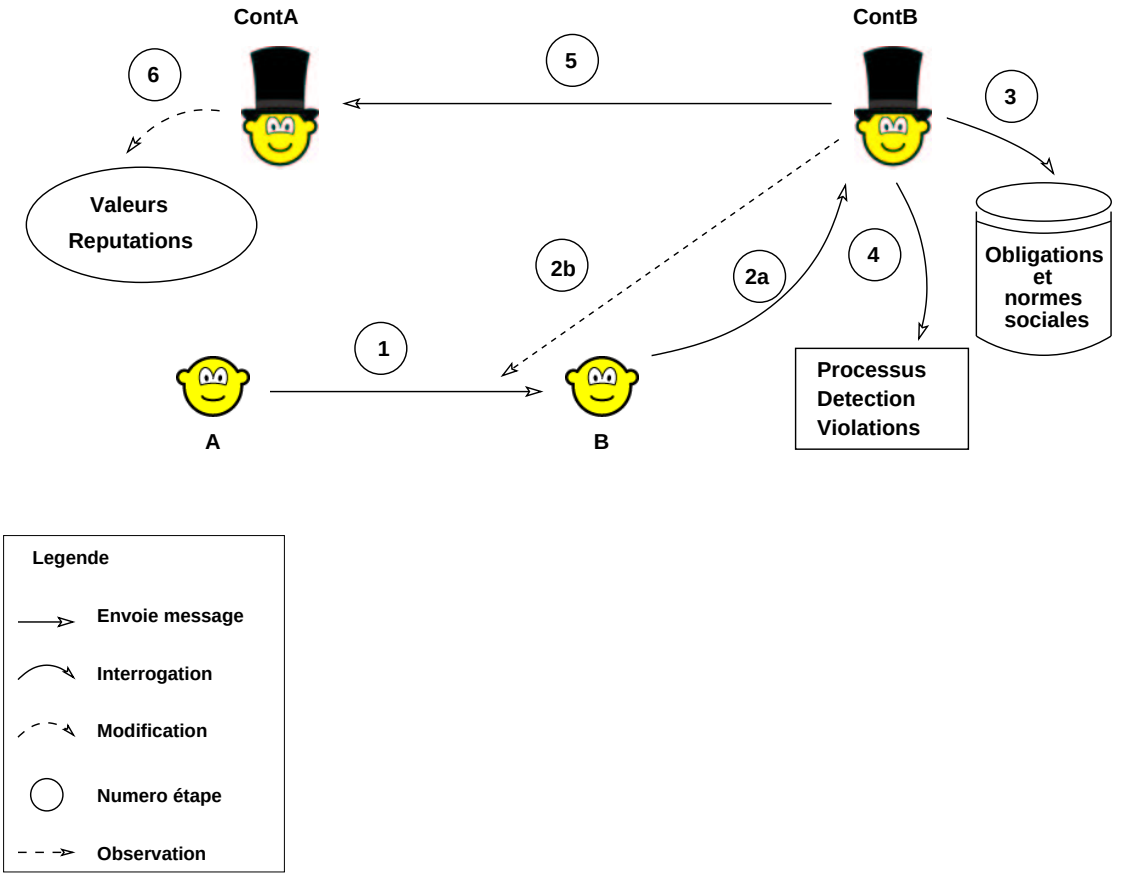

FIG. 3.3 – Scénario étapes par étapes

Reprenons l'exemple. La figure 3.3 illustre le scénario qui se produit si l'agent B ne respecte pas l'obligation.

Étape 1 L'agent A envoie un message $ask\_time(A, B, Time)$ à l'agent B.

Étape 2a B interroge son contrôleur afin de connaître une éventuelle obligation : il doit répondre à A mais ne le fait pas.

Étape 2b ContB observant B, détecte le message et le transforme en prédicat.

Étape 3 ContB interroge la base de faits avec ce prédicat pour connaître les obligations auxquelles sera soumis son agent.

Étape 4 Le temps que B avait pour répondre est écoulé, ContB lance le processus de détection de violations. Il obtient comme réponse $sanction(ContA, B, -10)$, -10 signifiant que l'obligation a une importance de 10 et qu'elle n'a pas été respectée.

Étape 5 ContB envoie $inform(sanction(ContA, B, -10))$ à ContA.

Étape 6 ContA prend en compte ce nouveau poids dans le calcul décrit section 3.1.5 pour modifier la valeur de réputation de B.

3.3 Conclusion

Le système que nous avons défini contient des normes que nous avons modélisées et formalisées. Les obligations comportent quatre paramètres : une condition, une contrainte, une sanction positive et une sanction négative, tandis qu'une norme sociale n'est définie qu'en fonction d'une condition et d'une contrainte. Ceci a été fait de façon suffisament générique pour pouvoir exprimer différentes sortes de normes.

Pour assurer le respect de ces obligations nous avons défini des agents contrôleurs, observant des agents applicatifs, capables de détecter et sanctionner les violations. Ces dernières peuvent être commises par leur agent applicatif ou par les agents qui interagissent avec leur agent applicatif.

Nous avons vu également un mécanisme de sanctions en terme de réputation. Les sanctions indiquent une valeur d'importance associée aux obligations, valeur pouvant être soit négative soit positive. Ces valeurs vont être utilisées en calculant une moyenne pour mettre à jour les valeurs de réputation.

Les contrôleurs laissent à leur agent applicatif toute leur autonomie et leur laissent également choisir l'importance accordée aux poids définis par les sanctions.

Les réputations permettent alors aux agents de détecter les agents malicieux et de les exclure du système permettant ainsi de le réguler.

Dans le chapitre suivant, nous nous intéressons au problème de l'émergence et de l'adaptation de normes afin d'en proposer une solution.

Chapitre 4

Adaptation et émergence de normes

Dans le chapitre précédent, nous donnons une modélisation pour représenter les obligations d'une manière statique, c'est-à-dire que le contenu est fixe. Elles ne peuvent donc pas s'adapter à l'évolution du système par manque de flexibilité. C'est pour cette raison que nous présentons ici des normes sociales adaptées à partir de ces obligations. De ces normes sociales pourront alors émerger de nouvelles obligations.

4.1 Émergence de normes

4.1.1 Définition de l'émergence

L'émergence est l'"apparition inattendue, à partir d'un système complexe, d'un phénomène qui n'avait pas semblé inhérent aux différentes parties de ce système. Ces phénomènes émergents ou collectifs montrent qu'un tout peut être supérieur à la somme de ses parties" [MINS 86]. Elle est définie comme l'apparition d'un fait social dans la société. En particulier, l'émergence de normes est due à un comportement global de la société [Dign 00].

Dans un système multi-agent ouvert, le système peut évoluer en fonction des agents qui y sont présents. Ces agents hétérogènes agissent tous de façons différentes et peuvent ne pas comprendre ce que d'autres attendent d'eux. Les raisons de cette incompréhension pourraient être l'utilisation de moyens de communications différents par exemple. Les agents vont donc utiliser un processus d'apprentissage pour connaître les actions qu'ils doivent faire ce qui conduira à la création de normes communes. L'émergence de normes est liée à un mécanisme de renforcement ainsi qu'à un mécanisme d'apprentissage [Cont 01].

4.1.2 Types d'émergences

Le renforcement

Nous assistons à un renforcement lorsque l'accomplissement d'une action est suivi de l'apparition ou de la disparition d'un stimulus ou d'un événement (agent de renforcement). De plus, il y a renforcement lorsque cette situation augmente la probabilité que ce comportement soit répété. Deux formes de renforcement existent. La première est le renforcement positif, qui consiste à encourager la poursuite d'un comportement, c'est-à-dire de faire en sorte que la probabilité qu'un comportement réapparaisse augmente suite à un stimulus. La seconde, le renforcement négatif, consiste à l'élimination ou le retrait d'un stimulus

suite à l'émission d'un comportement. Le renforcement peut se faire avec des récompenses ou des sanctions.

IONESCU et al. [Ione 04] ont fait des recherches sur l'établissement et le renforcement de politiques communes dans les systèmes P2P (pair à pair). Ils prennent l'exemple d'une communauté fonctionnant comme Gnutella mais qui échange des informations plus délicates et indispensables que des fichiers musicaux. Ils mettent en place un mécanisme de contrôle distribué pour le renforcement évolutif des politiques communes du P2P : LGI (Law Governed Interaction). Dans les systèmes P2P, les échanges s'effectuent directement entre les membres d'une communauté. Si tous ces membres respectent le même protocole, les échanges vont être sûrs. Les buts des politiques régissant un système tel que Gnutella sont d'obtenir une coordination entre les membres du groupe et d'assurer la sécurité des membres de la communauté ainsi que des informations partagées. L'accomplissement de ces buts va donner naissance à des contraintes sur les membres de la communauté et sur leurs comportements lors de leurs interactions.

Il existe deux façons d'établir des politiques communes : soit les membres du groupe sont spontanément d'accord (accord volontaire), soit ces politiques leurs sont imposées. Un accord volontaire n'est fiable que si tout le monde a un intérêt de se conformer à la politique et que si quelqu'un ne s'y conforme pas, les autres ne soient pas affectés.

Imposer une telle politique dans une communauté large et distribuée est difficile. Une solution est la présence d'un médiateur contrôlant toutes les interactions de tous les membres. Ce médiateur va détecter les violations des politiques communes que peut faire son agent et va le sanctionner jusqu'à ce qu'il respecte ces politiques. Elles vont pouvoir être ainsi renforcées pour émerger dans le système.

Pour conclure, un système multi-agent est régi par un ensemble de normes, connues de tous les agents. Ces derniers peuvent violer ces normes pour des raisons qui leur sont propres. Une norme n'émergera dans le système que lorsqu'une majorité d'agents la respectera. Son respect peut être renforcé par l'application de sanctions.

L'apprentissage

L'apprentissage est un processus permettant aux agents d'apprendre ce qu'ils doivent faire suite à des situations inconnues.

Dans [Shoh 97], SHOHAM et TENNENHOLTZ étudient l'émergence des conventions sociales dans des travaux de la théorie des jeux. Les agents interagissent en partageant les ressources du système et se mettent d'accord sur certaines règles afin de réduire les conflits et avoir un comportement coopératif. Pour eux, toutes les règles ne peuvent pas être convenues à l'avance car les caractéristiques de la société changent. Il est donc difficile de créer des règles. Il faudrait que la société converge vers une convention de façon dynamique.

Dans leur modèle, SHOHAM et TENNENHOLTZ expliquent l'émergence de conventions. Des agents individuelles interagissent avec d'autres afin d'obtenir un maximum d'informations. Chaque agent adapte ensuite son comportement en fonction de ce cumul d'informations mais aussi en fonction du comportement d'un autre agent. Il choisit sa stratégie avec une règle de sélection simple et naturel : Highest Cumulative Strategy (HCR). HCR est une règle de mise à jour locale utilisée pour l'émergence. SHOHAM et TENNENHOLTZ donnent l'exemple d'une règle de mise à jour : "Si dans le futur plusieurs choix sont possibles, chacun d'eux est essayé plusieurs fois, celui qui donnera la meilleure récompense sera choisi.". En accord avec HCR, un agent change d'action si et seulement si le total des récompenses obtenues pour cette action dans les m dernières itérations est plus important que celui de l'action habituelle (m étant un seuil défini au départ). L'émergence de conventions vient du fait que les agents ont une règle de sélection de stratégie (HCR), basée sur l'historique

des actions et sur les récompenses reçues correspondantes. Une convention émerge si tous les agents acceptent la nouvelle stratégie.

Pour conclure, l'émergence de normes par apprentissage apparaît lorsque les agents se trouvent face à une situation inconnue qu'ils doivent résoudre. Dans ce cas, les agents vont utiliser un processus d'apprentissage pour connaître l'action à effectuer. Un exemple de processus pourrait être qu'un agent essaie plusieurs actions pour une même situation et prend la meilleure. Il peut alors créer une nouvelle norme à condition qu'il connaisse le formalisme de représentation des normes pour qu'elle puisse être comprise par les autres agents du système.

Nous avons vu deux types d'émergence : par renforcement et par apprentissage. Mais ce ne sont pas les seules. En effet, un agent peut simplement vouloir modifier une norme déjà existante pour l'adapter ses besoins personnels. Le processus d'adaptation de normes est détaillé dans la section suivante.

4.2 Adaptation de normes

Les agents sont soumis à des normes qu'ils doivent respecter mais il est possible que certaines de ces normes deviennent au fil du temps inadaptées à leurs besoins. Il faut alors adapter, dans notre cas modifier, les normes pour atteindre plus facilement les buts personnels des agents.

Nous donnons une définition de ce qu'est une adaptation de norme et quel est le protocole utilisé pour qu'elle devienne une norme sociale.

4.2.1 Définition de l'adaptation

L'adaptation d'une norme est la modification du contenu d'une norme déjà existante dans le but d'en créer une nouvelle. Ce sont les conditions, les contraintes et éventuellement les sanctions qui vont être adaptées par les agents.

4.2.2 Évolution en normes sociales

Lorsqu'un agent adapte une norme à ses besoins, il essaie de la faire accepter par d'autres agents dans le but de la faire émerger. Le problème qui se pose est que tous les agents ne pourront pas être prévenus de ce changement car ils se trouvent dans un système décentralisé. De ce fait des groupes vont se former avec des normes différentes, ces normes seront donc des normes sociales car propre à un groupe.

Pour faire émerger une norme, un agent applicatif interagit avec les autres agents applicatifs qu'il perçoit dans son entourage. Il envoie à chacun d'entre eux un message de la forme $propose(norm(Condition, Constraint))$ et attend leur réponse. Si au bout d'un certain temps, la majorité des agents ne lui a pas répondu positivement, la norme proposée ne deviendra pas une norme sociale. En revanche, si le nombre de réponses positives est majoritaire, l'agent applicatif ajoutera à la base de faits logiques de son contrôleur le prédicat $norm(Condition, Constraint)$. Il diffusera ensuite la nouvelle norme sociale aux autres agents applicatifs de son entourage avec un message de la forme $accepted(norm(Condition, Constraint))$.

Ces normes sociales donneront ainsi à l'agent applicatif des indications sur le comportement attendu en fonction d'une situation donnée. Cependant, la sanction étant implicite, le respect de cette norme n'est pas assuré. Afin d'atteindre plus rapidement ses buts, un agent peut faire évoluer une norme sociale en obligation, ce qui permettra le renforcement de cette norme.

4.2.3 Évolution des normes sociales en obligations

Lorsqu'un agent applicatif décide de faire évoluer une norme sociale en obligation, il en informe tout d'abord son contrôleur car seuls les agents contrôleurs ont la possibilité d'accéder aux obligations.

L'agent contrôleur va alors propager le message aux autres agents contrôleurs qui demanderont à leur tour à leurs agents applicatifs s'ils veulent accepter cette nouvelle obligation. Les agents contrôleurs renverront alors la réponse de leurs agents applicatifs au contrôleur à l'origine de la demande. Ce contrôleur va ensuite analyser tous les messages reçus pour calculer le nombre de réponses positives. Si ce nombre est supérieur à un seuil donné, l'obligation est acceptée et le contrôleur diffuse cette nouvelle obligation aux contrôleurs de son entourage.

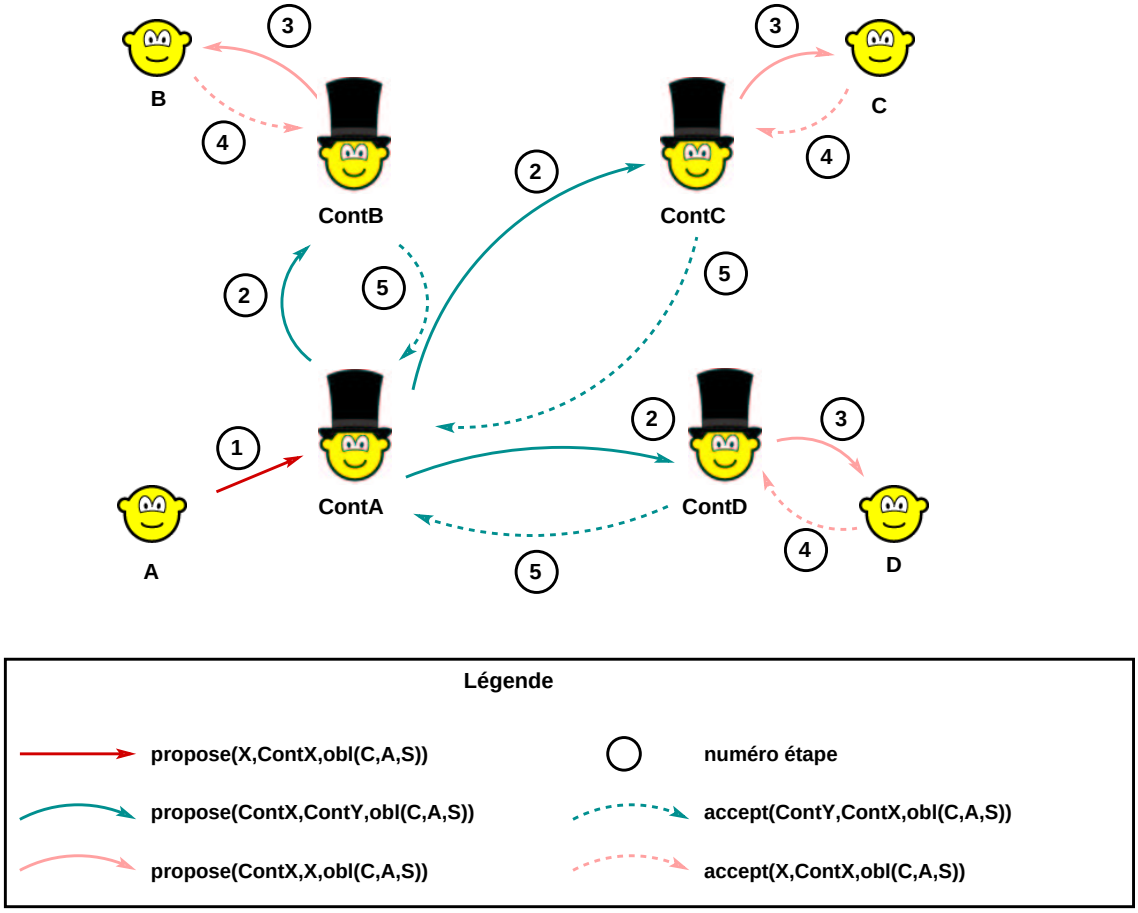

FIG. 4.1 – Processus d'évolution

La figure 4.1 illustre ce processus de demande d'évolution. Il se déroule en cinq étapes :

Étape 1 le message $propose(A, ContA, rule(Condition, Constraint, PositiveSanction, NegativeSanction)$ est envoyé par l'agent A à son contrôleur pour lui proposer une obligation ;

Étape 2 le contrôleur de A propage ce message à tous les autres agents contrôleurs de son entourage ;

Étape 3 chaque contrôleur demande alors à son agent s'il accepte ou non cette nouvelle obligation ;

Étape 4 chaque agent répond ou non à son contrôleur s'il décide d'accepter la nouvelle obligation ;

Étape 5 les contrôleurs transmettent alors à $ContA$ la réponse de leur agent.

$ContA$ compte ensuite le nombre de réponses positives et diffuse la nouvelle obligation si ce nombre est supérieur à un certain seuil.

4.3 Conclusion

Nous avons vu plusieurs types d'émergences : par renforcement, par apprentissage et suite à une adaptation. Ce dernier type va permettre aux agents applicatifs de modifier les normes présentes dans le système en fonction de leurs besoins. Ils vont essayer de faire émerger les nouvelles normes qu'ils ont créées en normes sociales de façon à former des groupes partageant les mêmes buts et leur permettant d'atteindre plus rapidement les leurs. Si ces normes sociales ne sont pas suffisamment respectées, les agents applicatifs vont alors demander à leur contrôleur de les transformer en obligations pour faire en sorte de les renforcer.

Dans le chapitre suivant, nous présentons une application de l'intégration de normes dans un système décentralisé et ouvert dans laquelle les normes n'évoluent qu'en normes sociales.

Chapitre 5

Application au partage de fichiers dans des réseaux P2P

Le modèle agent applicatif/agent contrôleur est illustré à l'aide d'une application au partage de fichiers. Nous positionnons tout d'abord le cadre d'étude, pour ensuite définir des normes dans le formalisme vu précédemment, ainsi que le langage de communication et le formalisme des normes, pour finir en présentant et commentant les résultats obtenus.

5.1 Cadre d'étude

Nous voulons introduire des normes dans un système ouvert et décentralisé en nous appuyant sur un réseau P2P, plus particulièrement sur le réseau Gnutella. Nous donnons une brève description d'un réseau P2P, du protocole Gnutella et du simulateur utilisé pour finir avec les objectifs de notre application.

5.1.1 Réseau P2P et Gnutella

Définition d'un réseau P2P

Les réseaux P2P permettent l'échange de services entre des utilisateurs, principalement l'échange de fichiers. L'architecture de ces réseaux est décentralisée (à l'inverse du modèle client/serveur) et ouverte : un utilisateur peut rentrer à tout moment dans le système.

Il existe plusieurs types de réseaux P2P ne suivant pas les mêmes protocoles comme le réseau eDonkey ou le réseau Gnutella.

Notre application s'appuye sur le réseau Gnutella dont le protocole est décrit par la suite.

Protocole Gnutella

Les machines dans un réseau Gnutella sont appelées des *servents* (**serv**eur/cli**ents**). Les *servents* envoient des trames :

- PING et PONG pour explorer le réseau et découvrir de nouveaux servents ;
- QUERY et QUERYHIT pour la recherche de fichiers.

Les requêtes envoyés ont une durée de vie : Time To Life (TTL). La figure 5.1 illustre une recherche de fichiers. Les étapes sont les suivantes :

Étape 1 le servent contacte ses voisins et leur envoie une requête QUERY ;

Étape 2 les voisins examinent la requête et y répondent s'ils possèdent quelque chose correspondant à la requête ;

Étape 3 les voisins envoient la requête à leurs autres voisins ;

Étape 4 les voisins des voisins refont les étapes 2 et 3 ;

Étape 5 après un certain nombre d'étapes, la propagation de la requête s'arrête car le TTL est arrivé à zéro ;

Étape 6 si le servent initial a reçu des réponses, il peut se connecter au servent possédant le fichier en utilisant une commande $HTTPGET$;

Étape 7 le servent possédant le fichier accepte la connection et répond avec le fichier demandé.

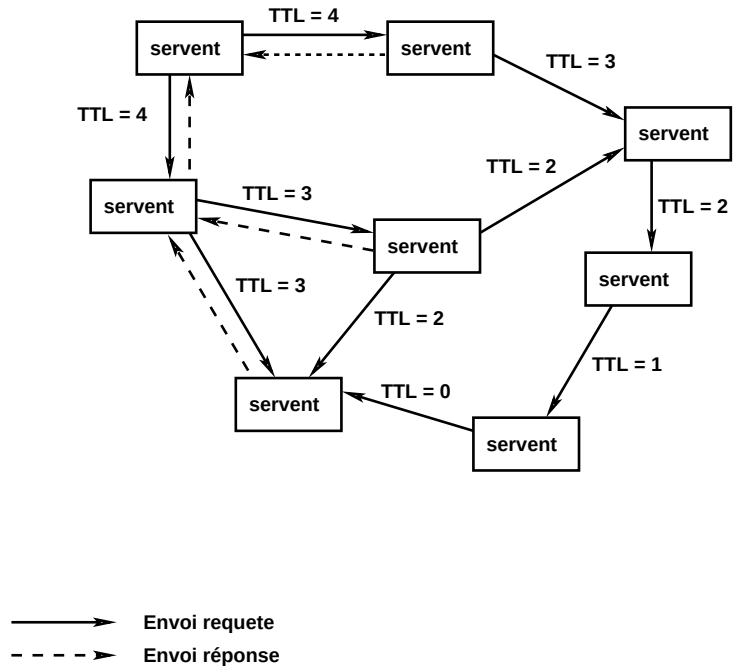

FIG. 5.1 – Recherche de fichiers et propagation du TTL dans un réseau Gnutella

La particularité du protocole Gnutella est que les requêtes ont une durée de vie limitée. En l'absence de TTL, comme un servent diffuse une requête à tous les autres servent voisins, le réseau serait vite encombré et surchargé.

5.1.2 Description du simulateur

Nous utilisons un simulateur pour recréer un réseau P2P. Il existe plusieurs simulateurs réseaux et Gnutella. Nous avons choisi PeerSim, un simulateur issue du projet BISON (Biology-Inspiredtechniques for Self-Organization in dynamic Networks) [Mont 03].

Le projet BISON veut résoudre les problèmes que des millions de connexions et d'utilisateurs peuvent poser. Il s'inspire des processus biologiques afin de développer des tech-

niques et des outils pour construire un réseau de système d'informations robustes, s'organisant et se réparant seul.

PeerSim a été développé pour supporter le dynamisme d'un réseau. Ce simulateur est composé de composants redéfinissables et "pluggables". Il est écrit en java et téléchargeable sur www.sourceforge.net.

5.1.3 Intégration des normes dans un système de type Gnutella

L'application présentée ci dessous consiste à intégrer des normes dans un réseau de type Gnutella.

Dans les systèmes P2P, les échanges s'effectuent directement entre les membres d'une communauté. Si tous les membres respectent le même protocole, les échanges vont être surs. Le but d'introduire des normes dans un système tel que Gnutella est d'obtenir une coordination entre les membres du groupe et d'assurer leur sécurité.

Le système que nous avons présenté va s'apparenter à un réseau Gnutella. Chaque *servent* sera représenté par un agent.

Dans la suite, nous définissons d'abord le langage de communication que vont utiliser les agents, ensuite les normes introduites dans le système et enfin le fonctionnement de l'application avec les résultats obtenus .

5.2 Fonctionnement de l'application et objectif

Fonctionnement

Notre application s'appuie sur le protocole Gnutella et en particulier sur la notion de TTL.

Les agents définis dans notre système ont pour but de trouver des fichiers à travers le réseau. Chaque agent aura un TTL qui lui est propre, qu'il pourra augmenter s'il ne trouve pas suffisamment de fichiers ou baisser dans le cas contraire et un TTL maximal qui servira de seuil au TTL.

L'intérêt de définir un TTL et un TTL maximal est de donner une certaine liberté aux agents dans la définition de la profondeur à laquelle va se propager une requête dans le réseau. Si un agent a un comportement social, il essayera de limiter la propagation de la requête si ce n'est pas utile en ajustant son TTL à ses besoins.

Au démarrage du système nous fixons une valeur pour le TTL et une pour le TTL maximal qui sont identiques pour tous les agents et pouvant être apparentées à des normes sociales. Il existe une norme qui dit que le TTL doit être inférieur au TTL max, c'est-à-dire que si un agent reçoit une requête avec un TTL qui est supérieur à son TTL max, il peut appliquer une sanction. Cette norme est exprimée sous la forme d'une norme sociale et dépend des paramètres de chaque agent. Les agents vont ensuite pouvoir ajuster le TTL jusqu'à ce qu'il rentre en contradiction avec le TTL max. Dans ce cas là, ils devront modifier la valeur du TTL max. Le problème qui va se poser est qu'un agent pourra avoir un TTL max supérieur à celui de son voisin et un TTL en accord avec son TTL max mais pas avec celui de son voisin. Un agent a donc intérêt à se trouver des voisins avec un TTL maximal en accord avec le sien.

Objectif

Notre objectif est de réorganiser le réseau en fonction du TTL et de la valeur de réputation. Les agents malicieux devraient être exclus du système et des sous groupes

devraient se former en fonction des valeurs de TTL maximal. La reconfiguration du réseau est possible car les agents peuvent à tout moment couper les connexions et changer de voisins.

5.3 Le langage de communication

Il est indispensable de définir le langage de communication avant de définir les normes. En effet, ces dernières vont s'appliquer sur ce langage, leur formalisation va en dépendre.

Dans le cas de la recherche de fichiers, un agent applicatif peut :

- envoyer des requêtes ;
- recevoir des requêtes ;
- traiter les requêtes reçues ;

À partir de cette listes d'actions, les seuls types de communication vont être l'envoi et la réception de requêtes. Nous allons donner la syntaxe et la description du langage utilisé.

5.3.1 Syntaxe du langage

Le langage aura deux performatifs : $ask\_[action]$ et $answer\_[action]$. Les actes de langage, quant à eux, auront la forme suivante :

- $ask\_[action](id, ttl, beneficiaire, sender, receiver, [options])$;
- $answer\_[action](id, ttl, beneficiaire, sender)$.

où id est l'identifiant de la requête, ttl est sa durée de vie, $beneficiaire$ est l'agent à l'origine de la requête, $sender$ est celui qui l'envoie, $receiver$ est celui qui la reçoit, et $[action]$ représentent l'ensemble des actions que les agents peuvent effectuer.

5.3.2 Description du langage

Le langage va donc être décrit à partir des actions que les agents peuvent faire. Les agents possèdent une valeur pour le TTL et une autre valeur servant de seuil au TTL.

Dans le cas de la recherche de fichiers dans un système de type Gnutella, nous avons en fonction des actions le langage de communication suivant :

- rechercher des fichiers
  $\Rightarrow ask\_seek(id, ttl, beneficaire, sender, receiver, fichier)$,
  $\Rightarrow answer\_seek(id, ttl, beneficaire, sender)$;
- modifier la valeur du TTL
  $\Rightarrow ask\_modifyTTL(id, ttl, beneficaire, sender, receiver, valeur)$;
- modifier la valeur du TTL seuil
  $\Rightarrow ask\_modifyTTLmax(id, ttl, beneficaire, sender, receiver, valeur)$,
  $\Rightarrow answer\_modifyTTLmax(id, ttl, beneficaire, sender)$;
- demander le téléchargement du fichier
  $\Rightarrow ask\_give(id, ttl, beneficaire, sender, receiver, fichier)$,
  $\Rightarrow answer\_give(id, ttl, beneficaire, sender)$.

Une action est associée à chaque requête. Les actes de langage $ask\_[action](listeparametres)$ et $answer\_[action](listeparametres)$ donnent respectivement les actions $asked\_[action](listeparametres)$ et $answered\_[action](listeparametres)$. Par exemple, l'action associée à l'envoi d'un message de recherche d'un fichier est : $asked\_seek(id, ttl, beneficaire, sender, receiver, fichier)$.

5.4 Définition des normes

Le langage de communication décrit, nous pouvons définir les normes présentes dans le système.

Nous avons les obligations suivantes :

- à la réception d'une requête, l'agent doit la propager en baissant le TTL de 1 ;
- l'agent doit répondre s'il est en possession du fichier.

Nous avons les normes sociales suivantes :

- le TTL de la requête doit être inférieur au TTL seuil de l'agent qui la reçoit ;
- à la réception d'une requête demandant la modification du TTL max, l'agent peut la propager.

Les valeurs du TTL et du TTL maximal sont considérées comme des normes sociales. Nous avons donc : $norm(TTL, egal(val^*))$ et $norm(TTLmax, inf(val^*))$, où * indique des paramètres dynamiques.

Nous formalisons maintenant, pour chaque action que les agents peuvent effectuer, les protocoles de communications et les obligations associées. Nous prenons trois agents A, B et C.

5.4.1 La recherche de fichier

Lorsqu'un agent reçoit une requête de recherche de fichiers, s'il possède le fichier, il a l'obligation de répondre à l'agent à l'origine de la requête. Ensuite, il doit propager la requête à tous ses voisins en décrémentant la valeur du TTL et si cette valeur est supérieure à 0 (figure 5.2).

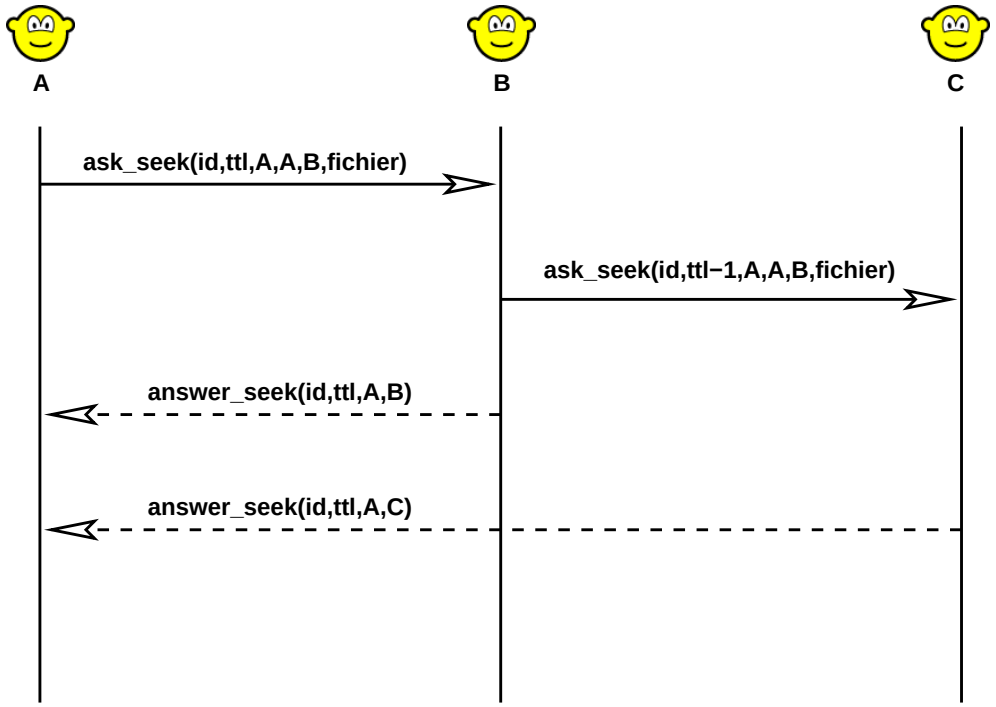

FIG. 5.2 – Protocole de recherche d'un fichier

Obligation 1 : Si TTL>0, faire suivre la requête et baisser le TTL de 1

Formalisation

$rule(asked\_seek(id, ttl, A, B, C, nomFic) \wedge sup(ttl, 0), asked\_seek\_neighbor(id, ttl-1, A, C, [neighbors], nomFic), sanction(ContB, C, NegativeValeur), sanction(ContB, C, PositiveValeur))$

$asked\_seek\_neighbor(id, ttl - 1, A, C, [neighbors], nomFic)$ est un prédicat vérifiant que l'agent C a propagé la requête à tous ses voisins faisant partie de la liste $[neighbors]$.

Obligation 2 : Répondre si en possession du fichier demandé

Formalisation

$rule(asked\_seek(id, ttl, A, B, C, nomFic) \wedge possess(nomFic), answered\_seek(id, ttl, A, C), sanction(ContB, C, NegativeValeur), sanction(ContB, C, PositiveValeur))$

$possess(nomFic)$ est un prédicat qui est vrai si l'agent possède le fichier $nomFic$.

Norme 1 : le TTL de la requête doit être inférieur au TTL maximal de l'agent qui la reçoit

Formalisation

$norm(asked\_seek(id, ttl, A, B, C, nomFic) \wedge inf(ttl, ttlmax^*), treatmentSeek(id, ttl, A, B, C, nomFic))$

$treatmentSeek(id, ttl, A, B, C, nomFic)$ est un prédicat indiquant que les obligations 1 et 2 doivent être respectées.

5.4.2 La modification du TTL et du TTL max

Le traitement des requêtes n'est soumis à aucune obligation mais à des normes sociales.

À la réception d'une requête dont le contenu est la modification du TTL (figure 5.3), l'agent applicatif la perçoit comme un conseil et a le choix de la propager ou non. Par contre s'il reçoit une demande de TTL maximal (figure 5.4), il peut accepter cette modification et dans ce cas répond à l'agent à l'origine de la requête, et peut propager cette requête.

Un agent qui envoie une demande de modification de TTL max part en réalité à la recherche de nouveaux voisins, voisins acceptant son TTL max. Il ne gardera comme voisins que ceux qui accepteront sa demande même s'ils ne sont pas nombreux.

Norme 2 et 3 : le TTL de la requête doit être inférieur au TTL maximal de l'agent qui la reçoit

$norm(asked\_modifyTTL(id, ttl, A, B, C, val), answered\_modifyTTL(id, ttl, A, C))$

$norm(asked\_modifyTTLmax(id, ttl, A, B, C, val), answered\_modifyTTLmax(id, ttl, A, C))$

5.4.3 La demande d'un fichier

Après avoir envoyé une requête de recherche d'un fichier (figure 5.5), l'agent traite les réponses qu'il a reçu et demande le fichier à un agent. L'agent qui est en possession du fichier n'est pas obligé de le lui donner. En effet, il a répondu à la demande du premier

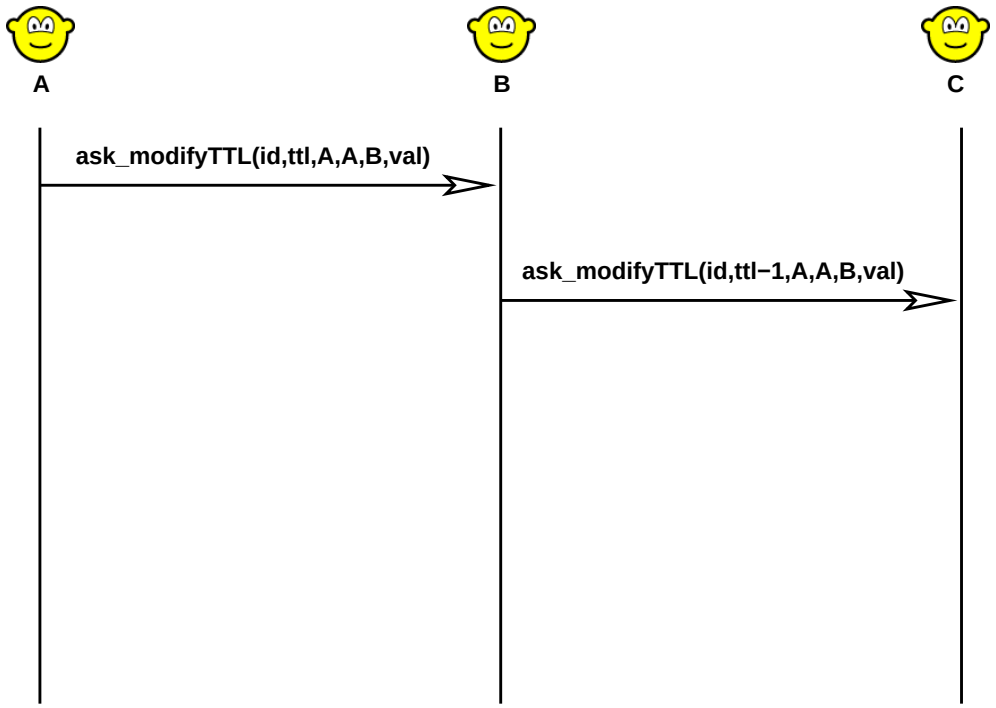

FIG. 5.3 – Protocole de demande de modification du TTL

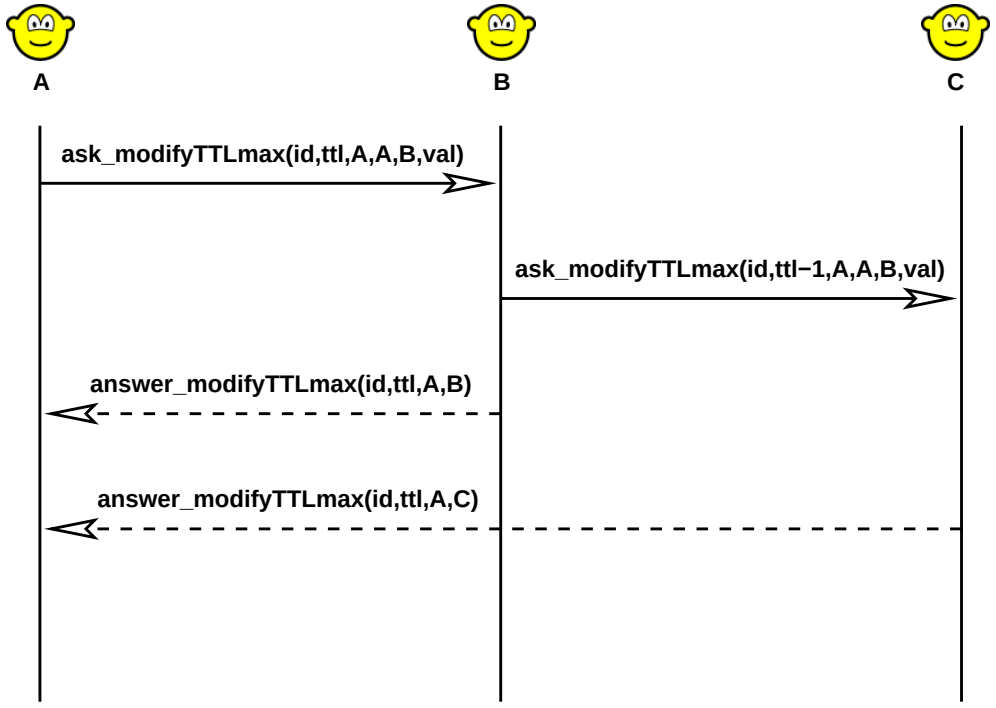

FIG. 5.4 – Protocole de demande de modification du TTL seuil

agent par obligation, mais ne veut pas le donner pour diverses raisons, par exemple à cause d'une mauvaise réputation qu'aurait l'agent initial, ou encore parce qu'il ne fait plus partie du réseau.

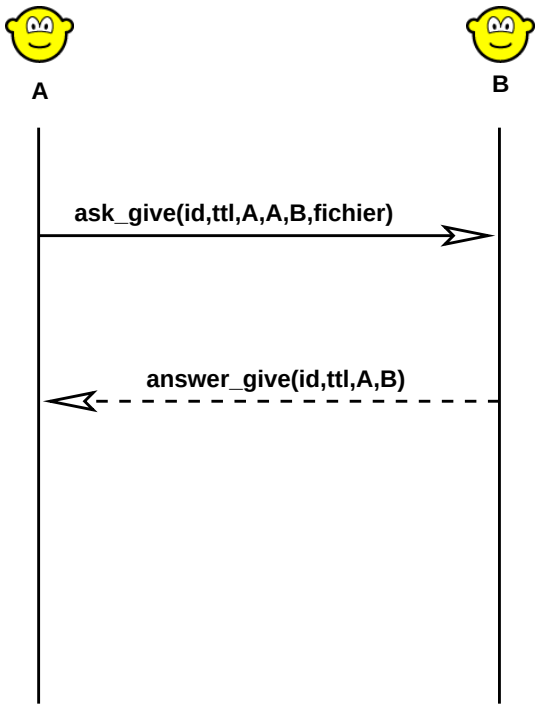

FIG. 5.5 – Protocole de demande d'un fichier

5.5 Comportement des agents

Dans l'application, nous définissons plusieurs comportements d'agents pour simuler l'hétérogénéité du système. Nous allons détailler les comportements que peuvent avoir les agents en fonction de différentes situations.

1. **Réception d'une requête de recherche de fichiers**
   - l'agent respecte les obligations 1 et 2,
   - l'agent ne respecte pas les obligations 1 et 2 ;
2. **Réception d'une requête de modification de TTL**
   - l'agent accepte le nouveau TTL,
   - l'agent respecte la norme 2 c'est-à-dire qu'il va propager la requête,
   - l'agent accepte le nouveau TTL et respecte la norme 2,
   - l'agent n'accepte pas le nouveau TTL et ne respecte pas la norme 2 ;
3. **Réception d'une requête de modification du TTL max**
   - l'agent accepte le nouveau TTL,
   - l'agent respecte la norme 3 c'est-à-dire qu'il va propager la requête,
   - l'agent accepte le nouveau TTL et respecte la norme 3,
   - l'agent n'accepte pas le nouveau TTL et ne respecte pas la norme 3 ;
4. **Absence de réponses suite à une requête de modification du TTL max**
   - l'agent change la valeur de son TTL maximal même si aucun agent n'y adhère,
   - l'agent ne fait rien.

Il est évident que les agents pourraient avoir bien d'autres comportements mais ceux-ci suffisent pour montrer l'intérêt de l'ajout des normes dans le système.

5.6 Résultats obtenus

Nous montrons comment l'intégration de normes dans le système influence sur le nombre de fichiers obtenus par des agents et sur la reconfiguration du réseau. Nous détaillons tout d'abord l'évolution d'un petit réseau pour ensuite montrer les résultats obtenus sur un plus gros réseau. Les tests sont fait sur un nombre de cycles que nous préciserons. À chaque cycle, un agent traite tous les messages qu'il a reçu et envoie une requête pour ses propres intérêts.

Les figures 5.7 et 5.8 montrent la différence de fichiers obtenus par tous les agents du système dans un réseau avec normes et sans norme.

5.6.1 Réseau à 6 agents

Prenons l'exemple d'un réseau contenant 6 agents, numerotés de 0 à 5. Le TTL est fixé à 1 et le TTL maximal à 3 au démarrage du système. Les agents 2 et 3 ne respectent pas les normes. Des commentaires sur les résultats sur la reconfiguration du réseau et sur le nombre de fichiers obtenus dans un système avec et sans normes, et ce durant 25 cycles sont faits.

Reconfiguration du réseau

La figure 5.6 illustre l'évolution d'un réseau. Les cycles présents dans la figure représente tous les changements qu'a pu avoir ce réseau. Le cycle 1 le représente à sa création. Le réseau se modifie au cycle 6 où tous les agents ont fait évoluer leur TTL maximal. Des liens se sont créés entre les agents 2 et 5 et les agents 0 et 2, l'agent 5 a accepté d'augmenter son TTL maximal même s'il n'en a pas immédiatement besoin. Au cycle 12, nous pouvons constater que les agents 0, 1, 4 et 5 ne veulent plus de l'agent 2 comme voisin à cause d'une trop mauvaise réputation. En effet, l'agent 2 est un agent malicieux et ne respectant pas les normes s'est trop fait sanctionner. Au cycle 13, ce sont les agents 0 et 1 qui ont exclu l'agent 3, lui aussi malicieux. Les deux agents non respectueux des normes se retrouvent à part. Nous pouvons remarquer que les quatre autres agents forment un réseau linéaire. Ils vont réussir à se reconnecter entre eux comme le montre le cycle 18 grâce à leur recherche de nouveaux voisins. Le cycle 24 montre juste l'évolution du TTL et du TTL maximal. L'agent 5 est le seul qui possède un TTL maximal inférieur à celui des autres mais il reste dans ce petit réseau pour ne pas se retrouver seul, ce qui ne lui permettrait plus d'obtenir des fichiers.

Fichiers obtenus en fonction du type de réseau

La figure 5.7 montre la différence de fichiers obtenus par les agents en intégrant ou non des normes dans le système. Si le système n'est régi par aucune norme, les agents connaissent quand même ce qu'ils doivent faire à la réception d'une requête pour la recherche d'un fichier. En revanche, il ne peuvent plus modifier le TTL. Nous pouvons observer que du cycle 0 jusqu'au 5 il n'y a aucune différence, ensuite du cycle 5 jusqu'au 13 le nombre de fichiers obtenus est plus important sans les normes. Cela peut s'expliquer par le fait que dans un système normatif, les agents connaissent la réputation des agents avec lesquels ils interagissent et auront tendance à refuser ou à ne pas répondre aux requêtes venant d'agents malicieux. Cette différence est due aux agents malicieux qui obtiennent des fichiers, donnés par les autres en ne sachant pas qu'ils ne sont pas honnêtes. Ensuite, au cycle 15 la différence s'inverse, c'est-à-dire que le nombre de fichiers obtenus est plus

important dans un système normatif. Ce changement s'effectue au moment où les agents malicieux sont exclus du système. Ces deux agents empéchaient les autres de s'échanger leur fichier, en bloquant les requêtes par exemple.

Synthèse

Ces deux résultats montrent qu'intégrer des normes dans un système multi-agent ouvert et décentralisé permet d'exclure les agents malveillants. Donner de l'autonomie aux agents pour qu'il puisse adapter le TTL à leurs besoins leur permet de trouver plus de fichiers. Dans cette aplication, les agents applicatifs vont émerger des normes sociales mais ils pourraient très bien les transformer en obligation avec le processus vu dans le chapitre précédent.

5.6.2 Réseau à 35 agents

Des tests ont été faits sur un réseau à 35 agents avec la configuration initiale représentée par la figure 5.9. Après 45 cycles, le réseau obtient la forme décrite par la figure 5.10. Nous pouvons constater que le réseau a évolué, excluant plusieurs agents, parmi eux se trouvent des agents malicieux ainsi que des agents qui n'ont pas accepté des TTL maximaux différents du leur.

Les résultats de la figure 5.8 illustre encore une fois le fait que le nombre de fichiers trouvés dans un réseau avec normes est plus important que dans un réseau sans normes, d'une part parce que les agents malicieux sont écartés du réseau et d'autre part parce que les agents applicatifs ont la possibilité de faire évoluer leur TTL en fonction de leurs besoins.

Jusqu'au cycle 16, le nombre de fichiers obtenus est plus important dans un réseau sans norme car aucun agent n'est exclu du système. Passé le cycle 16, les agents commencent à exclure d'autres agents et à augmenter leur TTL, ils atteignent ainsi d'autres fichiers.

5.6.3 Remarque

Dans les deux réseaux finaux (celui à 6 noeuds et celui à 35 noeuds), le TTL maximal des agents est supérieur à la profondeur du réseau. Par exemple, dans le plus petit réseau, au cycle 24, le TTL maximal est de 9 alors que la profondeur maximale du réseau est beaucoup plus petite. Cette valeur élevée du TTL est due au fait que les agents augmentent sans cesse leur TTL tant qu'il n'ont pas trouvé tous leurs fichiers. Il faudrait créer un mécanisme permettant aux agents de se rendre compte que les fichiers ne sont pas dans leur réseau et donc qu'ils ne doivent pas augmenter leur TTL.

5.7 Conclusion

Le système multi-agent que nous avons utilisé s'appuie sur le protocole Gnutella. Les agents ont pour but de chercher des fichiers en envoyant des requêtes, chacune ayant une durée de vie : le TTL. Ces agents vont devoir respecter les normes sous peine de voir leur réputation baissée. Ils ont également la possibilité de modifier la valeur du TTL pour satisfaire leur besoin. Si les agents ne sont pas malicieux, cette adaptation du TTL va permettre une régulation du réseau permettant d'éviter les encombrements inutiles. De plus des groupes d'agents vont se former, partageant chacun une valeur de TTL. Ce phénomène est dû à l'émergence de normes sociales sur les valeurs de TTL.

Un autre point que nous avons montré à travers cette application est que les agents obtiennent plus de résultats dans un système normatif. Ceci s'explique car les agents malicieux sont expulsés du système à cause de leur mauvaise réputation qu'ils se sont données à la suite de violations de normes.

Cette application a permis de montrer l'utilité de l'intégration de normes dans un système multi-agent ouvert et décentralisé.

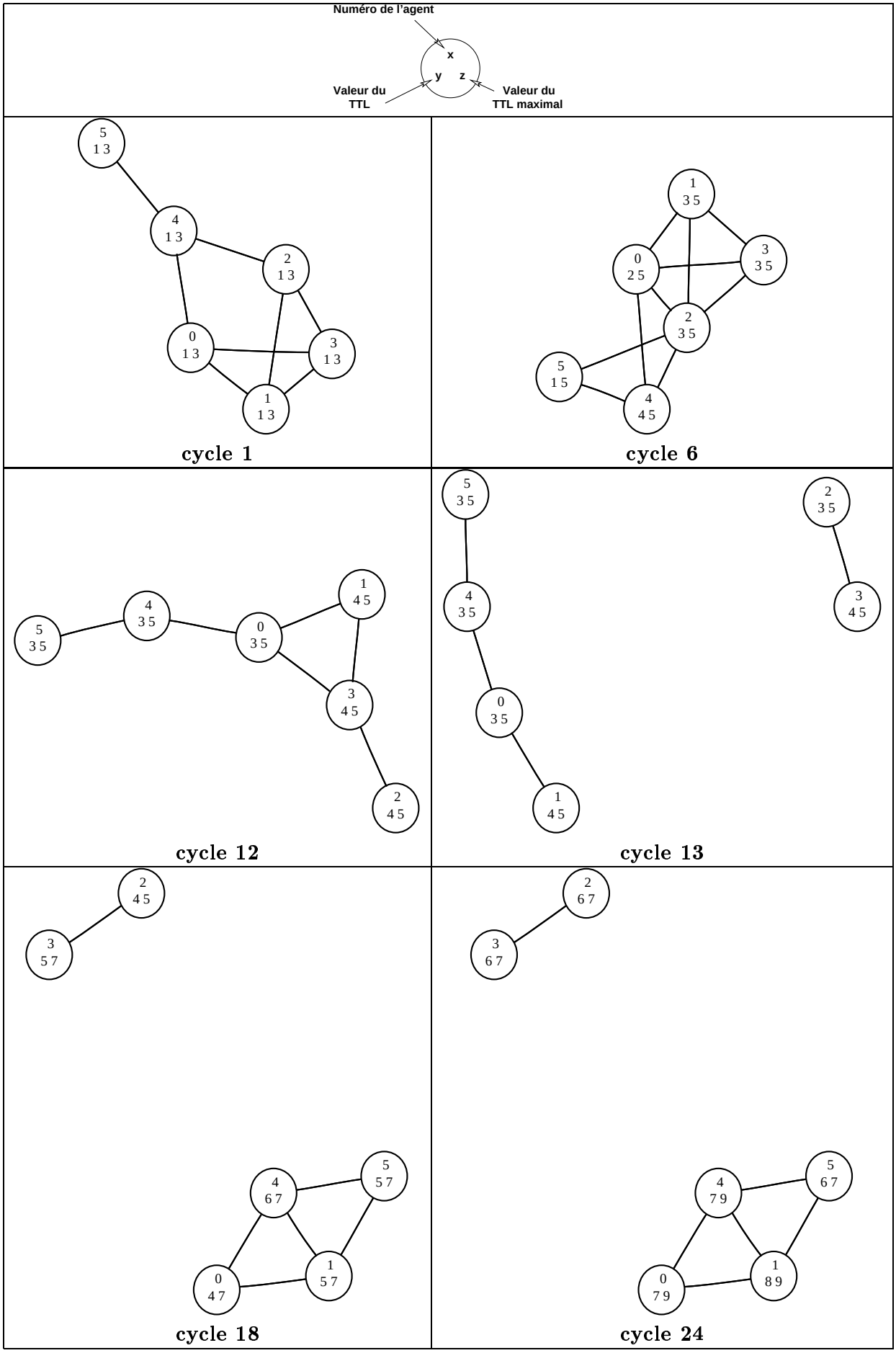

FIG. 5.6 – Evolution d'un réseau à 6 noeuds

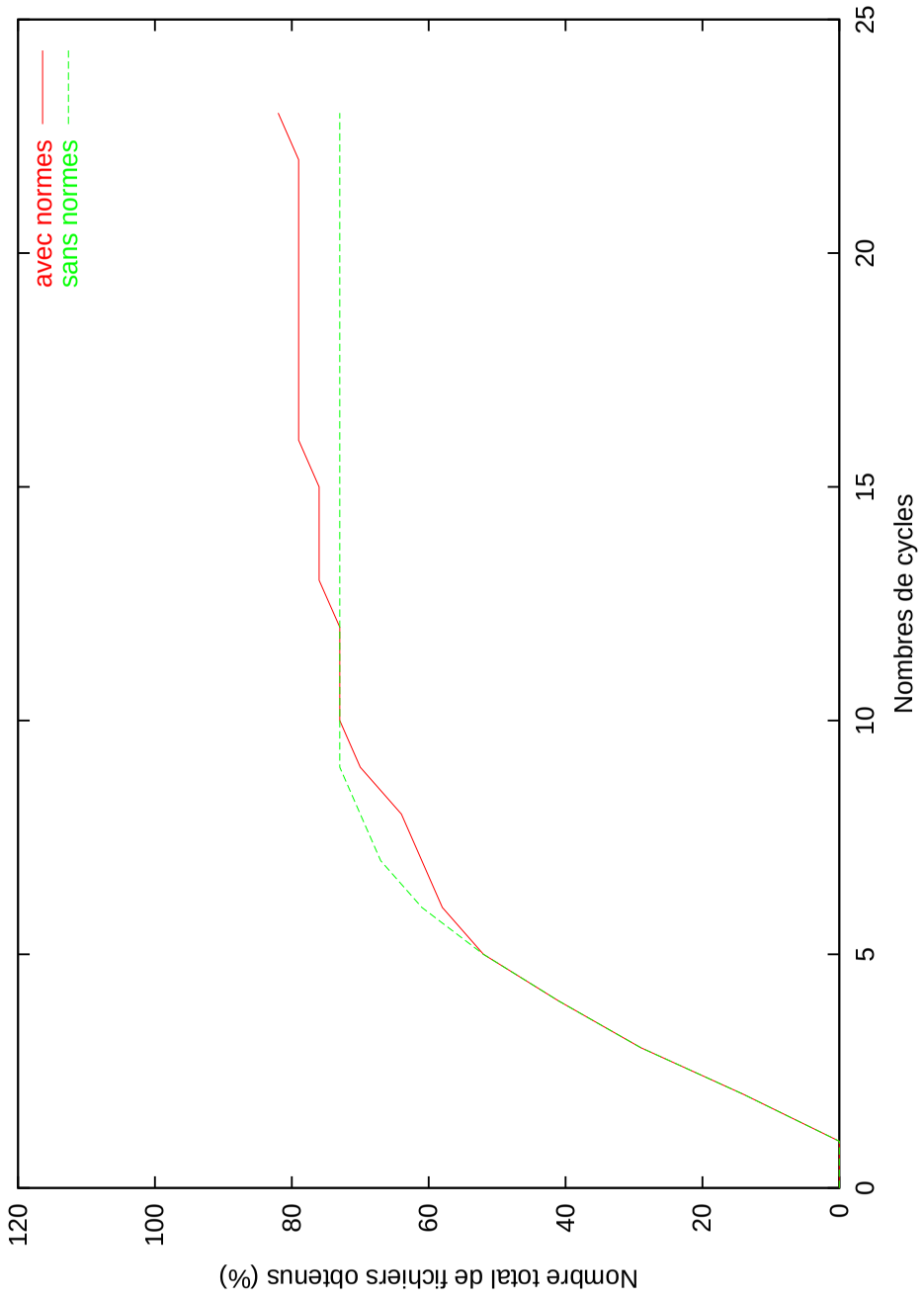

FIG. 5.7 – Résultats sur un réseau à 6 noeuds

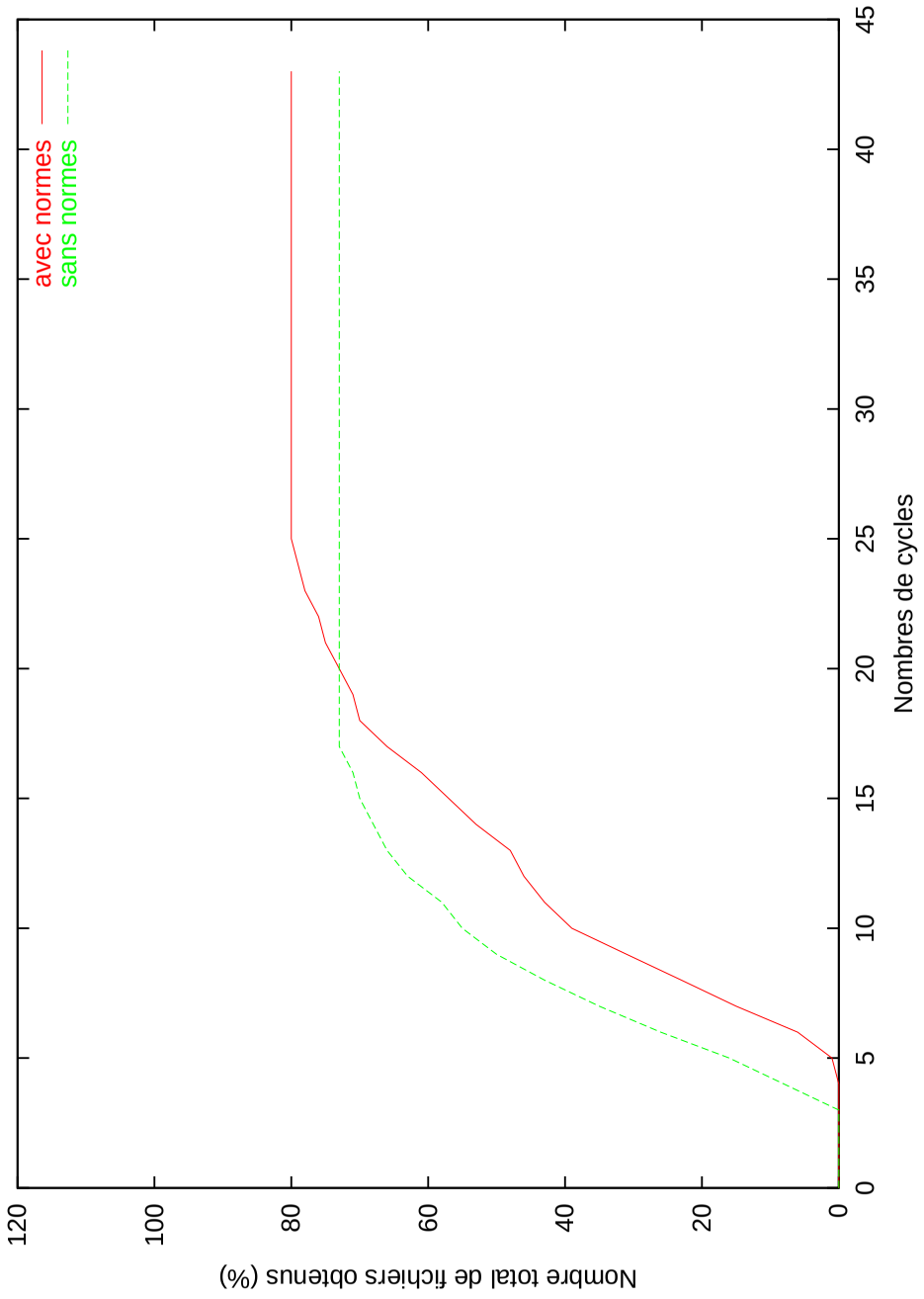

FIG. 5.8 – Résultats sur un réseau à 35 noeuds

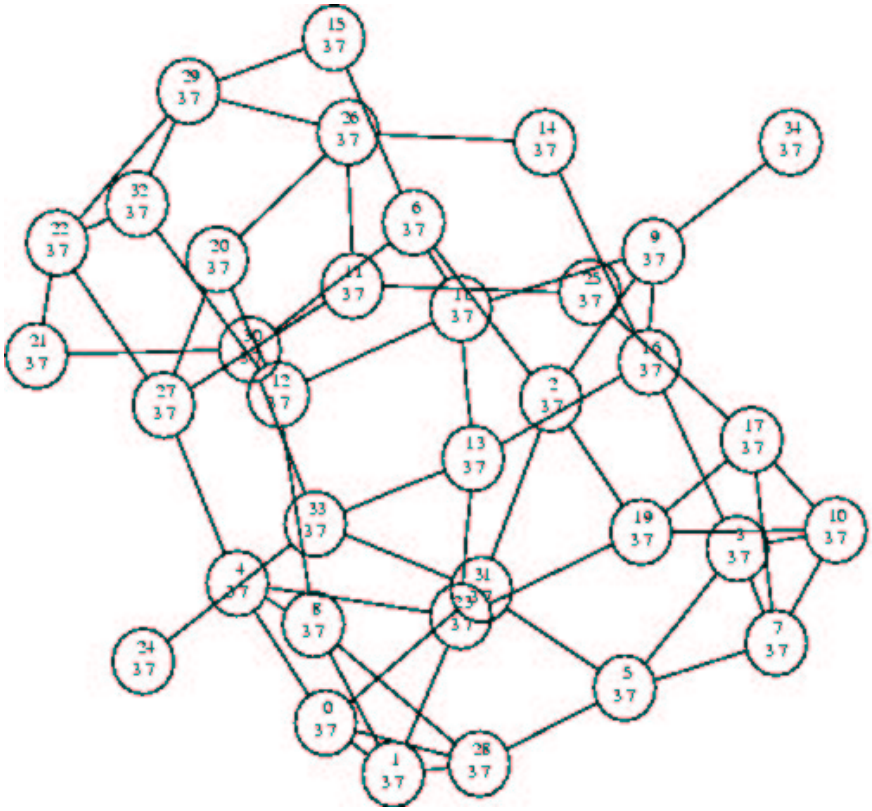

FIG. 5.9 – Réseau initial à 35 noeuds

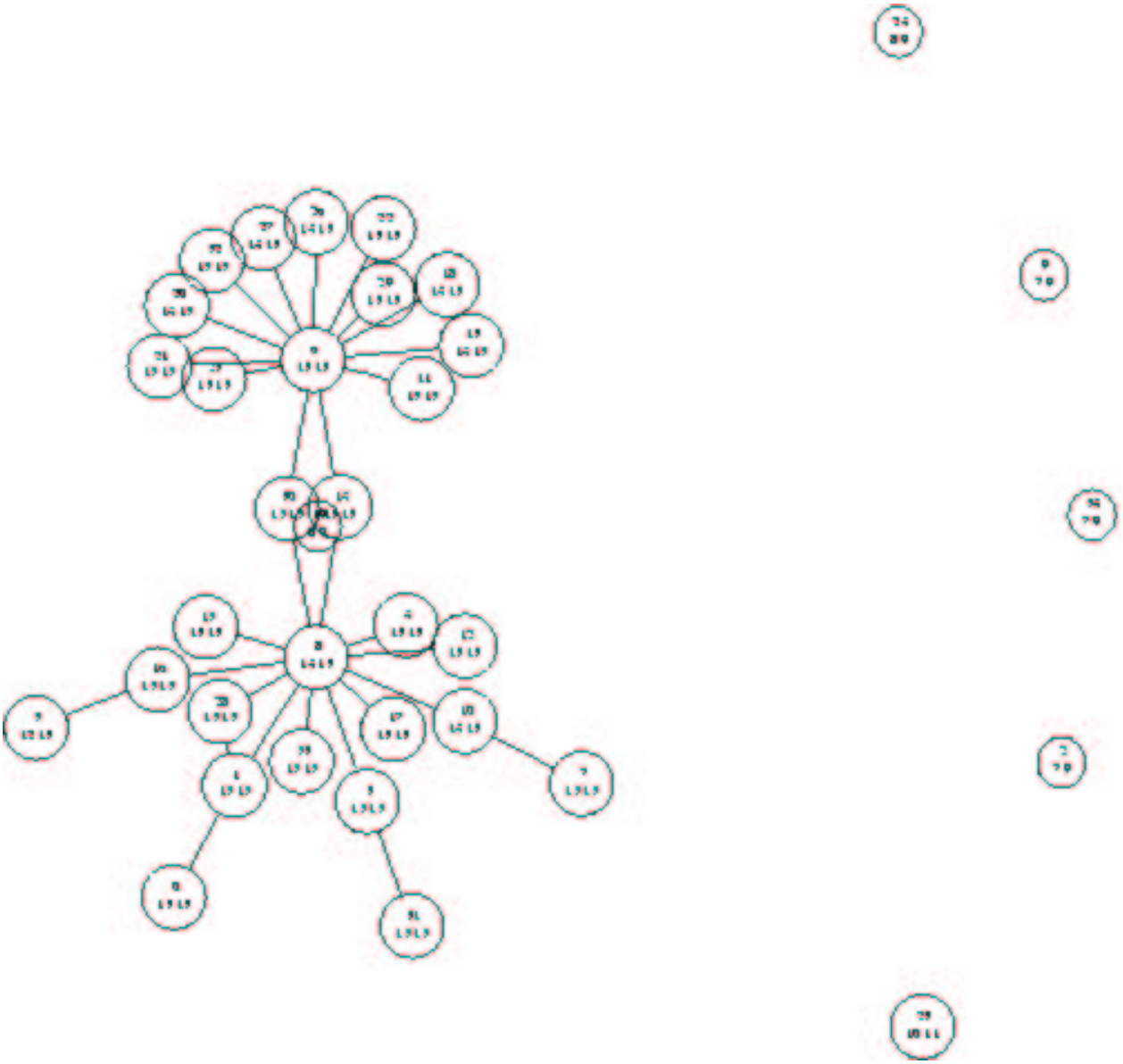

FIG. 5.10 – Réseau final à 35 noeuds

Chapitre 6

Conclusion

Notre objectif était de définir et d'intégrer des normes dans un système multi-agent ouvert et décentralisé afin de donner plus de flexibilité au système et de pouvoir sanctionner les agents ne respectant pas les contraintes définies dans ce système. Nous donnons dans ce chapitre une synthèse de notre travail tant sur le modèle que sur les résultats obtenus. Nous verrons pour finir les perspectives ouvertes par ce travail.

6.1 Synthèse

Nous avons vu dans le premier chapitre différentes interprétations de ce qu'est une norme suivant les domaines d'applications. En général, les normes sont des règles définies dans la société et imposées aux individus par une autorité qui sanctionne ceux qui ne respectent pas ces normes. Nous nous sommes appuyés sur cette définition des normes pour définir deux notions qui sont les obligations et les normes sociales afin de pouvoir construire notre système normatif.

Les obligations sont des contraintes imposées aux agents et dont la violation donne lieu à des sanctions explicites, connues de tous. Les normes sociales sont des règles auxquelles se conforment des groupes d'agents et qui ont émergé suite aux besoins de ces agents. Des sanctions implicites sont appliquées par les agents applicatifs en cas de violation. Pour assurer le respect des obligations, chaque agent applicatif est observé par un agent contrôleur dont le rôle est de détecter les obligations qui ne seraient pas respectées et de sanctionner les agents à l'origine des violations. Les sanctions se font en terme de réputation. Les agents contrôleurs jouent le rôle d'une autorité et sont dispersés dans le système car celui-ci étant décentralisé, il n'est pas possible de définir une autorité centrale. De plus les obligations définies dans le système peuvent ne plus convenir aux agents applicatifs et dans ce cas ces derniers adaptent certaines obligations pour les faire correspondre à leur besoin. Après avoir modifier le contenu d'une obligation, un agent applicatif va diffuser sa nouvelle norme aux agents qu'il perçoit dans son entourage dans le but qu'il l'adopte et que cette norme devienne une norme sociale. Si l'agent pense que sa norme doit être respectée par tous les agents, il va demander à son agent contrôleur de la faire évoluer en obligation de façon à ce que ce dernier puisse la renforcer. L'émergence de normes sociales va être à l'origine de la création de groupe d'agents partageant les même croyances.

L'application que nous avons développée est un système normatif décentralisé et ouvert qui s'appuie sur le protocole Gnutella et où les agents s'échangent des fichiers. Les résultats de cette application nous ont permis de montrer que l'intégration des normes dont les sanctions sont en terme de réputation est à l'origine de la régulation du système. En effet, des groupes d'agents se forment en fonction des normes sociales qu'ils partagent et les

agents malicieux (qui ne respectent pas les normes) sont exclus du système.

6.2 Limites

Nous avons étudié l'intégration des normes dans un système multi-agent ouvert afin de détecter les agents malicieux grâce aux violations des normes qu'ils pourraient commettre. Mais nous n'avons pas considéré le cas où un agent pouvait rentrer et sortir du système à tout moment. Nous n'avons pas non plus envisager le fait que les agents puissent avoir des moyens de communications différents ni qu'ils aient un processus de raisonnement évolué prenant en compte leurs buts à atteindre et les normes à respecter. Une autre limite au modèle est que le formalisme ne permet pas de prendre en compte des normes contradictoires, pouvant rentrer en conflit.

6.3 Perspectives

Dans la suite, une des premières choses à améliorer serait la prise en compte de normes conflictuelles dans le comportement de l'agent. Ce dernier devrait avoir un processus de raisonnement lui permettant de connaître la meilleure chose à effectuer.

Nous pourrions ensuite envisager des agents capables de créer de toutes nouvelles normes en faisant l'hypothèse que ces agents connaissent la composition d'une norme. À l'aide d'un processus d'apprentissage, ils pourraient trouver l'action à effectuer en présence d'une condition jusque là inconnue et être ainsi à l'origine de l'émergence d'une nouvelle norme dans le système. Dans ce cas là, ils ne se contenteraient pas d'adapter une norme mais ils en feraient émerger une nouvelle. Il serait également possible de trouver différents moyens d'apprentissage et de renforcement des normes.

Bibliographie

[Bois 05] O. Boissier. "Systèmes Multi-Agent Master Web Intelligence". http ://www.emse.fr/ boissier/enseignement/sma04/index.html, 2005.

[Cont 01] R. Conte, B. Edmonds, S. Moss, and R. K. Sawyer. "Sociology and Social Theory in Agent Based Social Simulation : A Symposium". In : *Computational & Mathematical Organization Theory*, pp. 183 – 205, Springer Science+Business Media B.V., Formerly Kluwer Academic Publishers B.V., October 2001.

[Cont 99] R. Conte, C. Castelfranchi, and F. Dignum. "Autonomous Norm Acceptance". In : J. Müller, M. P. Singh, and A. S. Rao, Eds., *Proceedings of the 5th International Workshop on Intelligent Agents V : Agent Theories, Architectures, and Languages (ATAL-98)*, pp. 99–112, Springer-Verlag : Heidelberg, Germany, 1999.

[Dign 00] F. Dignum, D. Morley, L. Sonenberg, and L. Cavedon. "Towards Socially Sophisticated BDI Agents". In : *ICMAS '00 : Proceedings of the Fourth International Conference on MultiAgent Systems (ICMAS-2000)*, p. 111, IEEE Computer Society, Washington, DC, USA, 2000.

[Dign 02] F. Dignum. "Abstract norms and electronic institutions". In : G. Lindemann, D. Moldt, M. Paolucci, and B. Yu, Eds., *RASTA workshop*, pp. 93–103, July 2002.

[Dign 99] F. Dignum. "Autonomous Agents with Norms". *Artificial Intelligence and Law*, Vol. 7, No. 1, pp. 69–79, 1999.

[Ferb 95] J. Ferber. *Les systèmes multi-agents.* Interéditions, 1995.

[Ione 04] M. F. Ionescu, N. H. Minsky, and T. D. Nguyen. "Enforcement of Communal Policies for P2P Systems.". In : R. D. Nicola, G. L. Ferrari, and G. Meredith, Eds., *COORDINATION*, pp. 152–169, Springer, 2004.

[Lope 02] F. López y López, M. Luck, and M. d'Inverno. "Constraining autonomy through norms". In : *AAMAS '02 : Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pp. 674–681, ACM Press, 2002.

[MINS 86] M. MINSKY. *The society of mind.* Simon and Schuster, 1986.

[Mont 03] A. Montresor, G. D. Caro, and P. E. Heegaard. *Architecture of the Simulation Environment.* June 2003.

[Mull 04] G. Muller and L. Vercouter. "Détection Décentralisée d'Agents Menteurs". In : O. Boissier and Z. Guessoum, Eds., *Actes des Journées Francophones des Systèmes Multi-Agents (JFSMA'04)*, pp. 243–248, Hermes–Lavoisier, 24–26 novembre 2004.

[Pasq 04] P. Pasquier, R. A. Flores, and B. Chaib-draa. "The enforcement of flexible social commitments". In : F. Z. M.-P. Gleizes, A. Omicini, Ed., *Fifth International Workshop Engineering Societies in the Agents World (ESAW)*, pp. pages 153–165, Springer-Verlag, 2004. Lecture Notes in Artificial Intelligence (LNAI).

[Rao 91] A. S. Rao and M. P. Georgeff. "Modeling Rational Agents within a BDI-Architecture". In : J. Allen, R. Fikes, and E. Sandewall, Eds., *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning (KR'91)*, pp. 473–484, Morgan Kaufmann publishers Inc. : San Mateo, CA, USA, 1991.

[Shoh 97] Y. Shoham and M. Tennenholtz. "On the emergence of social conventions : modeling, analysis, and simulations". *Artif. Intell.*, Vol. 94, No. 1-2, pp. 139–166, 1997.

[Stra 02] T. Stratulat. *Systèmes d'agents normatifs : concepts et outils logiques.* PhD thesis, Université de Caen, 2002.

[Tuom 95a] R. Tuomela. *The Importance of Us : A Philosophical Study of Basic Social Norms.* Stanford University Press, 1995.

[Tuom 95b] R. Tuomela and M. Bonnevier-Tuomela. "Norms and Agreement". *European Journal of Law, Philosophy and Computer Science 5*, pp. 41–46, 1995.

[Wrig 51] G. H. von Wright. *Mind, a quaterly review of psychology and philosophy*, Chap. 1, pp. 1–15. Vol. 60, Zdinburgh : T. Nelson and sons ltd, January 1951.